# Effects of Parameters of Enhanced A5/1

Nikesh Bajaj
Lovely Professional University
Electronics and Communication Department
Phagwara-144402, India

## ABSTRACT
The Global System for Mobile communication, GSM voice calls are encrypted using a family of algorithms collectively called A5. A5/1 is the stream cipher which encrypts the information transmitted from mobile user. Initially A5 algorithm was kept secret to ensure the security but as algorithm was disclosed many cryptanalytic attacks were proposed and proved the A5 algorithm cryptographically weak. In this paper, proposed enhanced A5/1 is described and it's analysis with different parameters is done. Enhanced A5/1 is proposed to make it robust and resistive to the attacks. Modification is done in two ways (1) feedback tapping mechanism which is enhanced by variable taps for LFSR (Linear Feedback Shift Register) and random shuffling of LFSRs, which increases the complexity of the algorithm without compromising the properties of randomness and (2) clocking rule. The modification has been proposed keeping the ease of implementation in mind. This modified algorithm has been simulated in MATLAB and tested its randomness properties by 'Randomness test suit' given by NIST-National Institute of Standard and Technology and obtained satisfactory results. Further analysis of A5/1 is done by varying it's parameters to achieve better results.

## General Terms
GSM Security, Stream Cipher, Cryptography.

## Keywords
A5/1, GSM security, stream cipher; randomness tests.

## 1. INTRODUCTION
With the current development of telecommunication and network technology, the mobile communication is becoming the important part of it. GSM uses the encryption algorithms collectively called as A5 family. A5/0 is no encryption. A5/1 is the "standard" encryption algorithm, while A5/2 is the "export" (weakened) algorithm. A5/3 is a new algorithm based on the UMTS/WCDMA algorithm Kasumi. All of these algorithms use a 64-bit key [1]. A5 is a stream cipher. Stream ciphers are symmetric-key ciphers that generate pseudo-random binary sequences which are used to encrypt the message signals on bit-by-bit basis. The encryption and decryption in stream ciphers is based on XOR operation. The strength and security of these ciphers depends upon the characteristics of bit sequences produced by the stream generator [2]. Research studies and analysis has shown that A5/1 has some weaknesses which lead A5/1 into cryptographic attacks. One of the weaknesses of A5/1 is fixed feedback polynomial of LFSRs and other is the weak clocking mechanism. A5/1 was initially cryptanalyzed by Golic [3] when only a rough outline of A5/1 was leaked. After releasing of A5/1 algorithm, it was cryptanalized by Biryukov, Shamir, and Wagner [4]; Biham and Dunkelman [5]; Ekdahl and Johansson [6]; Maximov, Johansson and Babbage [7] and recently by Barkan and Biham [8]. Most of the attacks against A5/1 are known plaintext attacks and use security weaknesses in the clock-controlling unit [9]. In this paper, feedback polynomial of LFSR used in A5/1 is replaced by variable feedback polynomial. The variation in the feedback polynomial is again randomized to maintain the randomness of sequence generated and complexity of algorithm. Further the shuffling of LFSRs is also introduced to strengthen the cipher to withstand against Berleykamp Massey attack [10] and some other attacks. The modification is also done in clocking mechanism of the A5/1. Though there are many modification and enhancement proposals but some of them are too complex to realize. The modification of algorithm is easy to realize. The proposed scheme has been coded and simulated in MATLAB R2010a.

The rest of paper is organized as follows. In section II, A5/1 stream cipher is described. Section III discusses the modified scheme of A5/1. In section IV, the results obtained by NIST's randomness test suit [11] and other analysis is discussed. Finally, section V concludes the work.

## 2. A5/1 STREAM CIPHER
A5/1 is a stream cipher used in GSM standard to encrypt the information of the mobile user [12]. GSM mobile information is transmitted as sequences of frames with the frame rate of 217 (frames per seconds) approximately, i.e. one frame is transmitted at every 4.6 milliseconds. The frame length is 228 bits, 114 bits for the communication in each direction. A5/1 is used as a key stream generator and produces 228 bits for each frame which are XORed with the frame bits. A5/1 is initialized using a 64-bit secret key together with a publicly-known 22-bit frame number. The A5/1 stream cipher is based on three linear feedback shift registers (LFSRs), R1, R2 and R3 of lengths 19, 22 and 23 bits respectively. The circuit diagram is shown in Figure 1. Three feedback polynomials used for LFSR R1, R2 and R3 are: x19 + x18 + x17 + x14 + 1, x22 + x21 +1 and x23 + x22 + x21 + x8 +1 respectively. Each LFSR is clocked cycles that depend upon majority rule [13]. Majority rule uses three clocking bits b1, b2 and b3 of LFSR R1, R2 and R3 respectively and determines the value of majority m using m = maj(b1, b2, b3). The majority rule is simply the majority among these bits, if two or more are 1 then the value of majority m is 1. Similarly, if two or more are 0 then majority m is 0. Now if bi = m then Ci will be 1 else Ci will be 0, and if Ci is 1 then register Ri will be clocked (shifted), where i=1, 2, 3. This means that if clocked bit of any LFSR is in majority then that LFSR will clocked. Thus the probability of an individual LFSR being clocked is 3/4. At each clocking, each LFSR generates one bit xi(t). All three bits are than XORed together to generate the final output bit z(t),

which is defined as z(t) = x1⊕x2⊕x3. This linear combination produces the output key stream z(t).

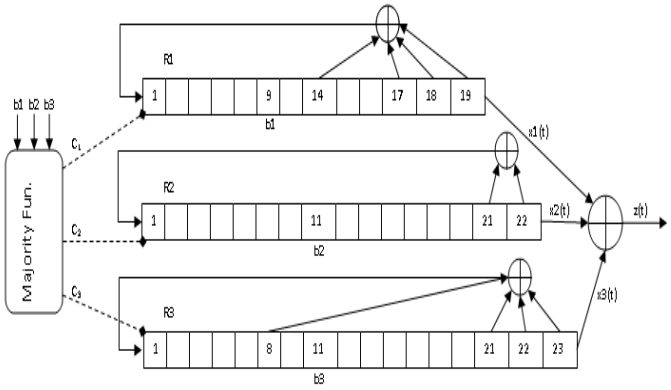



Fig 1: A5/1 Stream Cipher

## 3. PROPOSED MODIFICATIONS IN A5/1

In the conventional A5/1, two modifications have been proposed; one is in feedback tapping unit and other modification is in the clocking rule. The architecture of the proposed scheme is shown in Figure 2.

### 3.1 Feedback and State selection unit

In the modified A5/1, the feedback unit is modified in two different ways; each is described in this sub section.

#### *3.1.1 Shuffling LFSRs:*

The first modification that can be seen in figure 2 is that each LFSR is extended to 23 bit long, i.e. R1 LFSR which was initially 19 bit long is extended to 23 bit long LFSR by adding 4 bit shift register. Similarly R2 is extended to 23 bit long from 22 bit length by adding one bit shift register and R3 is already 23 bit long. The purpose of extending all LFSRs to 23 bit is to shuffle these LFSR. For example for an instant LFSR R1 can be used as 22 bit LFSR, R2 as 23 bit LFSR and R3 as 19 bit LFSR. These LFSRs can be permuted that means at any instant there should be one 19 bit, one 22 bit and one 23 bit LFSR to have basic A5/1 architecture. The shuffling of the LFSRs is done periodically and for the entire one period, fixed length of LFSRs will be used to generate the output stream sequence. Though shuffling is done periodically but the state of LFSR will change randomly, for example state of R1 (19 bit, 22 bit or 23 bit) will be chosen randomly. Let's consider initially R1, R2 and R3 is 19 bit, 22 bit and 23 bit respectively. After period of Ts (at Ts+1 instant), the states of R1, R2 and R3 are changed and R1, R2 and R3 will be 23 bit, 22 bit and 19 bit LFSR, and will remain same for next Ts period. Next shuffling will be at 2Ts+1 instant so on and so forth. This can be conclude as shuffling will be at nTs+1 instant, where n=1, 2, 3. . . As previously stated that shuffling is done periodically but state of R1, R2 and R3 will change randomly. As there are three states for each LFSR, there will be six different permutation or states are possible, which requires at least three bits to select one combination of three states. Considering these three bits as tc1, tc2 and tc3 together as Tc = [tc1 tc2 tc3]. As there are 8 possible combinations exist for these three bits Tc, whereas only 6 combinations are required, for this, two most likely permutations can be reused. This selection of stats is shown in Table1.

As shown in Table 1 three Tc bits will select the states of R1, R2 and R3 LFSRs, for example if Tc=[1 0 0] at shuffling instant, at nTs+1, then R1, R2 and R3 will be 22 bit, 23 bit and 19 bit LFSR respectively. This State selection of R1, R2 and R3 is implemented by using MUXs. As shown in Figure 2, the output of Ri LFSR which is xi(t) is selected by a MUX which has two bits si and ti where i=1, 2 and 3, when an Ri is 19 bit LFSR si and ti become 0 and 1 respectively and xi(t)= Ri (19). Similarly for 22 bit LFSR, si and ti become 1 and 0 respectively and xi(t) = Ri (22) and so on. It can be noted that si and ti never become [0 0], that's why one connection of MUX is left with no connection. As output is taken from 19th, 22nd or 23rd bit of LFSR that depend on si and ti (state of Ri) the resulting feedback should also depend on same si and ti. Figure 3 shows

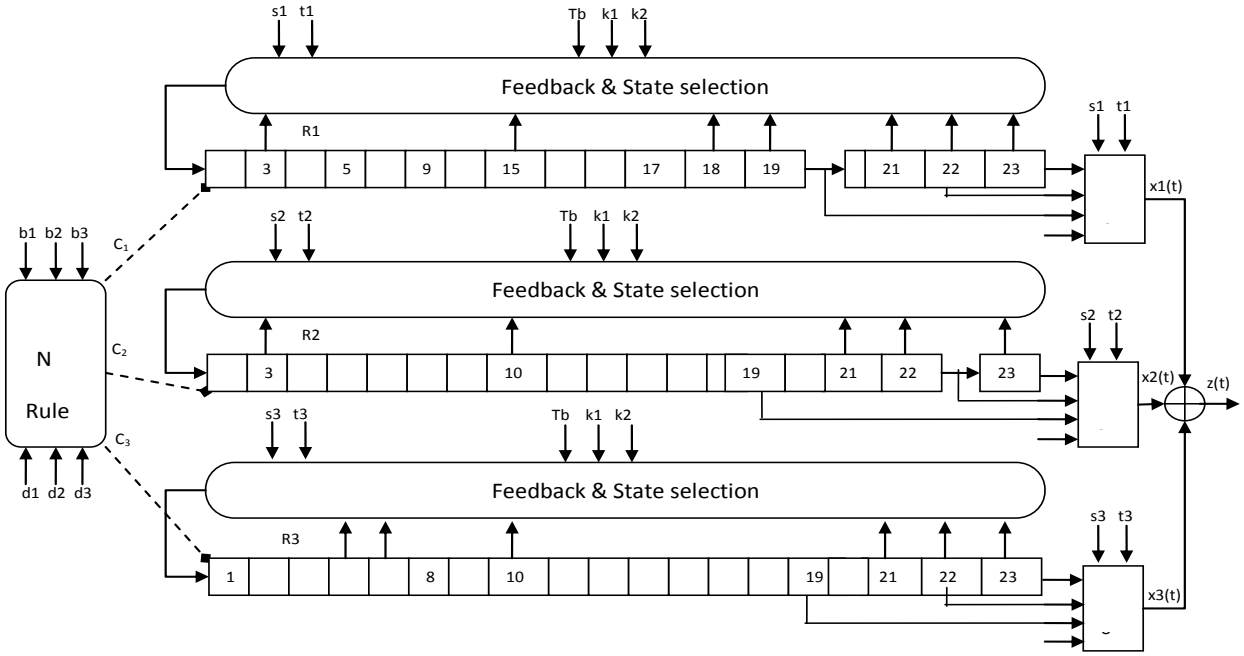



Fig 2: Proposed modified A5/1.

the internal architecture of feedback and selection block; here only internal architecture of R1is shown to describe the feedback selection according to state of R1.

**TABLE 1**

**Logic of State selection of R1, R2 and R3 LFSR**

| Tc=[tc1 tc2 tc3] | States of LFSRs | | | Bits to select the state | | |
|---|---|---|---|---|---|---|
| | R1 | R2 | R3 | s1 t1 | s2 t2 | s3 t3 |
| 0 0 0* | - | - | - | - | - | - |
| 0 0 1 | 19 | 22 | 23 | 0 1 | 1 0 | 1 1 |
| 0 1 0 | 19 | 23 | 22 | 0 1 | 1 1 | 1 0 |
| 0 1 1 | 22 | 19 | 23 | 1 0 | 0 1 | 1 1 |
| 1 0 0 | 22 | 23 | 19 | 1 0 | 1 1 | 0 1 |
| 1 0 1 | 23 | 19 | 22 | 1 1 | 0 1 | 1 0 |
| 1 1 0 | 23 | 22 | 19 | 1 1 | 1 0 | 0 1 |
| 1 1 1* | - | - | - | - | - | - |

As shown in Figure 3, all feedbacks (for 19 bit, 22 bit and 23 bit LFSR's feedback) are used to calculate a bit, f(t) to be feeded back to LFSR. At any instant three bits f1(t), f2(t) and f3(t) are calculated by taps of 19 bit, 22 bit and 23 bit LFSR respectively, but a bit f(t) is selected according to the present state of LFSR, i.e. according to s1 and t1, for example if R1 is in state of 22 bit LFSR then s1 and t1 will be 1 and 0, that will cause to MUX Mf (MUX used for selecting feedback bit) to select f(t)=f2(t) and MUX Mx (MUX used for selecting the output bit) x1(t)=R1(22).

### *3.1.2 Varying Feedback polynomials:*

In the conventional A5/1, the LFSRs have fixed feedback tap polynomial, that is for R1, R2 and R3 LFSRs the feedback taps are are: x19 + x18 + x17 + x14 + 1, x22 + x21 +1 and x23 + x22 + x21 + x8 +1 respectively. In our proposal multiple feedback polynomials are used for an LFSR. Changing the feedback taps was used in [14]. Four different feedback polynomials for each LFSR have been selected that are shown in Table 2.

The feedback polynomials are selected such that there would be only one tap different in all tap configurations of an LFSR, for example for 19 bit LFSR three tap position of all four taps are common i.e. (19 18 15) and only last tap position for each configuration is different. This will help to realize the circuit easily. Figure 3 shows the feedback connections for each state of LFSR that is grouped separately, it also shows the connections for all feedback taps which are required for any state of LFSR, for example, if R1 is in the state of 19 bits LFSR, than the feedback unit must be connected to all possible tap positions i.e. (19 18 15 13 12 10 3), as it has already mentioned that for ease of realization feedback polynomial are selected in such a manner that first three tap position is common only last tap is different. In Figure 3, common taps are shown by solid line where as the optional taps are shown by dashes line. Further, the internal connections of the feedback unit are shown in Figure 4; Figure 4(a) shows the feedback unit of 23 bit LFSR and Figure 4(b) shows the feedback unit of 19 bit LFSR. As there are 4 different feedback polynomials, two bits are needed to select one polynomial out of four. All four optional taps are connected to a

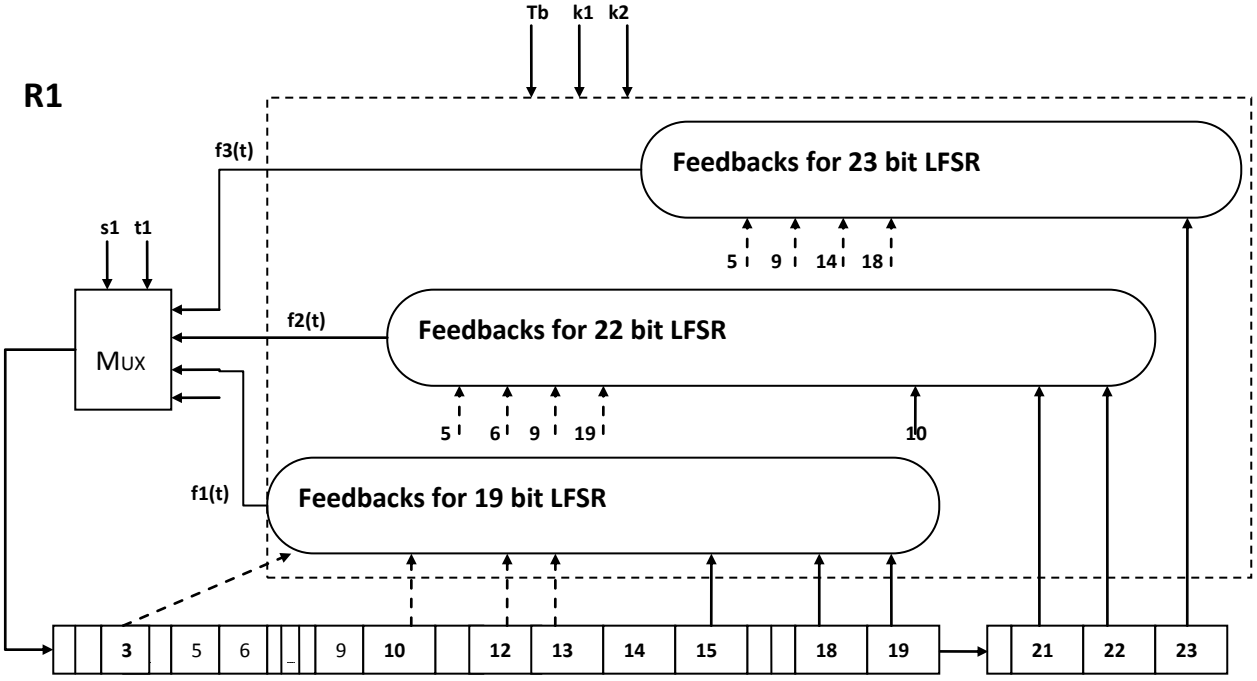



Fig 3: Internal architecture of feedback unit of LFSR R1

MUX and two selecting bits, k1 k2 are used to select one tap out of four taps.

**TABLE 2**

**Feedback Polynomials of LFSR used**

| Length of LFSR | Feedback Polynomials used |
|---|---|
| 19 bit LFSR | $x^{19} + x^{18} + x^{15} + x^{13} + 1$ |
| | $x^{19} + x^{18} + x^{15} + x^{12} + 1$ |
| | $x^{19} + x^{18} + x^{15} + x^{10} + 1$ |
| | $x^{19} + x^{18} + x^{15} + x^{3} + 1$ |
| 22 bit LFSR | $x^{22} + x^{21} + x^{10} + x^{19} + 1$ |
| | $x^{22} + x^{21} + x^{10} + x^{9} + 1$ |
| | $x^{22} + x^{21} + x^{10} + x^{6} + 1$ |
| | $x^{22} + x^{21} + x^{10} + x^{5} + 1$ |
| 23 bit LFSR | $x^{23} + x^{18} + 1$ |
| | $x^{23} + x^{14} + 1$ |
| | $x^{23} + x^{9} + 1$ |
| | $x^{23} + x^{5} + 1$ |

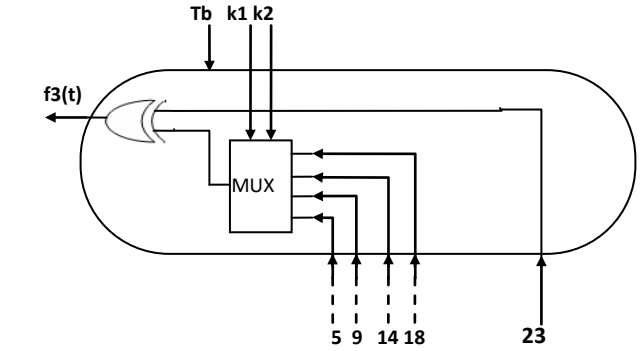
Again this feedback changing for an individual LFSR is a periodic. The period for changing the feedbacks of an individual has been chosen to make an LFSR cryptographically strong. As the Berleykamp-Massey attack requires 2*LC (LC linear complexity) bits of output binary sequence produced by a stream cipher in order to construct an LFSR of length LC that generates the same sequence [15]. So for changing the feedback polynomial it is chosen to change the feedback polynomial of an LFSR after it generates twice bits than its length. For this purpose a time vector Tb has been defined which will determine the timing instant when an LFSR needs to change the feedback taps. After initialization of the generator, first LFSR (19 bit) will change its feedback taps after 2*19=38 instants, i.e. at (38+1)th instant, and second LFSR (22 bit) will change its feedbacks after 44 instant of changing the first one, that is at (44+38+1)th instant. Similarly the third one will change its feedback at (46+44+38+1)th instant. Next, the first LFSR will change at (46+44+2*38+1)th instant, so on and so forth. For this to be

**TABLE 3**

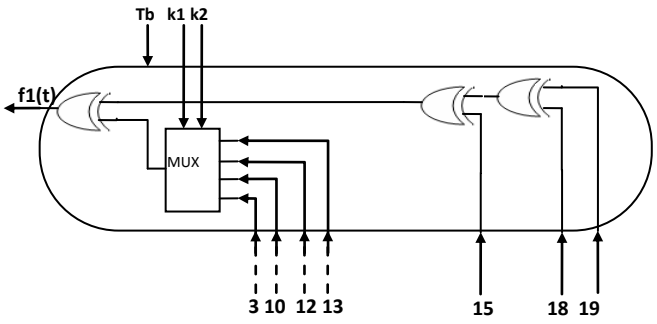
**The values of Time vector as function of time**

| At Time instant (i) | Time vector Tb |
|---|---|
| 39+128*j | [1 0 0] |
| 83+128*j | [0 1 0] |
| 129+128*(j-1) | [0 0 1] for j>1 |
| else | [0 0 0] |

accomplished time vector Tb is defined, if Tb = [1 0 0] than first LFSR will only change its feedback taps, if Tb= [0 1 0] then second, and for Tb= [0 0 1] third LFSR will change its feedback tap configuration. Time vector Tb is generalized as shown in Table 3, where time instant (i) is time which start from 0 and increases by one at every instant, where as j is defined as, j=0 for i<128, j=1 for i≥128 and i<256, and so on, j increases by one after every 128 instants. For Tb=[0 0 0] all LFSR will continue with their previous tap configuration. Though feedback changing instant has been defined periodically for an LFSR but again the selection for new tap for an LFSR depends on the value of k1 and k2. It can be noted that at any instant; k1. k2 and time vector Tb is same for all LFSRs, because at any instant only one LFSR will change its feedback taps.




(a) Feedback unit for 23 bit LFSR




(b) Feedback unit for 19 bit LFSR

Fig 4: Circuit diagram of feedback selection unit of LFSR R1

## 3.2 Clocking unit

The clock controlled unit of conventional A5/1 works on majority rule as discussed in section 2. That clocking mechanism has weakness that the probability that any LFSR will clocked (shifted) is ¾ which is 75% probability that any LFSR (R1, R2 or R3) will be clocked, whereas for randomness it should be 50%. For this new clocking rule has been introduced named as N-rule, this rule takes input as b1, b2, b3, d1, d2 and d3 and determine the clock outputs C1, C2 and C3; as shown in Figure 2. Now, if Ci is 1; Ri will be clocked and if Ci is 0; Ri will not be clocked, where i=1, 2, 3. If all clock outputs are 0 than again all LFSRs will be clocked. This is to avoid bit rate of output sequence be effected. The connections of bi and di are taken as follows; b1 =R1(19), b2 = R2(22), b3 =R3(23), and di =Ri(1). The clock outputs are as follows:

$$C_1 = b_1 \oplus d_3$$

$$C_2 = b_2 \oplus d_1$$

$$C_3 = b_3 \oplus d_2$$

The connections are selected such as to give better results. This improves the randomness of clocking, as there are 4 possible states out of 7, when an LFSR will be clocked. The probability of an LFSR to be clocked is now 4/7 which is approximately 57%, which is better than that of 75%.

## 3.3 Parameters of Enhanced A5/1

In the last two subsections of modifications, three parameters come into the picture. These are (1) Ts period of state changing, (2) Tc=[tc1 tc2 tc3], vector of three bits which selects the shuffling of the LFSRs and (3) k1, k2, that causes the selection of tapings for all LFSRs. These parameters need to be quantified carefully so that to achieve the better results, that includes obtaining the results and comparing with others. Out of these parameters shuffling period Ts has been varied and obtained results are analyzed in next section. Other two parameters Tc and k1, k2 are considered fixed.

# 4. RESULTS AND ANALYSIS

The aim behind the enhancements of the A5/1 encryption algorithm used in GSM standard is to increase its algorithm complexity without compromising the properties of randomness. The randomness of the generator can be tested by 'NIST randomness test suit'.

## 4.1 Effect of varying Shuffling period

In this subsection, effect of varying shuffling period has been analyzed against randomness of generated stream. Parameters, Tc and k1,k2 are fixed for this testing and are considered as tc1=R1(19), tc2=R2(22) and tc3=R3(23) for Tc=[tc1 tc2 tc3] and k1=R1(9)⊕ R2(11), k2=R2(11)⊕ R3(11). Parameter Ts, shuffling period of LFSR's states has been varying with values of 128, 256, 512 and 1024. This implies that if Ts=128 than all three LFSRs R1 R2 and R3 will change their stats after every 128$^{th}$ instants and will remain same for entire period of 128. Purpose of varying this parameter is to check that how shuffling of LFSRs effect randomness of generated stream. Should LFSRs shuffling period be less, to shuffle very frequently or more to shuffle moderately?

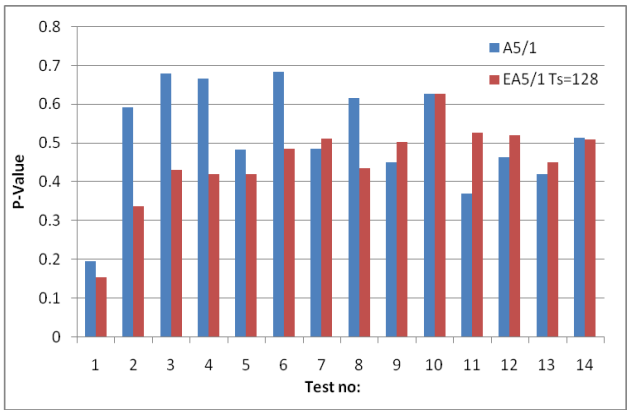



Fig 5: Comparison of P-values of A5/1 and EA5/1 with Ts=128

NIST test suit suggests that it is not enough to check the randomness of a stream generated by a generator, but several different streams should be generated and tested for randomness. Ten different bit streams have been generated from each generator. Length of the streams is 10000 bits. Each stream has

**TABLE 4**

**NIST Test Results**

| Test no: | Test | P-value of test | |
|---|---|---|---|
| | | Conventional A5/1 | EA5/1 with Ts=128 |
| 1 | Approximate Entropy Test | 0.195212 | 0.1558165 |
| 2 | Block Frequency Test | 0.5937233 | 0.3380135 |
| 3 | Commutative Sum (F) | 0.680391 | 0.4326668 |
| 4 | Commutative Sum (R) | 0.6667465 | 0.4209143 |
| 5 | FFT Test | 0.4835765 | 0.4196997 |
| 6 | Frequency Test | 0.6839889 | 0.4855879 |
| 7 | Linear Complexity | 0.4866099 | 0.5112491 |
| 8 | Longest Run of Ones | 0.6175319 | 0.4368533 |
| 9 | Overlapping Template on all ones Test | 0.4515866 | 0.5035248 |
| 10 | Rank Test | 0.6282948 | 0.6278368 |
| 11 | Run Test | 0.3704127 | 0.5285336 |
| 12 | Serial Test 1 | 0.463369 | 0.5212576 |
| 13 | Serial Test 2 | 0.4199995 | 0.4517657 |
| 14 | Non-periodic Template Test | 0.5136464 | 0.5111576 |

**TABLE 5**

**NIST Test Results**

| Test no: | P-value of test | | |
|---|---|---|---|
| | EA5/1 with Ts=256 | EA5/1 with Ts=512 | EA5/1 with Ts=1024 |
| 1 | 0.2118699 | 0.0790702 | 0.1234753 |
| 2 | 0.536233 | 0.4030937 | 0.6491708 |
| 3 | 0.3419494 | 0.5646854 | 0.6031661 |
| 4 | 0.4030039 | 0.5448438 | 0.7422541 |
| 5 | 0.4636885 | 0.4749704 | 0.5505559 |
| 6 | 0.3947158 | 0.5677295 | 0.6081831 |
| 7 | 0.6557424 | 0.6755905 | 0.524217 |
| 8 | 0.4322986 | 0.4616398 | 0.5523193 |
| 9 | 0.411355 | 0.4508905 | 0.5380101 |
| 10 | 0.5180858 | 0.5606459 | 0.4427349 |
| 11 | 0.4657274 | 0.3765621 | 0.3880858 |
| 12 | 0.4505824 | 0.4694956 | 0.5850065 |
| 13 | 0.5857059 | 0.4780418 | 0.6506241 |
| 14 | 0.516975795 | 0.512445096 | 0.492966115 |

been tested with NIST test suit and P-values of ten streams for same test are averaged and tabulated. First shuffling period Ts has been considered 128. Table 4 shows the results of NIST test suit for conventional A5/1 and enhanced A5/1 (EA5/1). This can be observed from Table 4 that EA5/1 passes all the random tests as P-value of any test is not below 0.001. Figure 5 is the plot of averaged P-values of ten streams for A5/1 and EA5/1 with shuffling period of 128 versus tests. In figure 5 x-axes represent test number which is corresponding to Table 4. It can be observed from figure 5 that EA5/1 with Ts =128 has more P-value for some tests e.g. test no: 7, 9, 11, 12 and 13 and has less P-value for some tests e.g. test no: 1, 2, 3, 4, 5, 6, 8 and 14.

Results are obtained for enhanced A5/1 (EA5/1) with shuffling period Ts = 256, 512 and 1024. Again ten streams have been generated for each case with stream length of 10000 bits. P-values of each test are averaged and tabulated in Table 5. In Table 5 test numbers are corresponding to Table 4. Figure 6, 7 and 8 are comparative plots of conventional A5/1 with EA5/1 with Ts equal to 256, 512 and 1024 respectively.

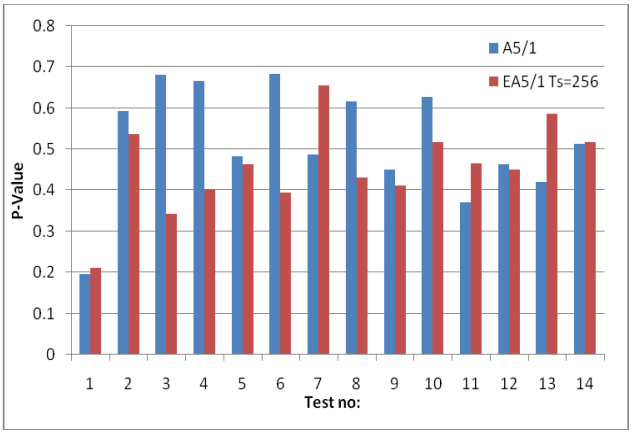



Fig 6: Comparison of P-values of A5/1 and EA5/1 with Ts=256

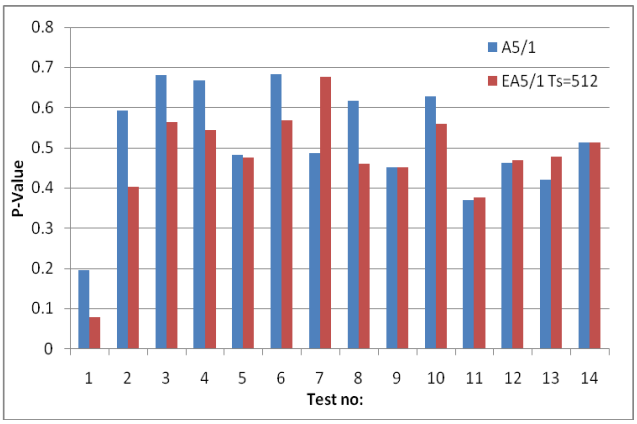



Fig 7: Comparison of P-values of A5/1 and EA5/1 with Ts=512

It can be observed from figure 6, 7 and 8 that shuffling period of 1024 is better in comparison to 128, 256 and 512. In figure 8 graph shows that P-values of many tests for EA5/1 are more than conventional A5/1. It also can be observed that P-value is

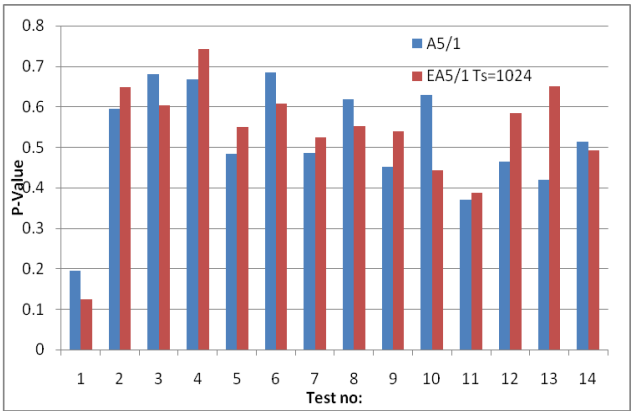



Fig 8: Comparison of P-values of A5/1 and EA5/1 with Ts=1024

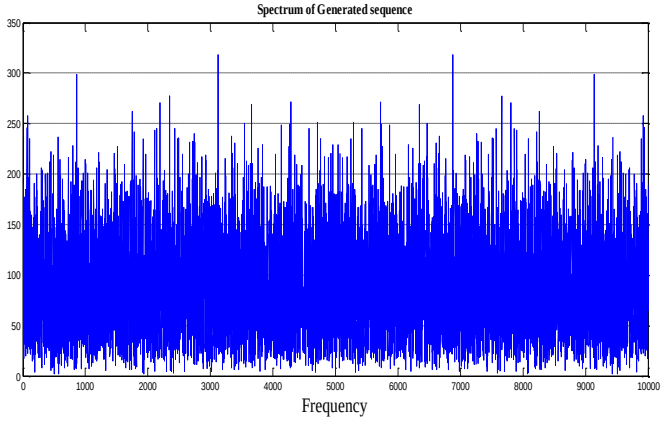



Fig 9: Spectrum of generated bit stream of EA5/1

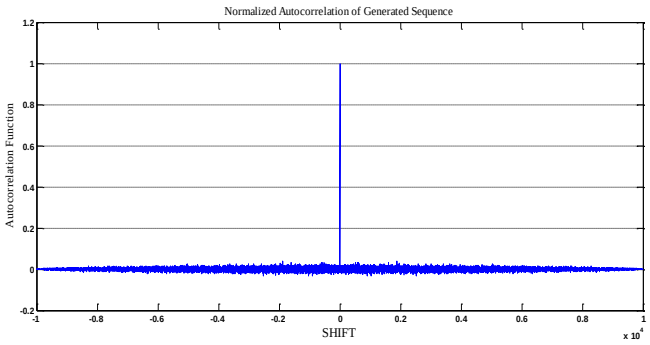



Fig 10: Autocorrelation of generated stream of EA5/1

significantly more e.g. commutative sum and serial teats etc. It is important to note that no test has P-value less than 0.01 that's mean it passes the tests and EA5/1 is a satisfactory random generator. For some tests the P-values are almost near to each other. The spectrum and autocorrelation of sequence generated by EA5/1 is shown in Figure 9 and Figure 10 respectively. Spectrum analysis of sequence implies that sequence is very close to the noise, as there is no dominating frequency component almost all frequency components are present in sequence. The observation of autocorrelation of sequence shown in figure 10 implies that sequence is not correlated to itself. It is close to noise. These two results are also showing the satisfactory randomness of the sequence generated by EA5/1.

## 5. CONCLUSION

In this paper, effect of parameters of enhanced A5/1 stream cipher is proposed. Modifications are done to improve the complexity of A5/1 algorithm to make it robust to attacks. It is observed that the changing of feedback taps and the shuffling of LFSRs is an effective to make the generator stronger. Now, cryptanalyst has to identify four feedback polynomials instead of one for each LFSR. This generator is also robust to Berleykamp Massey attack. Algorithm become more complex to break due to introduced shuffling of LFSRs. Though algorithm became complex but it is easy to realize. Based on the observations and results, it can be concluded that the proposed scheme is robust to the cryptographic attacks compare to the conventional A5/1 stream cipher. Hence the proposed scheme generates cryptographically better binary sequence than the A5/1 stream cipher of GSM with minor increase in the hardware.

## 6. REFERENCES

[1] G. Rose, A précis of new attacks on GSM encryption, Qualcomm, Australia, 10 september 2003.

[2] R. Mita, G. Palumbo and M. Poli, Pseudo-random sequence generators with improved inviolability performance, IEE Proceedings of Circuits, Devices and Systems, vol. 153, pp 375-382, 2006.

[3] J. Golic, Cryptanalysis of alleged A5 stream cipher, Advances in Cryptology, proceedings of EUROCRYPT'97, LNCS, vol. 1233, pp. 239–255, Springer-Verlag, 1997.

[4] A. Biryukov, A. Shamir, and D. Wagner, Real time cryptanalysis of A5/1 on a PC, Advances in Cryptology, proceedings of Fast Software Encryption'00, LNCS, pp. 1–18, Springer-Verlag, 2001.

[5] E. Biham, and O. Dunkelman, Cryptanalysis of the A5/1 GSM stream cipher, Progress in Cryptology, proceedings of INDOCRYPT'00, LNCS, pp. 43–51, Springer-Verlag, 2000.

[6] P. Ekdahl, and T. Johansson, Another attack on A5/1, IEEE Transactions on Information Theory, vol. 49, pp. 284-289, 2003.

[7] A. Maximov, T. Johansson, and S. Babbage, An improved correlation attack on A5/1, proceedings of SAC 2004, LNCS, vol. 3357, pp. 1–18, Springer-Verlag, 2005.

[8] E. Barkan, and E. Biham, Conditional estimators: an effective attack on A5/1, proceedings of SAC 2005, LNCS, vol. 3897, pp. 1–19, Springer-Verlag, 2006.

[9] S. E. AlAschkar and M. T. El-Hadidi, Known attacks for the A5/1 algorithm: a tutorial, International Conference on Information and Communications Technology (ICICT'03), pp. 229-251, 2003.

[10] B. Schneier, Applied Cryptography, protocols algorithm and souce code in c, Second edition, John Wiley & Sons Inc.

[11] Andrew Rukhin et all, NIST, A Statistical Test Suit for random and pseudorandom number generators for cryptographic applications. NIST Special Publication 800-22 , with revisions dated May 15, 2001.

[12] Recommendation GSM 02.09, European Telecommunications Standards Institute (ETSI), Security aspects.

[13] M. Galanis, P. Kitsos, G. Kostopoulos, N. Sklavos, O. Koufopavlou and C. E. Goutis, Comparison of the hardware architectures and FPGA implementations of stream ciphers, proceedings of the 11th IEEE International Conference on Electronics, Circuits and Systems (ICECS'04), pp. 571- 574, 2004.

[14] W. Ahmad, O, Farooq and Izharuddin, Stream Ciphering using a novel Pseudo-Random generator, the ICFAI University Journal of Electronics Engineering, vol 1, No.1, 2008.

[15] A. Braeken, Cryptographic properties of boolean functions and s-boxes, Katholieke Universiteit Leuven, Belgium, Ph.D Thesis, March 2006.