Improvement A5/1 Encryption Algorithm Based On Sponge Techniques

Asst.Prof. Dr. Hala B. Abdul Wahab
Computer Science Department
University of Technology
Baghdad -iraq
hala_bahjat@yahoo.com

Mohanad A. Mohammed
Computer Science Department
University of Technology
Baghdad -iraq
mohanad_ali1986@yahoo.com

***Abstract*— A5/1 stream cipher is used in Global System for Mobile Communications (GSM) in order to provide privacy on air communication. In this paper introduce new improvements to the A5/1 stream cipher based on using new technology concepts called sponge function. Sponge functions that represent in this paper constructed based on combine between the advantage of stream cipher and hash concepts. New S-box generation is proposed to provide the dynamic features to the sponge technology in order solve the weakness that appear in majority function that used in A5/1 stream cipher by provide dynamic behavior in number of registers and transformation . According the experimental results and the compassion between the A5/1 and the proposed improvement shown the proposed algorithm will increase the randomness features for the A5/l algorithm. The output bit-stream generated by the proposed stream cipher has improved the randomness performance and provide more security to the GSM security algorithm.**

Keywords*— sponge, s –box, stream cipher, A5/1, randomness, GSM.

I. INTRODUCTION

The communications of mobiles become more acceptances in technical society. By using mobile, anyone can be in any place. In 2007, the mobile users became 2.83 billion person and 2.28 billion (i.e. 80.5%) person use GSM (the Global Service for Mobile communications). The security of Global Service for Mobile communications originally mean ability to provide services related to security such as confidentiality, authentication, and anonymity, of persons data and information related to signal. The security goals of GSM are as follows:
- Mobile user's authentication with network.
- User data and information signal confidentiality.
-Users Anonymity
-Using Subscriber Identity Module (SIM) as a security module. [1]
A5/1 is a stream cipher encryption algorithm used in Global Service for Mobile communications to cipher the data in the air transmission. Global Service for Mobile communications conversations are treated and sent as frames. Every 4.6 Milliseconds one frame is sent 228 bit length; 114 bits are used for the communication in one direction. A5/1 algorithm main purpose is to use it as a key generator and resulted 228 bits of key which is XORed with the bits of frame. [2]. The sponge construction is a concept accepts long input and produce output sizes, which allows building different primitives like stream cipher and hash function .this mean input is short (usually the key) and the output produced is as long as the plaintext (message) to encrypt.
The sponge function takes input with variable-length and produce arbitrary output length using a fixed length transformation. [3].

978-1-4673-6636-6/15/$31.00 ©2015 IEEE.

II. *A5/1 Stream cipher description*

A5/1 first initialized by using a 64-bit session key (secret key) with a 22-bit frame number (public key). The A5/1 algorithm consists of three (LFSRs) linear feedback shift registers, R1, R2 and R3 with 19, 22 and 23 bits lengths respectively. Three polynomials used for LFSR R1, R2 and R3 are:

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

Registers are clocked using irregular clocking which depends on the majority rule. The majority rule take the three clocking bits C1, C2 and C3 of registers R1, R2 and R3 and count the value of majority value m ass follow m = maj(C1, C2, C3), if two or more are 1 then the majority value m is 1. And reverse. [2]

A. A5/1 algorithm [4]

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

DOI : 04.IJIS.2015.1.2

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.

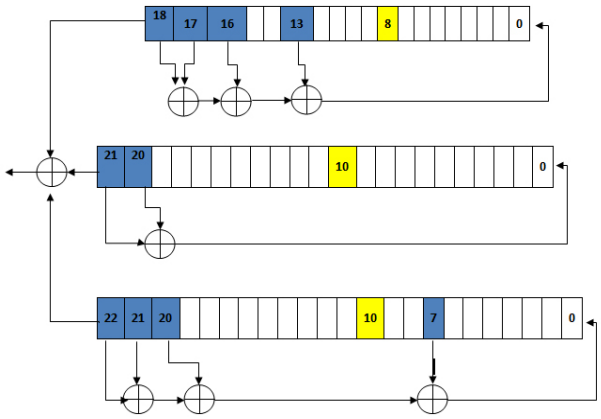

Fig. 1.original A5/1 algorithm

B. *Sponge function:*

For cryptographic systems, we need to generate unpredictable bits, even if part of this sequence is revealed. Sponge function is similar to a stream cipher. If we have unknown key, the key must be infeasible to any infer on the key stream, even if part of it is known, The sponge function shown in figure (2, 3) have (variable-length input) and (variable-length output), it is unify way the stream ciphers with functionality of hash functions. [5].

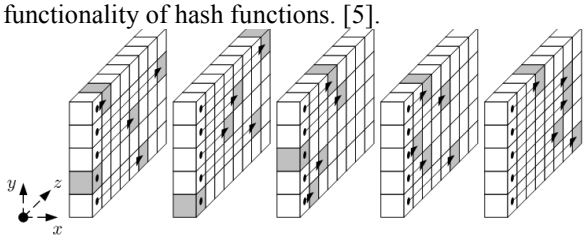

Fig. 2.sponge function (6)

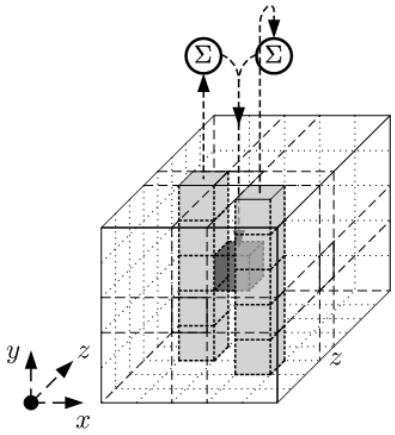

Fig. 3.Keccak sponge function [7]

C.*The main A5/1 algorithm improvement:*

The proposed a sponge function that modifies the A5/1 to (dynamic majority based on dynamic LFSR length), the permutation of the system is based on five s-boxes tables build on simple mathematical operations and the length of the LFSR is based on two s-boxes.

III. SPONGE FUNCTIONS:

In this section represent the proposed session key generation based on sponge functions behavior. The session key is divided into two parts (sk1, sk2) each part is (32 bit), and each part is converted to decimal (every two bits will represent one decimal value), sk1 and sk2 will used to choose the length of LFSR by using two tables, original session key bit will determine the table used and the output of that table represent the length of LFSR of that bit only. Each LFSR length will own transformation table for shifting mechanism .

A. *Sponge function dynamic length:*

Five LFSR used, the session key is divided to two parts sk1 and sk2, each two bits of sk1 and sk2 is converted to decimal , the length is detected using two tables , the first table is based on the following equation :
((A+1) + (B+1) MOD 4
The second table is based on the following equation:
((A+1) XOR (B+1) MOD 4
Each bit on session key will detect the table used (for each bit on the session key different length)

TABLE I. LENGTH1 TABLE

((A+1) + (B+1) MOD 4	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

DOI : 04.IJIS.2015.1.2

TABLE II. LENGTH2 TABLE

((A+1) XOR (B+1) MOD 4	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

B . Sponge function transformation:

The length of LFSR will be dynamic and range from 1 to 5 (for each bit in session key), length decided for that bit will enter permutation operation in dynamic tables based on the following equations:

T 1: (A XOR B MOD 2) // where two LFSRs used

T 2: (A + B MOD 3) // where three LFSRs used

Tapped bit in LFSR1 represent A and tapped bit in LFSR2 and LFSR3 represent B.

T 3: (A XOR B MOD 4) // where four LFSRs used

Tapped bit in LFSR2 and LFSR3 represent A and tapped bit in LFSR1 and LFSR4 represent B.

T 4: (A + B MOD 5) // where five LFSRs used

Tapped bit in LFSR1 and LFSR5 represent A and tapped bit in LFSR2, LFSR3 and lfsr4 represent B.

TABLE III. T1 TABLE

(A XOR B MOD 2)	0	1
0	0	1
1	1	0

TABLE IV. T2 TABLE

(A + B MOD 3)	0	1	2	3
0	0	1	2	0
1	1	2	0	1

TABLE V. T3 TABLE

(A XOR B MOD 4)	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

TABLE VI. T4 TABLE

(A + B MOD 5)	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

IV. PROPOSED A5/1 ENCRYPTION 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 the sponge function concept. In the following figure (3), shown the A5/1algorithm based on 5 LFSR, and figure (5, 6) shown the proposed A5/1 improvement.

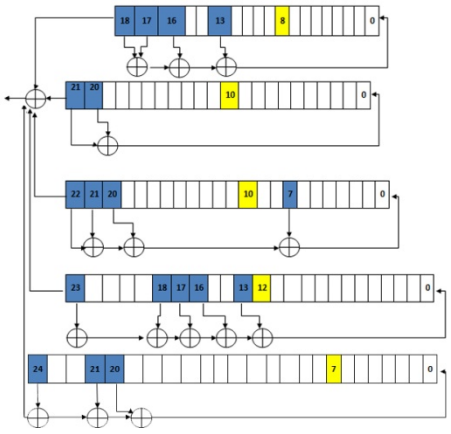

Fig. 4. The A5/1 algorithm with 5-LFSR.

The LFSR used in the proposed system is as follow:

$$f(x) = 1 + x^{14} + x^{17} + x^{18} + x^{19} \quad (4)$$

$$f(x) = 1 + x^{21} + x^{22} \quad (5)$$

$$f(x) = 1 + x^{8} + x^{21} + x^{22} + x^{23} \quad (6)$$

$$f(x) = 1 + x^{14} + x^{17} + x^{18} + x^{19} + x^{24} \quad (7)$$

$$f(x) = 1 + x^{21} + x^{22} + x^{25} \quad (8)$$

DOI : 04.IJIS.2015.1.2

NNGT Int. J. on Information Security, Vol. 4, October 2015

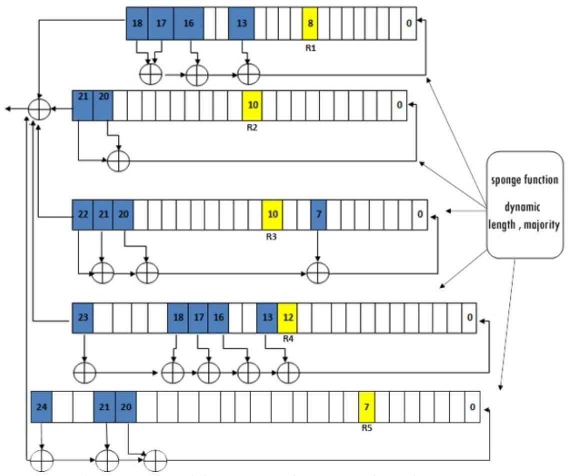

Fig. 5.A5/1 with proposed sponge function.

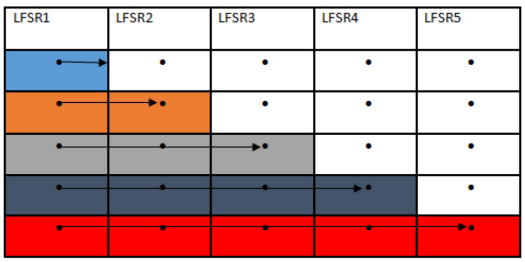

Fig .6.proposed sponge function dynamic length.

A. A5/1 algorithm based on the proposed improvements:

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: 64 bits session key is split to sk1 and sk2, both sk1 and sk2 is converted to decimal.
Step4: (100) times the session key bit will choose the table that will decide the length of LFSR for that bit only and the length(LFSRs) will be permuted and shifted using special tables (T1,T2,T3,T4,T5) and discarding any output
Step5: (228) times the session key bit will choose the table that will decide the length of LFSR for that bit only and the length(LFSRs) will be permuted and shifted using special tables (T1,T2,T3,T4,T5) to generate the keystream.
Step6: all steps repeated for the next frame.

B. Experimental results:

In this section we perform the comparison results between the algorithms A5/1 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
{0,1,1,1,1,1,1,0,1,0,0,1,0,0,1,0,0,0,0,1,0,0,1,0,0,0,1,1,0,0,0,1,1,1,1,0,0,1,1,1,1,1,0,0,11,1,0,0,1,0,1,1,0,1,1,1,0,1,1,0,0,0,0,0,1,0,0,1,1,1,1,0,0,1,0,1,1,1,0,0,1,0,1,1,0,0,1,1,1,1,1,0,0,0,1,1,0,1,1,0,1,1,0,0,0,1,0,0,1,0,1,0,0,0,0,1,0,0,1,0,1,0,0,0,0,0,1,1,1,1,1,1,1,1,1,0,0,0,0,0,1,1,1,1,1,0,1,0,0,0,1,1,0,1,0,0,1,1,1,0,0,1,1,1,0,0,1,1,1,0,1,1,0,1,0,1,1,1,1,1,1,1,1,1,0,0,0,1,0,1,0,0,1,0,1,0,1,1,0,1,1,0,0,1,1,0,0,1,1,1,1,0,1,1,1,1,1,1,1,0,0,0,0,1,0}
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, 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, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 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
{0,0,1,1,0,0,1,0,0,0,1,1,0,1,0,0,0,1,0,1,0,1,1,1,0,0,0,0,1,1,0,0,0,1,1,1,0,0,1,1,0,0,1,1,1,0,1,1,0,0,1,1,1,1,1,0,1,0,1,1,0,0,0,0,0,1,1,1,0,0,1,1,0,0,0,0,1,0,1,1,0,1,1,1,1,0,0,0,1,1,0,0,1,0,0,1,0,1,0,1,1,0,1,0,0,1,0,0,0,0,1,1,0,0,0,0,1,0,1,0,1,0,0,0,0,0,0,1,1,0,1,0,1,0,0,1,1,0,1,1,0,1,1,0,0,1,0,1,0,0,0,0,1,0,0,0,1,1,0,1,0,0,0,0,0,0,0,1,1,1,0,1,0,0,1,1,0,1,1,1,1,0,1,1,0,0,0,0,1,0,1,1,1,0,1,1,0,0,1,0,1,0,0,0,1,0,1,0,1,1,1,1,1,0,0,1,0,1,1,1,0,0,0,1,1,1,1,1}
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
{1,1,0,1,0,0,0,1,1,0,1,0,1,0,0,0,0,1,0,1,1,1,1,1,1,0,1,1,0,0,1,0,1,0,0,0,1,1,1,0,0,1,1,1,0,0,1,0,0,1,1,1,1,0,1,1,1,1,1,1,0,1,0,0,1,1,1,0,1,0,0,1,0,1,0,1,1,1,0,1,0,0,1,1,0,0,1,0,1,0,1,0,1,0,1,1,0,1,1,0,0,1,1,1,0,1,1,1,1,0,0,0,0,0,0,0,0,1,1,0,1,0,0,1,1,0,0,0,0,1,0,1,1,0,0,1,0,0,0,1,0,0,1,1,0,0,0,0,1,0,0,0,0,0,0,1,0,1,0,1,1,1,1,0,1,0,0,1,1,0,0,1,1,0,1,0,1,0,0,1,0,1,0,1,1,0,0,0,1,1,1,0,1,0,0,0,1,1,0,1,0,0,1,0,0,0,1,1,1,1,0,0,0,0,1,0,0,0,1,1,0,0,0,0,1,1,0}
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, 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, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 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,

DOI : 04.IJIS.2015.1.2

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 keystream) will be
{
1,0,0,1,1,1,0,1,0,0,0,0,1,1,1,0,0,0,0,1,1,0,1,0,1,0,0,0,1,1,1,1,
0,0,0,1,1,0,1,0,1,0,0,0,0,1,1,1,0,0,0,1,1,1,1,0,1,1,1,1,0,1,0,0,
1,1,0,1,0,1,0,1,0,1,1,1,1,0,0,0,0,0,0,1,0,0,1,1,1,0,0,1,0,0,1,1,
1,0,0,0,1,0,1,1,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,1,0,0,0,1,1,0,0,
0,1,1,1,1,0,1,0,0,0,1,1,1,0,0,0,0,0,0,0,0,1,0,1,0,0,1,0,0,0,0,0,
1,1,0,0,1,1,0,0,1,1,0,0,0,1,0,0,1,1,0,0,1,1,1,0,0,0,0,0,1,0,0,1,
1,1,0,1,1,1,1,1,0,1,1,1,1,1,0,1,0,1,1,0,0,0,1,1,1,0,0,1,0,1,0,1,
1,0,1,1}

C. 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 as shown in (figure (7), and table (7))

TABLE VII. TESTS RESULTS COMPARISON FOR KEY STREAM

Original A5/1		Proposed A5/1		
Frequency test	2.1228070175438 6	Frequency test	0.0	Must be ≤ 3.8415
Serial test	4.8683824097689 1	Serial test	0.825025117860719	Must be ≤ 5.9915
Poker test	7.3684210526315 8	Poker test	2.315789473684 21	Must be ≤ 14.0671
Run test	5.4919965924748	Run test	0.947315542986953	Must be ≤ 9.4877
Auto correlation test	pass	Auto correlation test	pass	Must be ≤ 3.84

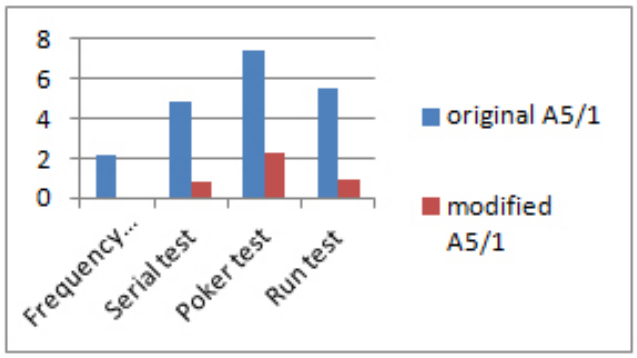

Fig .7.Randomness tests results

CONCLUSION

The proposed algorithm succeed to introduce new 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 , the improvements to the A5/1 stream cipher based on using (sponge concepts) by adding S-box generation succeed to provide the dynamic features. The sponge technology help to solve the majority function weakness that appear in standard A5/1 algorithm according to the dynamic sponge behavior in number of registers and transformation are provided. The proposed algorithm increase the randomness features for the A5/l algorithm and the output bit-stream generated by the proposed stream cipher has improved the randomness performance and provide more security to the GSM security algorithm.

REFERENCES

[1] Mohsen Toorani , Ali A. Beheshti , “Solutions to the GSM Security Weaknesses “,Proceedings of the 2nd International Conference on Next Generation Mobile Applications,Services, and Technologies “, University of Glamorgan, Cardiff, UK, Sep. 2008.

[2] Musheer Ahmad and Izharuddin , “Enhanced A5/1 Cipher with Improved Linear Complexity” , College of Engineering and Technology, AMU, Aligarh India , 2009.

[3] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche , “Duplexing the sponge: single-pass authenticated encryption andother applications” available at http://sponge.noekeon.org/SpongeDuplex.pdf .

[4] Prof. Darshana Upadhyay, Dr. Priyanka Sharma, Prof.Sharada Valiveti , “Randomness analysis of A5/1 Stream Cipher for secure mobile communication”, IJCSC volume 5 , 2014 .

[5] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche ,” Sponge-based pseudo-random number generators” , STMicroelectronics , NXP Semiconductors.

[6] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche “Keccak and the SHA-3 Standardization” , NIST, Gaithersburg, , 2013.

[7] Soham Sadhu , “Keccak discussion” , 2012 available at www.cs.rit.edu/~hpb/Lectures/20112/S_T/Src/32/keccak.pdf.

DOI : 04.IJIS.2015.1.2