Improvement Majority Function in A5/1 stream cipher Algorithm

Dr. Hala Bahjat
Computer Science Department, University of Technology/Baghdad
Mohanad Ali
Computer Science Department, University of Technology/Baghdad
Email:mohanad_ali1986@yahoo.com

Received on:2/3/2015 & Accepted on:11/6/2015

ABSTRACT

Security is an important issue, especially in today's technologically advanced society. Global System for Mobile Communications (GSM) is a world-wide standard for digital wireless communication. GSM uses A5/1 stream cipher in order to provide privacy on air communication. This paper introduce new improvements to the A5/1 stream cipher in order overcome the weakness that appear in clocking mechanism that used in A5/1 stream cipher. New S-box generation is proposed to increase the efficient for A5/1 majority function and improve randomness features. The randomness results confirm that the output bit-stream generated by the proposed stream cipher has improved the randomness performance.

Keywords: GSM, A5/1 stream cipher, LFSR, S-Box, and randomness.

تطوير دالة الاسبقية في خوارزية A5/1 للتشفيير الانسيابي

الخلاصة

يعتبر امن المعلومات الرقمية مشكلة كبيرة ومهمة في عالم الاتصالات و خاصة عندما اصبحت التكنولوجيا تلعب مساحة واسعة في المجتمع ، النظام العالمي للاتصالات المتنقلة (GSM) هو النظام القياسي المستخدم عالميا في الاتصالات الرقمية اللاسلكية ، GSM يستخدم خوارزمية A5/1 للتشفير الانسيابي لتوفير السرية في فضاء الاتصالات ، في هذا البحث نحن نقدم تطوير جديد على خواررزمية A5/1 العامة لتجاوز نقاط الضعف التي ظهرت في عملية التزحيف فيها .

عملية توليد لـ S-BOX جديد مقترح لزيادة فاعلية دالة الاغلبية في خوارزمية A5/1 وتحسين الخصائص العشوائية اليها ، نتائج الفحوصات العشوائية تؤكد ان السلاسل المخرجة المولدة من قبل التحسين المقترح طورت الاداء العشوائي للخوارزمية.

INTRODUCTION

GSM is digital mobile technologies that are used for send and receive (voice, data) as services. GSM allows people to move from network to another, and provide user mobility. GSM became the most popular system for mobile phone worldwide. 4.4 billion Person was using GSM in more than 219 countries [1].

The security of GSM is very important; the most GSM popular encryption algorithm is A5/1. Recent research studies show that A5/1 cipher is crypt analyzed by a number of attacks. It has feeble clocking mechanism and output bit sequence of A5/1 has low rate of linear complexity [2].

We can classify attacks on A5/1 stream cipher into two main attacks

- Known plaintext attacks

https://doi.org/10.30684/etj.34.1B.2
2412-0758/University of Technology-Iraq, Baghdad, Iraq

- Time-memory trade-off-attacks.

The time-memory trade-off attacks, is considered exponentially as an expensive, can be avoided by increasing the length of registers.

Stream cipher known-plaintext attack, it is applied when the intruder knows some bits of the key and the rest key bits are guessed from this known bits. A known plaintext attack happened when the intruder has access to both the ciphertext and plaintext. [6]

One of the most important parameter that stream cipher algorithms like A5/1 suffering from is using XOR at the last part of algorithm to implement the final key stream. This caused to those algorithm big security problems [7].

In [3], proposed a modified A5/1 algorithm they add two more LFSRs to the design of original (A5/1 which has 3 LFSRs) with new polynomials:

$$f(x) = 1 + x^{14} + x^{17} + x^{18} + x^{19} \quad (1)$$
$$f(x) = 1 + x^{21} + x^{22} \quad (2)$$
$$f(x) = 1 + x^{8} + x^{21} + x^{22} + x^{23} \quad (3)$$
$$f(x) = 1 + x^{14} + x^{17} + x^{18} + x^{19} + x^{24} \quad (4)$$
$$f(x) = 1 + x^{21} + x^{22} + x^{25} \quad (5)$$

They notice that the proposed LFSRs much higher of the original A5/1 algorithm , this is one of the characteristics which will give us sequences which have best statistical properties comparable with original A5/1 the new proposed algorithm pass the frequency test , serial test , run test . In [4] , they add improvement to the original A5/1 algorithm which is considered unsafe and the try to secure it by using mechanism that use s-box with the dimension (4 * 16) of bits (0 and 1 only), three s-box is used and new method is suggested to decide the order of s-box to use between s-boxes, the researchers find that the proposed method have better randomness than A5/1.This method use three s-boxes from (Data Encryption Standard(DES) algorithm, these s-boxes is used only if the output of 3 LFSRs is 0,and use the original A5/1 otherwise , one of the LFSR is decide which of the two left is used to obtain output. And [5] they proposed new method to improve A5/1 by changing the clocking mechanism and add nonlinear function.

The clocking mechanism is mixing the register bit with other bits

M=C1.T1 ⊕C2.T2 ⊕C3.T3

"." Is refer to as the AND operation and "⊕" represent the XOR operation, T1, T2, T3 the tap bits of the equation where

T1 = R1 (13) ⊕R1 (16) ⊕R1 (17) ⊕R1 (18)

T2=R2 (19) ⊕ R2 (20)

T3=R3 (7) ⊕R3 (19) ⊕R3 (20) ⊕R3 (21)

And C1, C2, C3 represent the clocking taps.

The research proposed new nonlinear function to increase complexity of A5/1 as the following equation:

Z (t) = f(X1, X2, X3) = (X1.X2) ⊕(X1 ⊕X3). (X2.X3)

"." Represent the AND, "⊕" represent the XOR.

In this paper produce new improvement based on generate new S-box based on high and speed mathematical rules. New S-box used with 5 LFSR in stand of 3 LFSR to increase the quality for A5/1 majority function with consuming time.

A5/1 Stream Cipher Algorithm

A5/1 algorithm is the most widely used in GSM systems as encryption algorithm to encrypt data stream. The data transmitted as a sequence frames, each frame is 228bit (each 114 is for

communication in each direction), it is used 228 bits stream generated for each frame and it is XORD with the frame bits (frame bits represent the plain text).The A5/1 is started by using 64bits secret key (session key) with 22bits public key (frame number). It based on three LFSR shown in Figure (1) as follows [2]:

$$f(x) = 1 + x^{14} + x^{17} + x^{18} + x^{19} \quad (6)$$
$$f(x) = 1 + x^{21} + x^{22} \quad (7)$$
$$f(x) = 1 + x^{8} + x^{21} + x^{22} + x^{23} \quad (8)$$

Each LFSR is clocked based on majority rule (using three clocking bits X1,X2,X3 from R1,R2,R3 LFSR) the majority is counted as follow :
M = maj (X1, X2, X3)
Note: if two or more of X have 1 (tapped bit) the M is 1 otherwise M is 0
The clocked bit of each register is equal to M only these registers are clocked (regular clocking i.e: shifting). In the following illustrate A5/1 algorithm steps [2].

A5/1 algorithm [2]:
Input: 64 bit session key (secret key), 22 bit frame number (public key), 228 bit frame bits (plaintext).
Output: cipher text size 228 bits.
Process
Step 1: initialize 3 registers are set to zero.
Step 2:Load 64 bits of session key (secret key) + 22 bits of frame number (public key),session key and frame number is XORed bit-by-bit with the LSB (least significant bits), and the registers are clocked regularly.
Step 3 :(100) times the registers are cycled and discarding any output (all registers are clocked irregularly the majority function identify the shifted registers).
Step 4 :(228) times the registers are cycled (clocked irregularly the majority function identify the shifted registers) to generate the key stream.
Step 5: all steps repeated for the next frame.
Step6: end

In the following figure (1) shown the A5/1 algorithm based on 3 LFSR.

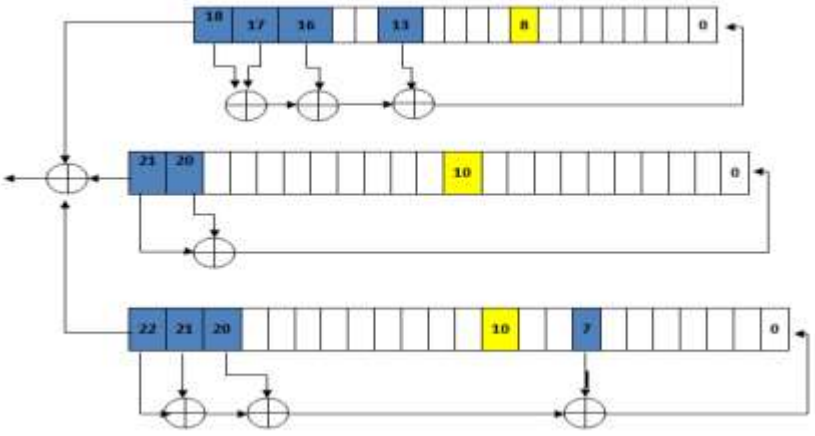

Figure (1): original A5/1 algorithm [3]

The main description for A5/1 algorithm improvement:

The new majority function based on the s-box concept is present, where new s-box is built based on simple mathematical operations, taped bits are entered to the new majority rules and number of active registers are identified in the following subsections illustrated the proposed improvements

New s-box generation

5 LFSR has suggested, the s-box input (5 bits one bit from each taped register), the s-box value will determine the number of active register to be shift from the 5 LFSR, DES S-box concepts used with the proposed S-box where the first and last bits will represent the row number; the three middle bits will represent the column (row and column value converted to decimal).

We have two bits as a row data, the maximum decimal number will be 4 and three bits as a column data the maximum decimal number will be 8, new s-box matrix will be (4*8) shown in table (1).

Simple mathematics rules are used to generate new s-box as following:

((A + B) MOD 5 =C+1)

Where A is the row value , B is the column value , mod 5 is used to have values less than 5 each value is added to one (so we have at least one register to be shifted).

Table (1): New S-box generation

+	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0	1	2	3	4	0	1	2
1	1	2	3	4	0	1	2	3
2	2	3	4	0	1	2	3	4
3	3	4	0	1	2	3	4	0

To illustrate the proposed rules that used with new S-box and how the S-Box work, let assume 5 registers bits values equal to R1 (0), R2 (0), R3 (1), R4 (1), R5 (1) row number = 01 which is 1 in decimal, column number=011 which is 3 in decimal the output of s-box = 4 (this mean the first five (4+1) registers will be shifted, if the output of LFSR is 2 the first three registers are shifted and so on), the s-box determine the number of active registers from the five major LFSR.

Proposed algorithm

The proposed modified A5/1 aim to provide more level of randomness, that solve the main weakness problem that appear in majority function by modify the majority function in traditional algorithm with new s-box generation that shown in table(1) . In the following figure (2), shown the A5/1algorithm based on 5 LFSR (see section [3]), and figure (3) shown the proposed A5/1 improvement.

Figure (2): The A5/1 algorithm with 5-LFSR.

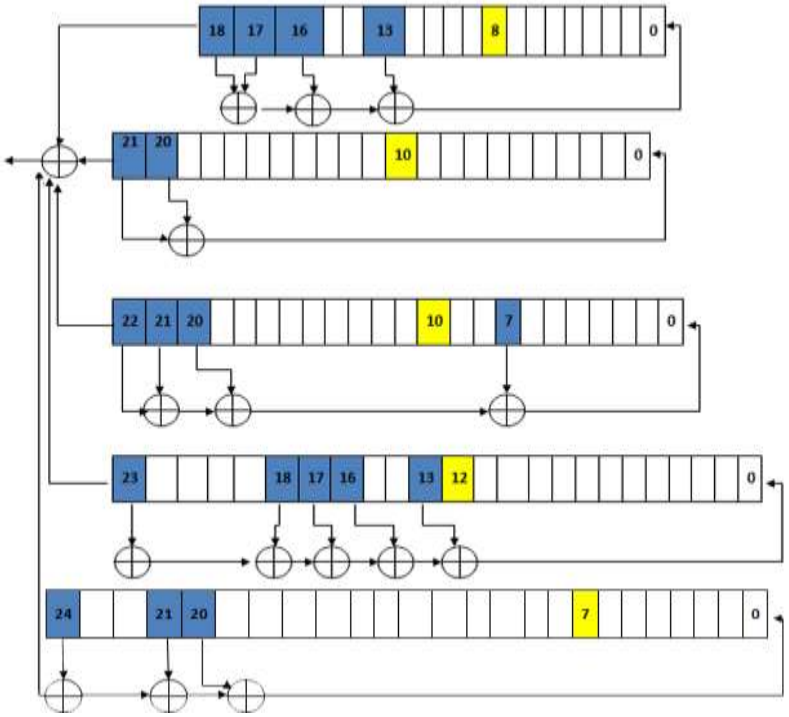

Figure (3): A5/1 with proposed S-box majority function.

A5/1 improvement algorithm:

Input: 64 bit session key (secret key), 22 bit frame number (public key), 228 bit frame bits (plaintext).

Output: 228 bit cipher text.

Process:

Step1: All 5 registers are set to zero.

Step2: Load 64 bits of session key (secret key) + 22 bits of frame number (public key),session key and frame number is XORed bit-by-bit with the LSB (least significant bits), and the registers are clocked regularly.

Step3: (100) times the registers are cycled and discarding any output (all registers are clocked irregularly the new majority function identify the shifted registers).

Step4: (228) times the registers are cycled (clocked irregularly the new majority function identify the shifted registers) to generate the key stream.

Step5: all steps repeated for the next frame.

End

In the following figure (4), shown A5/1 with taped registers values

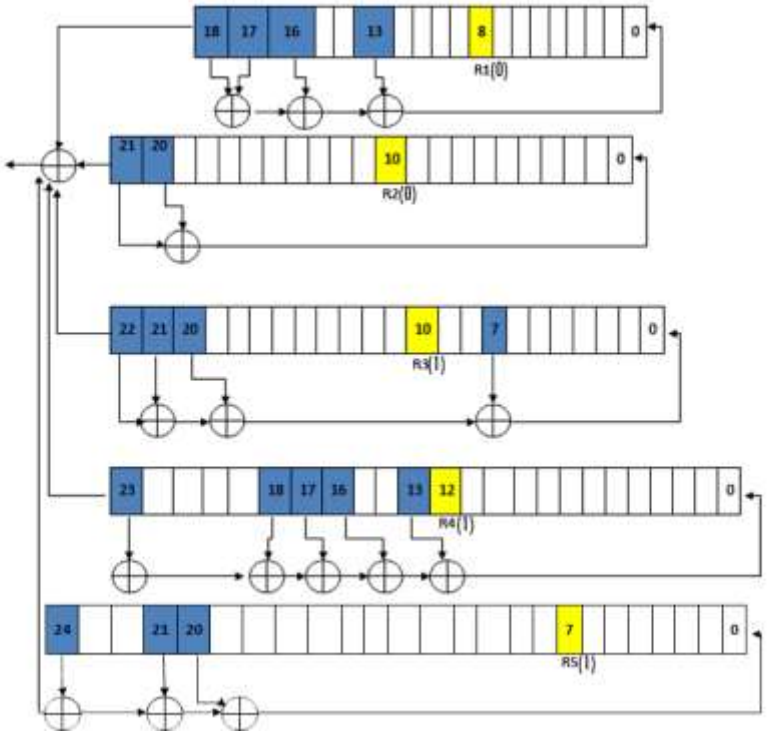

Figure (4): A5/1 with taped registers values

Experimental results:

In this section perform the comparison results between the algorithms (see section 3) and the A5/1 algorithm with improvement, in the following illustrated the implementation

- **Implement using Original A5/1 algorithm:**

Let session key entered to system be

{0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0}

And the frame number is

{1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1}

The key stream will calculated to be

{0,1,1,1,1,1,1,0,1,0,0,1,0,0,1,0,0,0,0,1,0,0,1,0,0,0,1,1,0,0,0,1,1,1,1,0,0,1,1,1,1
,1,0,0,11,1,0,0,1,0,1,1,0,1,1,1,0,1,1,0,0,0,0,0,1,0,0,1,1,1,1,0,0,1,0,1,1,1,0,0,1,
0,1,1,0,0,1,1,1,1,1,0,0,0,1,1,0,1,1,0,1,1,0,0,0,1,0,0,1,0,1,0,0,0,0,1,0,0,1,0,1,0,
0,0,0,0,1,1,1,1,1,1,1,1,1,0,0,0,0,0,1,1,1,1,1,0,1,0,0,0,1,1,0,1,0,0,1,1,1,0,0,1,1,
1,0,0,1,1,1,0,1,1,0,1,0,1,1,1,1,1,1,1,1,1,1,1,0,0,0,1,0,1,0,0,1,0,1,0,1,1,0,1,1,0,0,
1,1,0,0,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,0,0,0,0,1,0}

If we have

{0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1,
0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1,
1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0,
0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0,
1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1,
1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0,
1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1}

as a plaintext
Then the final cipher text (plaintext XOR key stream) will be

{0,0,1,1,0,0,1,0,0,0,1,1,0,1,0,0,0,1,0,1,0,1,1,1,0,0,0,0,1,1,0,0,0,1,1,1,0,0,1,1,0,
0,1,1,1,0,1,1,0,0,1,1,1,1,1,1,0,1,0,1,1,0,0,0,0,0,1,1,1,0,0,1,1,0,0,0,0,1,0,1,1,0,1,1,
1,1,0,0,0,1,1,0,0,1,0,0,1,0,1,0,1,1,0,1,0,0,1,0,0,0,0,1,1,0,0,0,0,1,0,1,0,1,0,0,0,0,
0,0,1,1,0,1,0,1,0,0,1,1,0,1,1,0,1,1,0,0,1,0,1,0,0,0,0,1,0,0,0,1,1,0,1,0,0,0,0,0,0,0,
1,1,1,0,1,0,0,1,1,0,1,1,1,1,1,0,1,1,0,0,0,0,1,0,1,1,1,0,1,1,0,0,1,0,1,0,0,0,1,0,1,0,1,
1,1,1,1,1,0,0,1,0,1,1,1,1,0,0,0,1,1,1,1,1,1}

- **Implement using the Proposed A5/1 algorithm improvement :**

Let session key entered to system be

{0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0,
0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0}

And the frame number is

{1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1}

The key stream will calculated to be

{0,1,1,1,1,0,1,1,1,1,0,1,1,0,0,0,1,0,0,1,0,1,0,0,1,0,0,1,1,0,1,1,0,0,1,0,0,1,1,1,0,1,
1,0,1,1,0,0,1,1,0,1,0,11,1,0,0,1,0,0,1,0,1,0,1,0,1,1,0,1,1,1,0,0,0,1,1,1,1,0,1,1,1,1,
0,1,1,1,1,0,0,0,0,1,1,1,0,0,1,1,0,0,1,0,1,1,0,0,1,1,0,0,1,0,1,1,0,1,1,1,0,0,0,1,0,0,1,
1,0,0,0,1,0,0,0,0,0,1,1,0,0,1,0,0,0,0,0,0,1,1,1,0,1,0,1,0,1,0,0,0,0,1,0,1,1,0,1,1,1,0,
0,0,1,1,0,1,0,1,0,1,1,0,0,0,1,1,1,0,1,0,0,1,0,0,1,0,1,0,1,0,0,0,0,1,1,0,1,0,1,1,0,1,0,
1,0,1,0,1,1,1,1,1,0,0,1,1,0}

If we have

{0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1,
1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1,
1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0,
1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1,
1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1,
0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1}

Then the final cipher text (plaintext XOR keystream) will be

{0,0,1,1,0,1,1,1,0,1,1,1,1,1,1,0,1,1,0,1,0,0,0,1,1,0,1,0,0,1,1,0,1,0,1,1,0,0,1,1,1,1,0,0,
1,1,0,0,1,1,0,1,1,0,0,1,0,0,0,1,0,0,1,0,1,0,1,1,0,0,1,1,1,1,0,1,0,1,0,1,0,0,1,0,1,1,0,1
,0,1,1,1,1,1,1,0,1,1,0,1,1,1,0,1,0,1,0,0,0,0,0,0,0,0,0,1,0,1,1,0,0,0,0,1,0,0,1,0,0,1,1,1,
0,1,1,1,1,0,1,0,0,0,0,0,1,1,1,0,0,0,0,1,1,0,0,1,1,1,1,1,1,1,0,0,0,0,1,0,1,0,1,1,1,0,0,1,1
,1,1,0,0,1,0,1,0,0,0,1,0,1,1,1,0,1,0,0,0,0,0,1,1,0,0,1,1,1,0,0,0,0,1,1,1,1,0,1,0,0,0,0,
0,1,1,0,0,1,1,1,0,1,1}

Security analysis and performance:

NIST Statistical test package approximate the randomness performance of output bit streams of two ciphers. The randomness results confirm that the output bit-stream generated by the proposed stream cipher has improved the randomness performance Randomness Key stream and cipher text are used in randomness tests and results of these tests used to prove that the keys have good randomness properties and can be used as good key in cryptography field.

All test are applied to the original and modified A5/1 algorithm (for keystream and cipher text) as shown in (figure (5), table (2) and table (3))

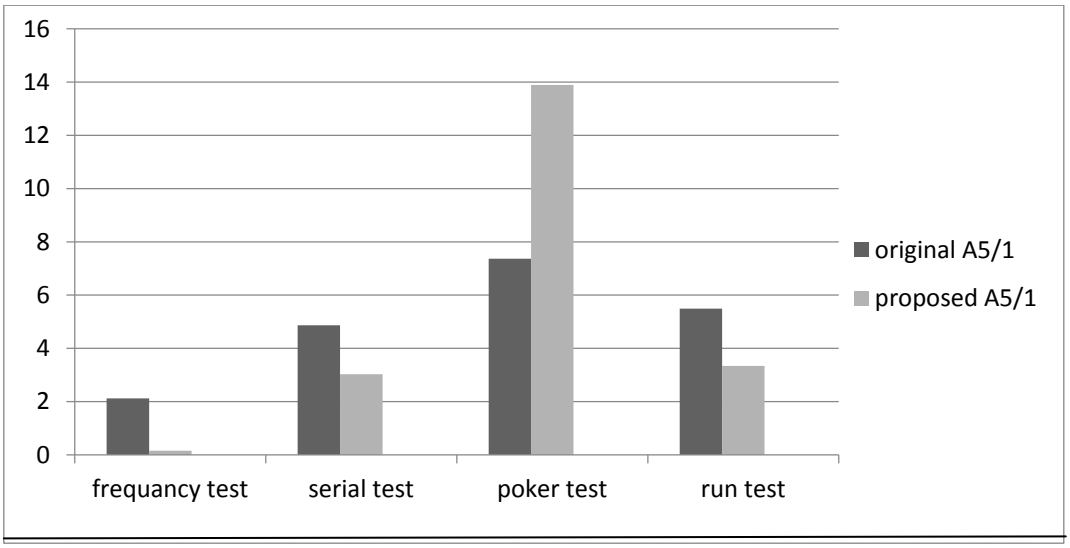

Figure (5): randomness tests results

Table (2) Tests Results Comparison For Keystream

Original A5/1		Proposed A5/1		
Frequency test	2.12280701754386	Frequency test	0.157894736842105	Must be ≤ 3.8415
Serial test	4.86838240976891	Serial test	3.0271272895896	Must be ≤ 5.9915
Poker test	7.36842105263158	Poker test	13.8947368421053	Must be ≤ 14.0671
Run test	5.4919965924748	Run test	3.33797835913117	Must be ≤ 9.4877
Auto correlation test	pass	Auto correlation test	pass	Must be ≤ 3.84

Table (3) Tests Results Comparison For Ciphertext

Original A5/1		Proposed A5/1		
Frequency test	0.43859649122807	Frequency test	0.157894736842105	Must be ≤ 3.8415
Serial test	0.490918927274123	Serial test	0.172501738928815	Must be ≤ 5.9915
Poker test	8.0	Poker test	4.63157894736842	Must be ≤ 14.0671
Run test	1.78604060702354	Run test	3.6570441078013	Must be ≤ 9.4877
Auto correlation test	pass	Auto correlation test	pass	Must be ≤ 3.84

(NOTE: Pass means that all bits in the 228 bit generated key pass the autocorrelation test, reference to NIST standards and qi square)

According to table (2) and table (3) .The proposed algorithm succeeds to pass all five popular randomness tests (frequency, serial, poker, run and Auto correlation test) as shown in figure (5) .

Table (2) shown the comparison results between the original A5/1algorithm and A5/1 algorithm with improvement in key randomness , Table (3) shown the comparison results between the original A5/1algorithm and A5/1 algorithm with improvement in cipher randomness.

Time complexity is checked to the suggested improvement the original algorithm work with (2 seconds) for key generation and encryption, the suggested algorithm need (2 seconds) for key generation and encryption

CONCLUSION

The proposed algorithm succeed to introduce improvement to the original A5/1 algorithm and the majority function , with respect to the secrecy and complexity and time factors to be efficient and applicable , we suggest an enhanced A5/1 algorithm used in GSM. The enhancements applied on the clocking mechanism (majority function) of the A5/1. It is noticed that it is more regularity in the clocking operation for the proposed approach. The s-box is used to provide decisions to which registers to be clocking. In the proposed schema we observe much better registers clocking. Also, the cipher text of the proposed algorithm has more complexity comparable with complexity the cipher text of the original A5/1 .we can conclude that the proposed method provides cryptographically better sequences than the original A5/1 cipher of GSM.

REFRENCES

[1].Magnus G., Kristian H. and Espen H., ‘’Decoding GSM’ M.Sc thesis in Communication Technology, Norwegian University of Science and Technology, 2010

[2]. Prof. Darshana Upadhyay, Dr.Priyanka Sharma, Prof.Sharada Valiveti, “Randomness analysis of A5/1 Stream Cipher for secure mobile communication”, IJCSC (International Journal of Computer Science and Network Security) VOLUME 5, 2014.

[3].Nur Hafiza Zakaria, Kamaruzzaman Seman, Ismail Abdullah , “Modified A5/1 based stream cipher for secured GSM communication”, IJCSC (International Journal of Computer Science and Network Security,) VOLUME 11, 2011.

[4]. Mi-Og P., Yeon-Hee C. and Moon-Seog J. “Modified A5/1Stream Cipher using S-boxes” IEEE, 2004.

[5]. Mahdi Madani, Salim Chitroub, “Enhancement of A5/1 Stream Cipher Overcoming its Weaknesses’’, ICWMC: The Tenth International Conference on Wireless and Mobile Communications, 2014.

[6]. IISER Pune, Dr. Homi Bhabha Road, Pashan Pune 411008,” An Improved Guess-and-Determine Attack on the A5/1Stream Cipher”, Computer and Information Science; Vol. 7, No. 1; 2014, Canadian Center of Science and Education, 2014.

[7]. Majid Bakhtiari 1 Mohd Aizaini Maarof,” An Efficient Stream Cipher Algorithm for Data Encryption”, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 1, May 2011,2011.