IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

# E-TDMRC: A novel approach in Cryptography

[1]Antu Annam Thomas

[1]Assistant Professor
[1]Dept. of Computer Applications
[1]Mar Thoma College, Tiruvalla.

*Abstract:* When the cryptographic systems were studied the role of randomness is very high. Time Dependent Multiple Random Cipher code is an effective encryption tool. It was studied that the system can be improved by improving the random number generation technique used. Random number generation technique is employed at three points in the algorithm. Increased level of randomness or by introducing improved random number generators in all these three areas the level of security offered by the code can be increased. So the paper revolves around Improving TDMRC Encryption System by bettering Randomness and thus coming up with a new encryption system known as Enhanced TDMRC (E-TDMRC).

***Index Terms* -TDMRC, LCG, NLCG.**

## I. Introduction

Huge amount of data of the order of millions or trillions of bytes is transferred everyday over internet. The security of this data is of top priority and a major challenge. Even a slight change in a bit of data cause many catastrophic effects. D*ata* can easily get lost in a *security* breach. Methods used for the prevention of security breaches are called data security measures. Data security is thus essential for integrity, reliability, usability, secrecy and safety of data.[1]-[4][10][11] Encryption is one such technique for data security. Time Dependent Multiple Random Cipher Code (TDMRC) is a tool for encryption. [5]

TDMRC can be called as 'Mega Extended ASCII Code' since it uses 256! ways of arranging 256 ASCII characters for encoding. Time Dependency, Poly Alphabetic Nature and use of Pseudo Random Number Generation Technique for code generation are the three complexities that are involved in TDMRC encryption system. Key of TDMRC Code consists of three elements Master Key that is derived from Real Time Clock, Poly Alphabetic Coefficient- P, that decides the number of cipher characters for any plain text character and P number of 4 digit sub keys.[5]

For random number generation traditional pseudo random number generator LCG is employed in TDMRC. In LCG the period length depends upon the value of increment, multiplicand and modulus. Brute Force attack is likely to be successful since the period repeats.[9] And added to these drawbacks increment, multiplicand and modulus if not properly chosen, the random numbers generated will not cover the entire range of possible values. Due to these disadvantages LCG cannot be good to be used for cryptographic applications. [6]-[9]

Thus the significance of choosing NLCG in place of LCG is important. In NLCG the period length is infinity since nesting is incorporated. Every number in a range defined by modulus value is likely to occur in case of NLCG. Due to period which is infinity and uniform occurrence of random numbers Brute Force attack is less likely or rather impossible. Due to close to ideal performance NLCG is cryptographically secure and be used for security applications. Thus TDMRC algorithm can be made more secure by any improvement in the traditional LCG. In this paper E-TDMRC is developed where LCG is substituted with NLCG thus ensuring improved security. [12][13]

## II. Structure of E-TDMRC

Enhanced Time Dependent Multiple Random Cipher Code's structure includes four levels of complexities.

### 2.1 Enhanced Time Dependency

E-TDMRC Code is time dependent. The codes used for any character differs depending upon time. Random Seeds that is used for code generation depends upon the Random Sub key value and the Master Key value. Master key value is the Real Time Clock value itself. Random Sub Key value is generated by NLCG that also uses Real Time Clock value in its algorithm thus contributing to Enhanced Time Dependency.

### 2.2 Enhanced Poly Alphabetic Nature

It is poly alphabetic. The code used for the same character at different locations of the plain text is different. Poly Alphabetic Coefficient (PAC) decides the number of cipher characters for any plain text character. PAC is generated by NLCG hence PAC value is ensured to be more random.

### 2.3 Enhanced Random Sub keys

NLCG is employed for generating P number of sub keys. Sub keys are multiplied by Master Key which is another true random value (System Clock Value) to generate P number of random seeds. Random Seed values are used for generating P number of Random Series. Since NLCG is used the randomness offered by Random Seed value is enhanced.

### 2.4 Enhanced Pseudo Random Nature

It uses NLCG as Random number generation technique for code generation. Since NLCG is used the Random series generated will possess all the advantages that an NLCG generated series will possess. Depending upon the random seed the codes will change. [5][12][13]

## III. Key of Enhanced-TDMRC

Key determines the security offered by the system. The key in Enhanced TDMRC involves three elements that together makes it more effective than TDMRC in cryptographic applications. Three elements that makes up key in E-TDMRC are given below.

### 3.1 Master Key

Master Key is derived from the Real Time Clock, as in the case of TDMRC. Real Time Clock value is a True Random Number value which is an 8 digit number obtained by combining the values of hour, minute, second and centi second.

### 3.2 Poly Alphabetic Coefficient (PAC)

PAC is a number, P, indicating the number of codes simultaneously used for any character in an encrypting session. This is to be decided at encryption stage.

Though the definition of PAC is same the method of generation used is different in E-TDMRC. PAC is generated by NLCG. Since NLCG is employed the randomness of 'P' is improved.

### 3.3 P Number of 4 digit Sub Keys

Four digit Sub Keys are also generated by NLCG. These four digit Sub Keys are multiplied with Master Key value to generate random seeds.

Thus increased randomness of the four digit Sub Keys increases the randomness of Random Seed values which is the product of Master Key value and the Sub Key values.

Thus from the above explanation on the structure of the key it is clear that the randomness of the key is enhanced which makes E-TDMRC cryptographically more secure. Random Sub Keys and PAC value P was generated using NLCG which promised increased performance. [5][12][13]

## IV. NLCG in E-TDMRC

Pseudo Random Number generator LCG is employed in three areas of TDMRC. Thus by substituting LCG with NLCG the entire system can be made more secure.

The areas in which LCG is substituted with NLCG are as follows:

### 4.1 Finding the Poly Alphabetic Coefficient- P

Poly Alphabetic Coefficient has a great role in making the system cryptographically secure. It is this factor that decides how many codes are used simultaneously for a character substitution. Thus by making this decision more random a drastic improvement can be made to the security of the system. Thus for improving randomness LCG is substituted with NLCG. [64]

### 4.2 Generating P Number of Sub Keys

Sub Key values are multiplied with Master Key value to generate Random Seed values. Thus Randomness factor of Sub Key values have significance and thus NLCG is introduced in place of traditional LCG for increased randomness.

### 4.3 Generating P Random Series

Random Series is that which decides which character is to be substituted for a given character. Close to ideal randomness is ensured for the random series by using NLCG as the series generator.

In short wherever NLCG is introduced in place of LCG performance of TDMRC is enhanced. The performance of a cryptographic algorithm is strongly related to the key strength. Since NLCG is used in E-TDMRC the key strength is increased.

## V. Algorithm of Enhanced TDMRC Code

The algorithm for Encryption and Decryption of E-TDMRC is given below.

### 5.1 Encryption Algorithm

*Step 1* Generate Poly Alphabetic Coefficient, P by using NLCG. PAC decides the number of codes that is to be used simultaneously. PAC is decided by using traditional random number generator LCG.

*Step 2* Now P number of Sub Keys are generated using NLCG.
Sub keys are generated randomly by using LCG. $S_1S_1S_1S_1$, $S_2S_2S_2S_2$,----, $S_pS_pS_pS_p$ are the sub keys that are generated.

*Step 3* Read the Real Time Clock Time (System Time ) with
accuracy to centi second and form an 8 digit number TTTTTTTT. This will act as the Master Key. Since real time clock value is used true randomness is exhibited by the value of master key.

*Step 4* Random seed is generated by multiplying the Master
Key with the Sub Key values and taking 8 digits of the product from extreme right to form P number of Random Seed values.

*Step 5* NLCG is employed for generating random series with
elements from 0-255, (00000000 - 11111111 in binary) arranged in a random fashion.

*Step 6* Now, data is taken in blocks of P number of ASCII
characters. ASCII value of each character is read and each character is substituted with element in the random series corresponding to this ASCII value. First random series is used for substitution of first character in block of P, ASCII characters and so on. [5]

Flowchart of the Encryption algorithm is given below. Chain of 8 bit ASCII is converted into chain of 8 bit E-TDMRC through the process of transliteration.

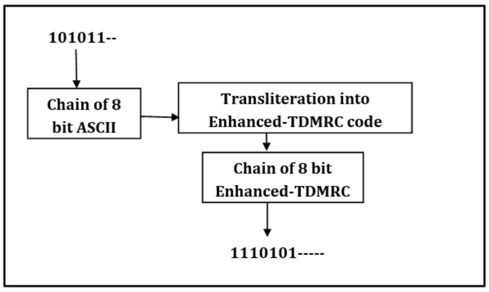

101011--
Chain of 8 bit ASCII
Transliteration into Enhanced-TDMRC code
Chain of 8 bit Enhanced-TDMRC
1110101-----

Figure 5.1 Encryption using E-TDMRC

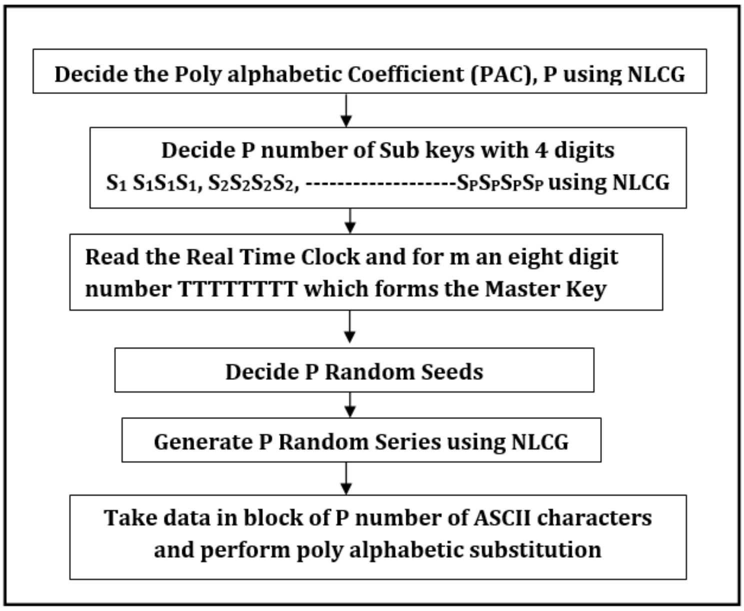

Decide the Poly alphabetic Coefficient (PAC), P using NLCG
Decide P number of Sub keys with 4 digits
$S_1$ $S_1S_1S_1$, $S_2S_2S_2S_2$, -------------------$S_PS_PS_PS_P$ using NLCG
Read the Real Time Clock and for m an eight digit number TTTTTTTT which forms the Master Key
Decide P Random Seeds
Generate P Random Series using NLCG
Take data in block of P number of ASCII characters and perform poly alphabetic substitution

Figure 5.2 Process of Transliteration into E-TDMRC

## 5.2 **Decryption Algorithm**

***Step 1*** Regenerate P number of random seeds and P numbers of random series using the same key used in the Encryption process. The Pseudo Random Number Generation algorithm used is NLCG.

***Step 2*** Now, data is taken in blocks of P number of Enhanced TDMRC coded characters. Find the ASCII value of each character and then substitute each character with the string character of the serial number value of the element in the random series, the element which is same as the ASCII character in the block. The first character in block of P characters be substituted with the string character of the serial number value of the element from first random series, second character with the string character with the serial number value of the element from second random series and so on.[5]

Flowchart of the Decryption Algorithm is given below in Figure 5.3

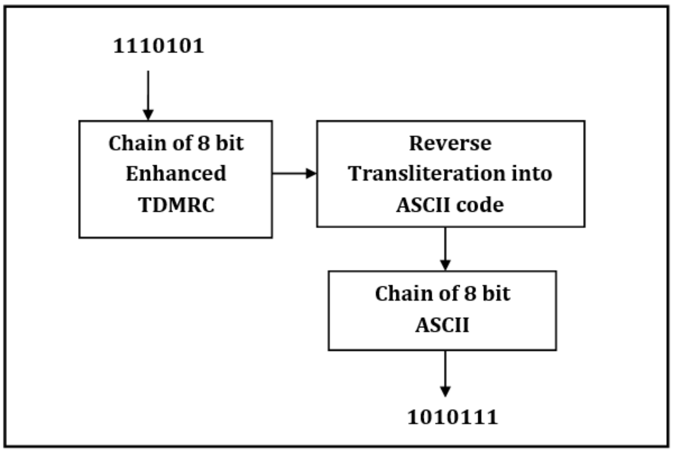
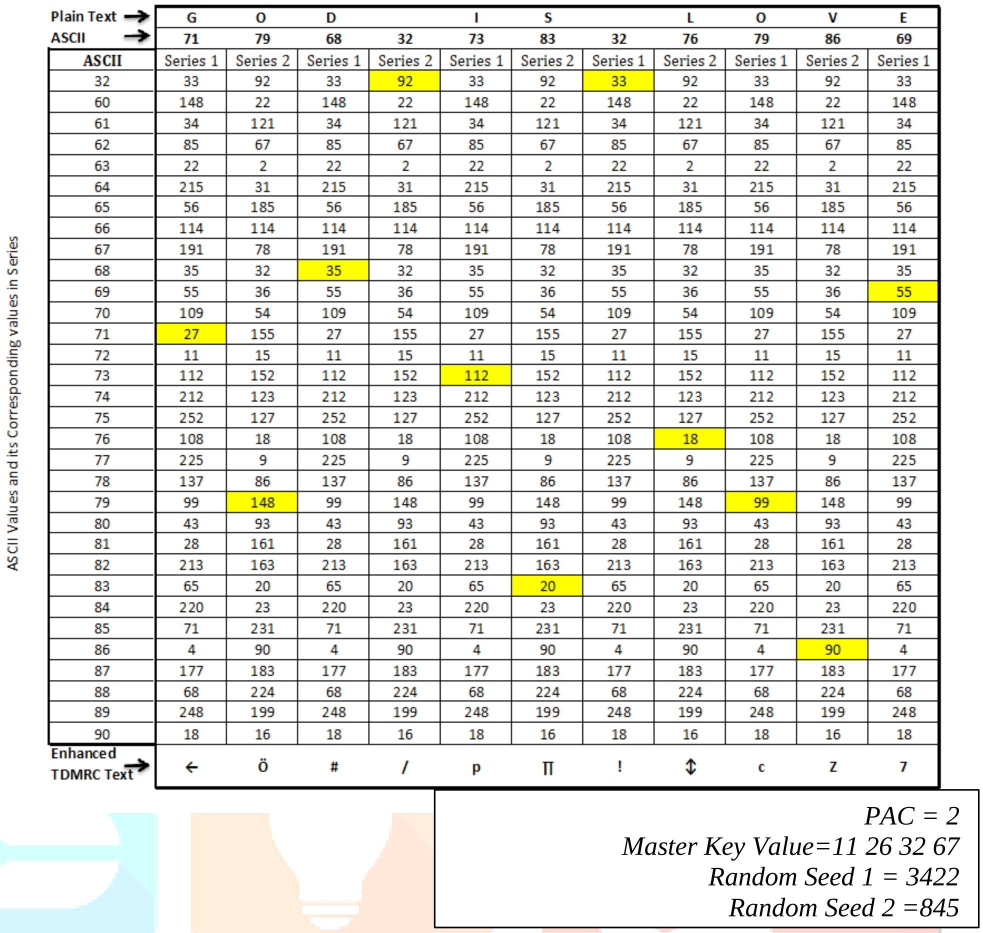

1110101
Chain of 8 bit Enhanced TDMRC
Reverse Transliteration into ASCII code
Chain of 8 bit ASCII
1010111

Figure 5.3 Enhanced TDMRC Decryption System

## VI. IMPLEMENTATION AND RESULT

Both Encryption and Decryption algorithm of Enhanced TDMRC were implemented and the output is shown below.



ASCII Values and its Corresponding values in Series

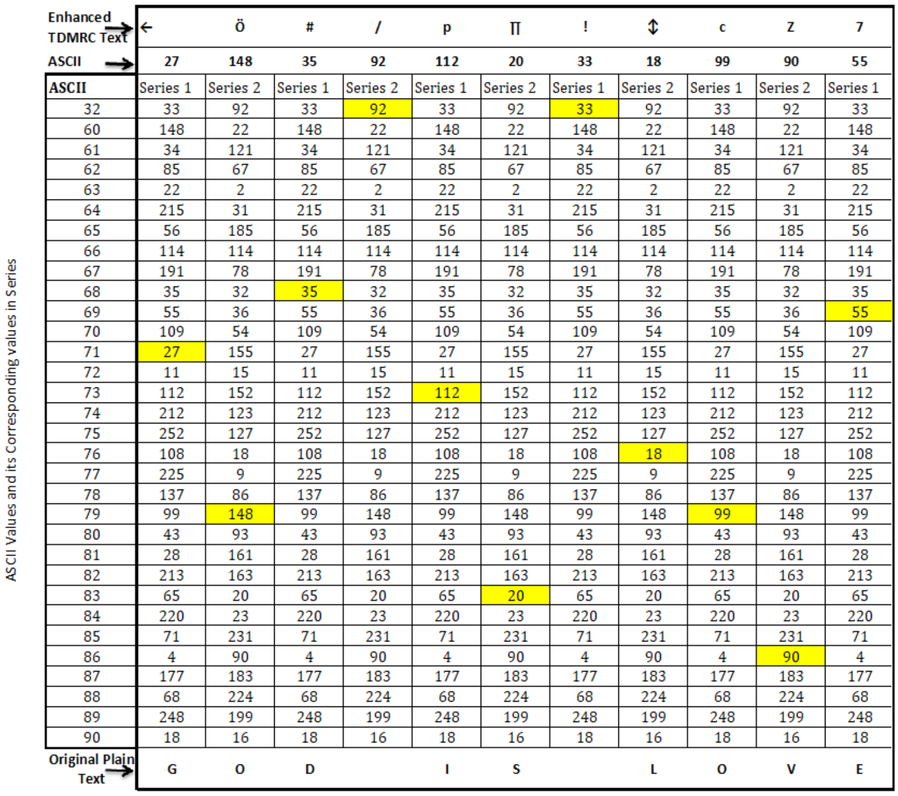
| Plain Text → | G | O | D | | I | S | | L | O | V | E |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ASCII → | 71 | 79 | 68 | 32 | 73 | 83 | 32 | 76 | 79 | 86 | 69 |
| ASCII | Series 1 | Series 2 | Series 1 | Series 2 | Series 1 | Series 2 | Series 1 | Series 2 | Series 1 | Series 2 | Series 1 |
| 32 | 33 | 92 | 33 | 92 | 33 | 92 | 33 | 92 | 33 | 92 | 33 |
| 60 | 148 | 22 | 148 | 22 | 148 | 22 | 148 | 22 | 148 | 22 | 148 |
| 61 | 34 | 121 | 34 | 121 | 34 | 121 | 34 | 121 | 34 | 121 | 34 |
| 62 | 85 | 67 | 85 | 67 | 85 | 67 | 85 | 67 | 85 | 67 | 85 |
| 63 | 22 | 2 | 22 | 2 | 22 | 2 | 22 | 2 | 22 | 2 | 22 |
| 64 | 215 | 31 | 215 | 31 | 215 | 31 | 215 | 31 | 215 | 31 | 215 |
| 65 | 56 | 185 | 56 | 185 | 56 | 185 | 56 | 185 | 56 | 185 | 56 |
| 66 | 114 | 114 | 114 | 114 | 114 | 114 | 114 | 114 | 114 | 114 | 114 |
| 67 | 191 | 78 | 191 | 78 | 191 | 78 | 191 | 78 | 191 | 78 | 191 |
| 68 | 35 | 32 | 35 | 32 | 35 | 32 | 35 | 32 | 35 | 32 | 35 |
| 69 | 55 | 36 | 55 | 36 | 55 | 36 | 55 | 36 | 55 | 36 | 55 |
| 70 | 109 | 54 | 109 | 54 | 109 | 54 | 109 | 54 | 109 | 54 | 109 |
| 71 | 27 | 155 | 27 | 155 | 27 | 155 | 27 | 155 | 27 | 155 | 27 |
| 72 | 11 | 15 | 11 | 15 | 11 | 15 | 11 | 15 | 11 | 15 | 11 |
| 73 | 112 | 152 | 112 | 152 | 112 | 152 | 112 | 152 | 112 | 152 | 112 |
| 74 | 212 | 123 | 212 | 123 | 212 | 123 | 212 | 123 | 212 | 123 | 212 |
| 75 | 252 | 127 | 252 | 127 | 252 | 127 | 252 | 127 | 252 | 127 | 252 |
| 76 | 108 | 18 | 108 | 18 | 108 | 18 | 108 | 18 | 108 | 18 | 108 |
| 77 | 225 | 9 | 225 | 9 | 225 | 9 | 225 | 9 | 225 | 9 | 225 |
| 78 | 137 | 86 | 137 | 86 | 137 | 86 | 137 | 86 | 137 | 86 | 137 |
| 79 | 99 | 148 | 99 | 148 | 99 | 148 | 99 | 148 | 99 | 148 | 99 |
| 80 | 43 | 93 | 43 | 93 | 43 | 93 | 43 | 93 | 43 | 93 | 43 |
| 81 | 28 | 161 | 28 | 161 | 28 | 161 | 28 | 161 | 28 | 161 | 28 |
| 82 | 213 | 163 | 213 | 163 | 213 | 163 | 213 | 163 | 213 | 163 | 213 |
| 83 | 65 | 20 | 65 | 20 | 65 | 20 | 65 | 20 | 65 | 20 | 65 |
| 84 | 220 | 23 | 220 | 23 | 220 | 23 | 220 | 23 | 220 | 23 | 220 |
| 85 | 71 | 231 | 71 | 231 | 71 | 231 | 71 | 231 | 71 | 231 | 71 |
| 86 | 4 | 90 | 4 | 90 | 4 | 90 | 4 | 90 | 4 | 90 | 4 |
| 87 | 177 | 183 | 177 | 183 | 177 | 183 | 177 | 183 | 177 | 183 | 177 |
| 88 | 68 | 224 | 68 | 224 | 68 | 224 | 68 | 224 | 68 | 224 | 68 |
| 89 | 248 | 199 | 248 | 199 | 248 | 199 | 248 | 199 | 248 | 199 | 248 |
| 90 | 18 | 16 | 18 | 16 | 18 | 16 | 18 | 16 | 18 | 16 | 18 |
| Enhanced TDMRC Text → | ← | ö | # | / | p | ∏ | ! | ↕ | c | Z | 7 |

*PAC = 2*
*Master Key Value=11 26 32 67*
*Random Seed 1 = 3422*
*Random Seed 2 =845*

Figure 6.1 Encryption using Enhanced TDMRC Encryption Algorithm

In the above sample output, the plaintext GOD IS LOVE was encrypted. The Enhanced TDMRC ciphertext was ← Ö#/ p∏!↕ c Z7. Here we can see that though the character O repeats, first O is substituted with Ö and second O is substituted with c. Similarly the space character is also substituted with different characters. This is due to the Poly Alphabetic nature of Enhanced-TDMRC, here the PAC value is 2 thus the number of characters used simultaneously for encryption is 2.

Decryption was performed with same PAC, Random Seed and Random Series Values as in Encryption stage and is shown in Figure 6.2.



ASCII Values and its Corresponding values in Series

| Enhanced TDMRC Text → | ← | Ö | # | / | p | Π | ! | ↕ | c | Z | 7 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ASCII → | 27 | 148 | 35 | 92 | 112 | 20 | 33 | 18 | 99 | 90 | 55 |
| ASCII | Series 1 | Series 2 | Series 1 | Series 2 | Series 1 | Series 2 | Series 1 | Series 2 | Series 1 | Series 2 | Series 1 |
| 32 | 33 | 92 | 33 | 92 | 33 | 92 | 33 | 92 | 33 | 92 | 33 |
| 60 | 148 | 22 | 148 | 22 | 148 | 22 | 148 | 22 | 148 | 22 | 148 |
| 61 | 34 | 121 | 34 | 121 | 34 | 121 | 34 | 121 | 34 | 121 | 34 |
| 62 | 85 | 67 | 85 | 67 | 85 | 67 | 85 | 67 | 85 | 67 | 85 |
| 63 | 22 | 2 | 22 | 2 | 22 | 2 | 22 | 2 | 22 | 2 | 22 |
| 64 | 215 | 31 | 215 | 31 | 215 | 31 | 215 | 31 | 215 | 31 | 215 |
| 65 | 56 | 185 | 56 | 185 | 56 | 185 | 56 | 185 | 56 | 185 | 56 |
| 66 | 114 | 114 | 114 | 114 | 114 | 114 | 114 | 114 | 114 | 114 | 114 |
| 67 | 191 | 78 | 191 | 78 | 191 | 78 | 191 | 78 | 191 | 78 | 191 |
| 68 | 35 | 32 | 35 | 32 | 35 | 32 | 35 | 32 | 35 | 32 | 35 |
| 69 | 55 | 36 | 55 | 36 | 55 | 36 | 55 | 36 | 55 | 36 | 55 |
| 70 | 109 | 54 | 109 | 54 | 109 | 54 | 109 | 54 | 109 | 54 | 109 |
| 71 | 27 | 155 | 27 | 155 | 27 | 155 | 27 | 155 | 27 | 155 | 27 |
| 72 | 11 | 15 | 11 | 15 | 11 | 15 | 11 | 15 | 11 | 15 | 11 |
| 73 | 112 | 152 | 112 | 152 | 112 | 152 | 112 | 152 | 112 | 152 | 112 |
| 74 | 212 | 123 | 212 | 123 | 212 | 123 | 212 | 123 | 212 | 123 | 212 |
| 75 | 252 | 127 | 252 | 127 | 252 | 127 | 252 | 127 | 252 | 127 | 252 |
| 76 | 108 | 18 | 108 | 18 | 108 | 18 | 108 | 18 | 108 | 18 | 108 |
| 77 | 225 | 9 | 225 | 9 | 225 | 9 | 225 | 9 | 225 | 9 | 225 |
| 78 | 137 | 86 | 137 | 86 | 137 | 86 | 137 | 86 | 137 | 86 | 137 |
| 79 | 99 | 148 | 99 | 148 | 99 | 148 | 99 | 148 | 99 | 148 | 99 |
| 80 | 43 | 93 | 43 | 93 | 43 | 93 | 43 | 93 | 43 | 93 | 43 |
| 81 | 28 | 161 | 28 | 161 | 28 | 161 | 28 | 161 | 28 | 161 | 28 |
| 82 | 213 | 163 | 213 | 163 | 213 | 163 | 213 | 163 | 213 | 163 | 213 |
| 83 | 65 | 20 | 65 | 20 | 65 | 20 | 65 | 20 | 65 | 20 | 65 |
| 84 | 220 | 23 | 220 | 23 | 220 | 23 | 220 | 23 | 220 | 23 | 220 |
| 85 | 71 | 231 | 71 | 231 | 71 | 231 | 71 | 231 | 71 | 231 | 71 |
| 86 | 4 | 90 | 4 | 90 | 4 | 90 | 4 | 90 | 4 | 90 | 4 |
| 87 | 177 | 183 | 177 | 183 | 177 | 183 | 177 | 183 | 177 | 183 | 177 |
| 88 | 68 | 224 | 68 | 224 | 68 | 224 | 68 | 224 | 68 | 224 | 68 |
| 89 | 248 | 199 | 248 | 199 | 248 | 199 | 248 | 199 | 248 | 199 | 248 |
| 90 | 18 | 16 | 18 | 16 | 18 | 16 | 18 | 16 | 18 | 16 | 18 |
| Original Plain Text → | G | O | D |  | I | S |  | L | O | V | E |

Figure 6.2 Decryption using E-TDMRC Decryption Algorithm

## VII. Enhanced Randomness in E-TDMRC

### 7.1 Randomness in Master Key Generation

In both the algorithms Real Time Clock Value is used for Master key Generation. Since true random source is employed for both the algorithms, both the algorithms are equally strong in Master key generation technique being used.

### 7.2 Randomness in selecting Poly Alphabetic Coefficient (PAC) , P

In Enhanced TDMRC, NLCG is employed for selecting P. Due to the increased randomness of NLCG, Enhanced TDMRC performs better than TDMRC in selecting P randomly. Nesting ensures that PAC value is random and cryptographically secure.

### 7.3 Randomness in selecting P number of Sub Keys

NLCG is employed in selecting 4 digit Sub Key values. Use of NLCG in place of LCG will enhance the randomness of generated Sub Key values. Since repetition at regular interval does not occur in case of NLCG generated series are more random.

### 7.4 Randomness in generating P Number of Series

In TDMRC, LCG is employed for random series generation. NLCG when used in place of LCG for random series generation will enhance the behaviour of TDMRC. NLCG with refinement is used for random series generation.[5]

## VIII. Analysis of PAC values generated

The E-TDMRC that employs NLCG for its PAC value generation the series generated will have its period as infinity.

5 9 3 2 8 1 6 7 2 1 4 6 8 1 4 3 9 2 4 6 7 9 1 5 3 7 9 5 6 2

Figure 8.1 PAC Values generated by NLCG in E-TDMRC

From the series it can be seen that cycle never occurs. That is even though an element have repetitive appearance a sequence never repeats. Thus if a 6 is encountered repeatedly we cannot predict the next value occurring as we did in the case of TDMRC because of period occurrence.

## IX. Analysis of Sub Key values generated

The Enhanced TDMRC employs NLCG for its Sub Key value generation so the series generated will have its period as infinity.

2334 5672 3465 1876 4567 9234 5672 1564 8265 1435 1234 8746 4253 8743 1009 2435 7654 2345 9265 2341 8754 2435

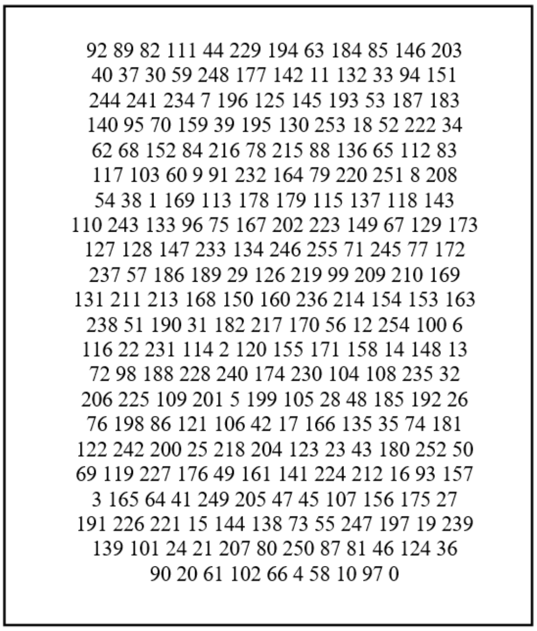
8655 2455 9865 2874 4536 1356 2846

Figure 9.1 Sub Key Values generated by NLCG in E-TDMRC

From the series it can be seen that cycle never occurs. That is even though an element have repetitive appearance a sequence never repeats. 5672 is encountered repeatedly in the series but we cannot predict the next value occurring as we did in the case of TDMRC.

## X. Analysis of Random Series Generated

Enhanced TDMRC was implemented and a sample random series is taken for result analysis. Statistical and Graphical study was done for analysing the randomness exhibited by the series generated by Enhanced TDMRC.



```
92 89 82 111 44 229 194 63 184 85 146 203
40 37 30 59 248 177 142 11 132 33 94 151
244 241 234 7 196 125 145 193 53 187 183
140 95 70 159 39 195 130 253 18 52 222 34
62 68 152 84 216 78 215 88 136 65 112 83
117 103 60 9 91 232 164 79 220 251 8 208
54 38 1 169 113 178 179 115 137 118 143
110 243 133 96 75 167 202 223 149 67 129 173
127 128 147 233 134 246 255 71 245 77 172
237 57 186 189 29 126 219 99 209 210 169
131 211 213 168 150 160 236 214 154 153 163
238 51 190 31 182 217 170 56 12 254 100 6
116 22 231 114 2 120 155 171 158 14 148 13
72 98 188 228 240 174 230 104 108 235 32
206 225 109 201 5 199 105 28 48 185 192 26
76 198 86 121 106 42 17 166 135 35 74 181
122 242 200 25 218 204 123 23 43 180 252 50
69 119 227 176 49 161 141 224 212 16 93 157
3 165 64 41 249 205 47 45 107 156 175 27
191 226 221 15 144 138 73 55 247 197 19 239
139 101 24 21 207 80 250 87 81 46 124 36
90 20 61 102 66 4 58 10 97 0
```

Figure 10.1 Random Series generated by NLCG in Enhanced TDMRC

## 10.1 Graphical Analysis

### 10.1.1 Bar Graph Analysis

Bar graph was plotted for the random series. In the case of Enhanced TDMRC the generated series has period infinity. Period is infinity for NLCG because the increment and the multiplicand value is never constant it forms the random number generated by a nested random series.

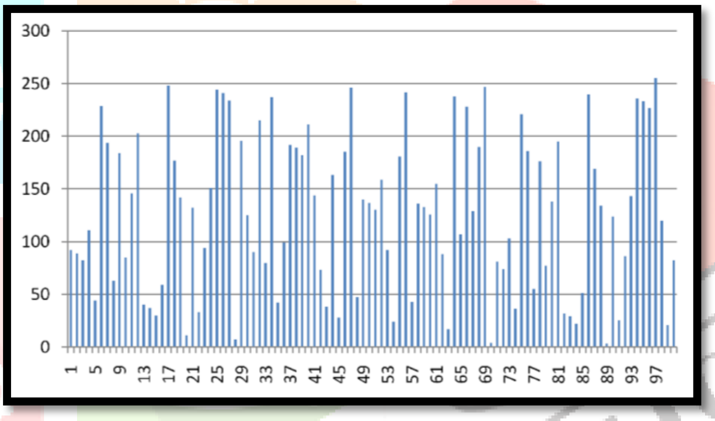


Figure 10.2 Bar Graph Analysis of Random Series generated by NLCG

### 10.1.2 Scatter Diagram Analysis

For analyzing first 30 elements are chosen from the NLCG series. Points on the graph are divided into four quadrants. If there are X points on the graph, Count X/2 points from top to bottom and draw a horizontal line. Count X/2 points from left to right and draw a vertical line. Here 30 points are considered so lines are drawn after 15 points and graph divided into four quadrants.

A = points in upper left + points in lower right
B = points in upper right + points in lower left
Q = the smaller of A and B
N = A + B

If Q is less than the limit, the two variables are related.
If Q is greater than or equal to the limit, the pattern could have occurred from random chance.
Now,

A=10, B=20, N=30, Q=10

From the Runs Test Table, limit=9,

Q>limit , Hence the test is passed

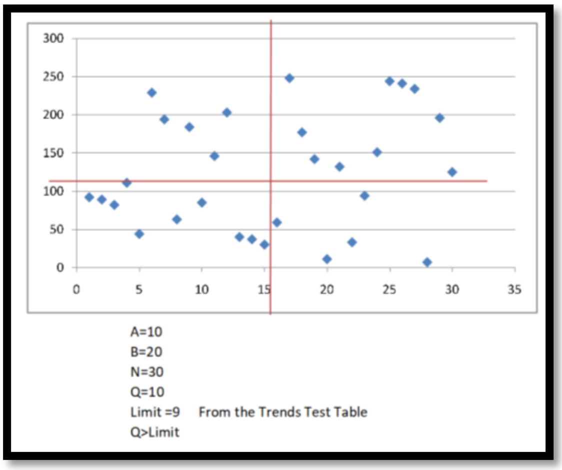


Figure 10.3 Scatter Diagram Analysis of NLCG generated Series

It is proved from the scatter plot analysis that the elements in both the series has no correlation. Further, Scatter diagram analysis proves that the series posses good randomness and that there is no linear relationship or correlation between the generated random values. Scatter diagram analysis proves that the values generated in both the series are by random chance.

## 10.2 **Statistical Analysis**

### 10.2.1 Kolmogorov Smirnov Test

The Kolmogorov-Smirnov test is a non-parametric test. It is ‘goodness of fit’ test to test the quality of Random Number Generators.

The test hypothesis is given below:

$H_0$ = Sequence being tested is random

$H_a$ = Sequence being tested is not random

$K^+$ = maximum observed deviation below the expected cdf $(\max(j/n - X_j))$

$K^-$ = minimum observed deviation below the expected cdf$(\max(X_j - (j - 1)/n)$

Table 10.1 K S Test analysis of Random Series generated by NLCG in Enhanced TDMRC

| J | Random numbers in sorted order, $X_j$ | Normalized $X_j$ / 256 | $(j/n - X_j)$ | $X_j$ -(j-1)/n |
|---|---|---|---|---|
| 1 | 7 | 0.02734375 | 0.0059896 | 0.0273438 |
| 2 | 11 | 0.04296875 | 0.0236979 | 0.0096354 |
| 3 | 30 | 0.1171875 | -0.0171875 | 0.0505208 |
| 4 | 33 | 0.12890625 | 0.0044271 | 0.0289063 |
| 5 | 37 | 0.14453125 | 0.0221354 | 0.0111979 |
| 6 | 40 | 0.15625 | 0.04375 | -0.0104167 |
| 7 | 44 | 0.171875 | 0.0614583 | -0.028125 |
| 8 | 59 | 0.23046875 | 0.0361979 | -0.0028646 |
| 9 | 63 | 0.24609375 | 0.0539063 | -0.0205729 |
| 10 | 82 | 0.3203125 | 0.0130208 | 0.0203125 |
| 11 | 85 | 0.33203125 | 0.0346354 | -0.0013021 |
| 12 | 89 | 0.34765625 | 0.0523438 | -0.0190104 |
| 13 | 92 | 0.359375 | 0.0739583 | -0.040625 |
| 14 | 94 | 0.3671875 | 0.0994792 | -0.0661458 |
| 15 | 111 | 0.43359375 | 0.0664063 | -0.0330729 |
| 16 | 125 | 0.48828125 | 0.0450521 | -0.0117188 |
| 17 | 132 | 0.515625 | 0.0510417 | -0.0177083 |
| 18 | 142 | 0.5546875 | 0.0453125 | -0.0119792 |
| 19 | 146 | 0.5703125 | 0.0630208 | -0.0296875 |
| 20 | 151 | 0.58984375 | 0.0768229 | -0.0434896 |
| 21 | 177 | 0.69140625 | 0.0085937 | 0.0247396 |
| 22 | 184 | 0.71875 | 0.0145833 | 0.01875 |
| 23 | 194 | 0.7578125 | 0.0088542 | 0.0244792 |
| 24 | 196 | 0.765625 | 0.034375 | -0.0010417 |
| 25 | 203 | 0.79296875 | 0.0403646 | -0.0070313 |
| 26 | 229 | 0.89453125 | -0.0278646 | 0.0611979 |
| 27 | 234 | 0.9140625 | -0.0140625 | 0.0473958 |
| 28 | 241 | 0.94140625 | -0.0080729 | 0.0414063 |
| 29 | 244 | 0.953125 | 0.0135417 | 0.0197917 |
| 30 | 248 | 0.96875 | 0.03125 | 0.0020833 |

$K^+ = 0.0994792$

$K^- = 0.0611979$

From KS test table at n=30 and 1-α=0.9

K = 0.21756

$K^+ < K$ and $K^- < K$ hence sequence generated by NLCG is random and pass KS test

### 10.2.2 **Runs Test**

A series of increasing or decreasing values is called a run. Number of increasing or decreasing values defines the length of the run. For starting the runs test median of first thirty elements are found out. If a value in the series is less than the median then it is denoted by -1 otherwise +1. After forming the series of +1 and -1, runs of +1 and -1 are counted and hypothesis testing is done.

$H_0$: Sequence is random

$H_a$ : Sequence is not random

Table 10.2 Runs Test analysis of Random Series generated by NLCG

| Random Series NLCG | Value>median +1 else -1 |
|---|---|
| | Median =118 |
| 92 | -1 |
| 89 | -1 |
| 82 | -1 |
| 111 | -1 |
| 44 | -1 |
| 229 | 1 |
| 194 | 1 |
| 63 | -1 |
| 184 | 1 |
| 85 | -1 |
| 146 | 1 |
| 203 | 1 |
| 40 | -1 |
| 37 | -1 |
| 30 | -1 |
| 59 | -1 |
| 248 | 1 |
| 177 | 1 |
| 142 | 1 |
| 11 | -1 |
| 132 | 1 |
| 33 | -1 |
| 94 | -1 |
| 151 | 1 |
| 244 | 1 |
| 241 | 1 |
| 234 | 1 |
| 7 | -1 |
| 196 | 1 |
| 125 | 1 |

From the sequence generated by NLCG, 30 samples are taken and median is calculated. Median is got as 118. Now all the values greater than 118 is denoted as +1 and values less than 118 as -1.

Number of runs is got as 14. Now, $n_1$, number of -1, is 15 and $n_2$ , number of +1 is 15. From runs table the test is passed if the number of runs is between 10 and 22. Here number of runs is 14 and the NLCG generated sequence posses the property of randomness.

## XI. Cryptanalysis of E-TDMRC

Depending upon the level of information a cryptanalyst have, cryptanalysis can be of three types.

i. Ciphertext Only Attack – Here the attack is solely based on the Ciphertext which is the only information that a Cryptanalyst have

ii. Known Plaintext Attack – Here the cryptanalyst have a Plaintext and its corresponding Ciphertext as his information.

iii. Chosen Plaintext Attack – Here the attack is based on a Chosen Plaintext and its corresponding Ciphertext.

In the first case the attack is not possible because Exhaustive Search Method is not practically possible in case of Enhanced TDMRC. At any instant Ciphertext will have a combined complexity of Random Seed Values which are generated based on Real Time Clock value and a randomly chosen Pixel Value, Poly Alphabetic Coefficient, Random Number Generation Technique and Nesting. [5]

Quantitatively in TDMRC as already mentioned Master key is an 8 digit number derived from the Real Time Clock Value with 8640000 possible values. As per the algorithm the first seed value is obtained by multiplying the first Sub Key Value with Master Key Value. Sub Key Value is having 1000 possible values. Thus the possible number of seed value for first code generation is 864 x $10^7$. Similarly there are 864 x $10^7$ possible values for each code generation. [5]

In the case of Enhanced TDMRC in addition to the real time clock value, intensity value of a pixel randomly chosen from the image is also considered. Thus the possible number of seed values for each code generation is 864 x $10^7$ x $256^t$ , where t is the number of pixels in the image and each pixel have 256 possible intensity values.

In the case of Enhanced TDMRC Poly Alphabetic Coefficient is also randomly chosen.

Let it be P then the possible number of keys are (864 x $10^7$ x $256^t$) $^P$.

Thus from the above discussion it is clear that cryptanalysis is not possible in Ciphertext only situation.

Poly Alphabetic and Random nature of Enhanced TDMRC ensures Brute Force Attack also is not possible.

The other two classifications of Cryptanalysis Known Plaintext Attack and Chosen Plaintext Attack will not be a threat to Enhanced TDMRC. This is because the Ciphertext generated for the same Plaintext will be different at different instant due to the Poly Alphabetic Nature of Enhanced TDMRC.

Enhanced TDMRC is thus cryptographically very secure and cryptanalysis is rather impossible.

## XII. Computation Time Analysis

When Enhanced TDMRC Code is used for encryption, the codes corresponding to data is generated in advance and hence there is no time delay at the moment of encryption. But when we consider the case of other encryption methods the complex calculation is done on each block of data at the time of encryption which gives way to time delay making it not suitable for high speed real time application.[5]

## XIII. Conclusion

By substituting LCG with NLCG performance improvement of TDMRC was made possible which gave way to the development of Enhanced TDMRC. When nesting was introduced into traditional LCG the concept of period was removed from the generated series that is the period was made infinity. True random source, prime factorization problem and nesting all together contributes to the improved performance of NLCG and in turn improved performance of Enhanced TDMRC. The statistical and graphical tests conducted concluded the enhanced behavior of Enhanced TDMRC. The computation time analysis also shows that E-TDMRC is suitable for high speed real time application. Thus to sum up the study has proved Enhanced TDMRC to be apt for cryptographic and other security related activities.

## References

[1]. Harshavardhan Kayarkar, Sugata Sanyal, "Classification of Various Security Techniques in Databases and their comparative analysis", ACTA TECHNICA CORVINIENSIS –Bulletin of Engineering, Fascicule 2 [April–June],2012,pp. 135-138

[2]. The Art of Computer Programming - Donald E Knuth – Vol 2- Addison Wesley Publication – 1968

[3]. Introduction to Cryptography with Coding Theory – Wade Trappe and Lawrence C. Washington

[4]. Cryptography and Network Security Principles and Practices, Fourth Edition. By William Stallings. Publisher: Prentice Hall. Pub Date: November 16, 2005.

[5]. "Data Security in Fault Tolerant Hard Real Time Systems, Use of Time Dependant Multiple Random Cipher Code", A thesis submitted by Varghese Paul in partial fulfillment of the requirements for the degree of Doctor Of Philosophy of Cochin University of Science and Technology

[6]. Kinga Marton, Alin Suciu, Losif Ignat, "Randomness in Digital Cryptography: A Survey" Romanian Journal Of Information Science and Technology ,Volume 13, Number 3, 2010, 219–240

[7]. Norm Matloff, "Random Number Generation",February 21, 2006

[8]. Harald Niederreiter, "Random Number Generation and QuasimMonte Carlo Methods", Society for Industrial and Applied Mathematics, Philadelphia, 1992.

[9]. "Linear Congruential Generators" by Joe Bolte, Wolfram Demonstrations Project.

[10]. "True Random Number Generators Secure in a Changing Environment", Boaz Barak, Ronen Shaltiel, and Eran Tromer, Department of Computer Science and Applied Mathematics, Weizmann Institute of Science , Rehovot, ISRAEL

[11]. Adi A. Maaita, Hamza A. A. Al_Sewadi, "Deterministic Random Number Generator Algorithm for Cryptosystem Keys", International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:9, No:4, pp 972-977, 2015

[12]. Antu Annam Thomas and Varghese Paul, "Performance Enhancement of Cryptographic Algorithms by Increasing Randomness through Nesting In Random Number Generators", International Journal of Scientific Research in Computer Science, Engineering and Information Technology, Volume 2, Issue 5, May 2017, pp. 203- 209

[13]. Antu Annam Thomas and Varghese Paul, "Nested Random Number Generator", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 7, Issue 5, May 2017, pp.767-773