J. Basic. Appl. Sci. Res., 3(1)564-573, 2013

ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

Analysis of LFSR Based Snow Family against Guess and Determine Attack

[1]Tarranum Naz, [2]Muhammad Tahir, [3]Aihab Khan

[1]Department of Computer Science, Fatima Jinnah Women University, Rawalpindi, Pakistan
[2] Department Computer Engineering, Sir Syed University of Engineering and Technology, Karachi
[3] Department of Computer Science, Fatima Jinnah Women University, Rawalpindi, Pakistan

ABSTRACT

Stream Cipher is a cryptographic primitive that is used to make sure privacy on a communication channel. SNOW family is a typical example of word oriented stream ciphers based on Linear Feedback Shift Register (LFSR). In this paper two versions of SNOW family have been analyzed against Guess and Determine (GD) Attack. Original SNOW 2.0 is an improved version of SNOW 1.0 claimed to be more secure and efficient in performance. Vulnerabilities in SNOW 2.0 give rise to another version of SNOW 2.0 called Modified SNOW 2.0. Both versions have claim that their model is secure against Guess and Determine attack. The purpose of this paper is to verify their claimed and to determine that which version has more resistance against Guess and Determine attack. Both algorithms are evaluated experimentally in two phases. Analysis and experimental results indicate that, for small number of key streams Modified SNOW 2.0 shows more resistance against GD attack. But when the number of key streams generated becomes larger Original SNOW 2.0 becomes more secure.

KEYWORDS: Guess and Determine Attack, Linear Feed Back Shift Register (LFSR), Modified SNOW 2.0.Original SNOW 2.0, Stream Cipher.

INTRODUCTION

SNOW is a word oriented stream cipher. The very first version of SNOW (i.e. SNOW 1.0) was submitted to NESSIE projects in March 2000. Some weaknesses were found in SNOW 1.0 [9] therefore a second version named SNOW 2.0 has been introduced. The new version is schematically a small modification of the original construction. Although SNOW 2.0 was appear to be more secure, but some flaws have also been found in it [3] which results in proposal of Modified Version of SNOW 2.0 [9]. The one reason of failure of prior two versions is Guess and Determine Attack. It has been detailed discussed in detail prior version of this paper [15].

Guess and Determine attacks have been laid in the class of General attacks. These attacks have often been used heuristically [14]. In these attacks the contents of some other cells can be determined. The first version of SNOW (SNOW 1.0) was captured by GD attack in a way that there is only one input from the LFSR to FSM, if an attacker make guess on that input it will enable him/her to invert the operation in FSM to determine the accurate keystream. And the second version SNOW (SNOW 2.0) was captured by correlation attack [12].

Over the last few years, a rising but limited number of papers have been published offering two or more designs of single stream cipher with claim that each version is more secure and reliable then the previous one. So it is essential to give information to the open world that which one is more secure amongst all. Currently two versions of SNOW are available. This paper investigates the resistance of these two versions of SNOW against GD attack and gives information to the open world that which version is more secure. Which should be recommended to open world for use.

Analysis of SNOW 1.0: SNOW 1.0 is the first version of SNOW family. It is a generator which generates keystream with the help of LFSR and FSM. It has LFSR of length 16. And input from LFSR is fed into FSM. FSM further consists of two 32-bit registers named R1 and R2. The generator is working in a way that, first of all the process of key initialization has been done. This process provides initial values to LFSR and FSM. The input of the FSM is bitwise added to last entry of the LFSR and generates first 32 bits of keystream. Then after every clocking next 32-bits of keystream are generated. The graphical model of SNOW 1.0 is shown in Fig 1.

Corresponding Author: Muhammad Tahir, Computer Engineering Department, Sir Syed University of Engineering and Technology, Karachi

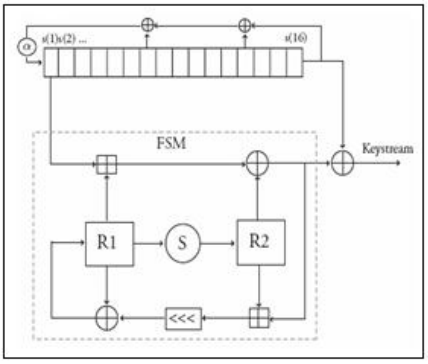


Fig 1: A Schematic Model of SNOW 1.0

Symbols involved in model are described in Table 1 which is given below:

Table 1: Symbols of SNOW

Symbols	Description
$\oplus$	Bitwise XOR
⊞	Addition Modulo 2^{32}
<<<	A cyclic shift of 7 steps
S	S-Box Operation

Mathematical Model of SNOW 1.0: The LFSR has a primitive feedback polynomial,

$P(x) = x^{16} \oplus x^{13} \oplus x^{7} \oplus \alpha^{-1}$

Furthermore let $s_{15},s_{14}……..s_0$ be the initial state of the LFSR, R1 and R2 are the registers of FSM. Working of SNOW 1.0 is divided into following steps:

Step I
Linear FeedBack Shift Register and FSM registers (R_1 and R_2) are initialized.

Step II
32- bits (i.e. first register of LFSR) from LFSR is inserted into FSM as an input.

Step III
Output of FSM is calculated as:
FSMout = ((s(1) ⊞ R_1) $\oplus$ R_2)
Step IV
32 bits of running key have been calculated as:
running key = FSMout $\oplus$ s(16)
Step V
The next state of the FSM is computed as:
tempR1 = ((FSMout ⊞ R_2) <<<) $\oplus$ R_1
R_2 = S (R_1)
R_1 = tempR1
S(x) denotes the S-Box operation; it splits the input x (32 bits) into 4 bytes (in a group of 8 to 8 bit S-boxes) from most significant to least significant byte. Then each byte passes over a nonlinear mapping from 8 bits to 8 bits [5].
Analysis of SNOW 2.0: SNOW 2.0 is an enhanced version of SNOW 1.0 which is also proposed by its former creators Patrick Ekdahl and Thomas Johansson in 2002 [6].

The word size of SNOW 2.0 remains same (32 bits) as of SNOW 1.0 and length of LFSR is again 16. But the feedback polynomial is different, now the two different elements α and α^{-1} are involved in feedback polynomial. In SNOW 2.0 FSM has two inputs words are taken from LFSR instead of one. And the running key is generated by XOR between FSM's output and last entry of LFSR as in SNOW 1.0. The cipher works in a way that, first of all LFSR and registers of FSM (R1 and R2) have been initialized by the process of key initialization. There is a small difference in the operation of the both versions. In first version SNOW 1.0 the first symbol was read out before the clocking of cipher but in second version it is read out after the cipher is clocked once [6]. For the generation of keystream, SNOW 2.0 take two inputs (i) a secret key of either 128 or 256 bits and (ii) a publicly known 128 bit initialization value IV. The IV value is basically considered as a four word input IV= IV_3, IV_2, IV_1, IV_0)

The graphical model of SNOW 2.0 is given below in Fig 2. Where as the symbols involved in model have been described in Table 1:

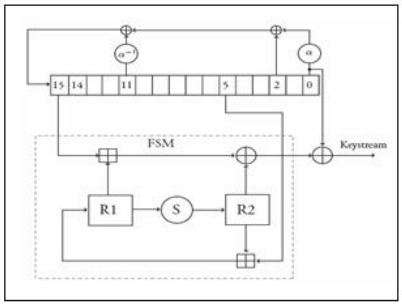


Fig 2: A schematic model of SNOW 2.0

Mathematical Model of SNOW 2.0: The LFSR has a primitive feedback polynomial,

$P(x) = \alpha^{-1} x^{11} \oplus x^2 \oplus \alpha x$

Working of SNOW 2.0 is divided into following steps:

Step I
Linear FeedBack Shift Register (LFSR) is initialized, and registers of Finite State Machine (FSM) R_1 and R_2 are set to be zero.

Step II
The cipher clocks 32-times without producing any keystream and incorporate output of FSM back into the LFSR.

Step III
The proper working of cipher has been started and output of FSM has been computed as:

$f_t = (S_{t+15} \boxplus R1_t) \oplus R2_t$, $t \geq 0$

Step IV
The running keystream is calculated as:

$z_t = f_t \oplus S_t$, $t \geq 1$

Step V
The cipher clocks and next state of the FSM is computed as:

$R1_{t+1} = S_{t+5} \boxplus R2_t$ and
$R2_{t+1} = S(R1_t)$ $t \geq 0$

Analysis of Modified Version of SNOW 2.0: A number of weaknesses have been found on SNOW 2.0 by P. Hawkes and G. G. Rose [7] and also by D. Coppersmith, S.Halevi, and C.Jutla [8]. Then another version of SNOW 2.0 is proposed in 2005 [9] called Modified version of SNOW 2.0.

As modified SNOW 2.0 is an enhanced version of SNOW 2.0. Therefore small amount of modifications have been done in basic design of SNOW 2.0. Both prior versions have basically the same design with minor changes. In modified version of SNOW 2.0 the linearity is converted into non linearity or in other words LFSR property is converted in NLFSR property. For security consideration following changes has been done in SNOW 2.0.

1. After XORing α (alpha) with S_{t+2} take a circular left shift.
2. After XORing α^{-1} (alpha inverse) with S_{t+11} take circular left shift again.
3. And take circular left shift of ($R_1 \boxplus S_{t+15}$) once again before XORing with R_2.

The graphical model of improved version is represented in figure 3. It is clear from the diagram that, there is a need of memory buffers at three different places to store the bit stream on temporary basis.

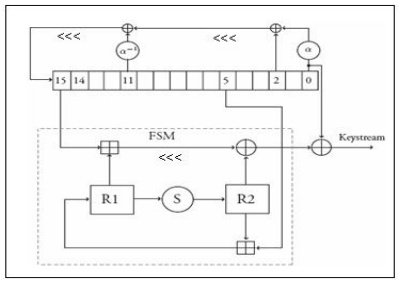


Fig 3: Model of Modified SNOW 2.0

Mathematical Model of Modified version of SNOW 2.0: The LFSR has a primitive feedback polynomial,

$P(x) = \alpha^{-1} x^{11} \oplus x^2 \oplus \alpha x$

Working of Modified Version of SNOW 2.0 is divided into following steps:

Step I
Linear FeedBack Shift Register (LFSR) is initialized and two registers (R_1 and R_2) of Finite state Machine (FSM) are set to be zero.

Step II
The cipher will clock 32-times without producing any output. The output of FSM in incorporated back into the LFSR.

Step III
The input to the FSM(St+5 and St+15) are given and output of the FSM is calculated as:

$tempf_t = S_{t+15} \boxplus R1_t$

take circular left shift of ($S_{t+15} \boxplus R1_t$) before Xoring with R_2

$tempf_t = (S_{t+15} \boxplus R1_t) <<< 7)$

$f_t = tempf_t \oplus R2_t$, $t \geq 0$

Step IV
The running keystream is calculated as:

$z_t = f_t \oplus S_t$, $t \geq 1$

Step V
The next state of FSM is computed as:

$R1_{t+1} = S_{t+5} \boxplus R2_t$ and

$R2_{t+1} = S(R1_t)$ $t \geq 0$

Step VI
The next state of LFSR is calculated as:
$S_{16} = ((\alpha^{-1} S_{t+11}) <<<,7) \oplus S_{t+2} \oplus ((\alpha S_t) <<<,7)$

Cryptanalysis of Snow Family
This section will introduce some criteria according to which reliability of both versions have been checked.

Guess and Determine Attack: In this paper the GD attack is applied in a way that Guess has been made on secret key and on basis of these guesses keystream is determined. For each guess the cipher will run for some time and match the output from the trial generator with the original sequence.

Working of GD Attack: The algorithm for GD attack is working in a way that:

Step I
Make guess on secret key and IV values.
Step II
Convert the guessed key into binary form.
Step III
Transform binary form of guessed key into 32 bits form.
Step IV
Secret key is ready for initialization process. Initialization has been done.
Step V
After initialization generator will start working and keystreams will be determined.

Working of Comparison Algorithm: After the generation of key streams of both versions and attacking key streams the next step is to evaluate the effect of GD attack on both versions. For this purpose an algorithm is designed which compares the original key streams with the attacking key streams. Working of this comparison algorithm is divided into following steps:

Step I
Take input of keystream from Original file and store it in array of size 8.

Original File
ABCDEFHT
EFDHVFGT
JKLMASDE

Array 1

A	B	C	D	E	F	H	T

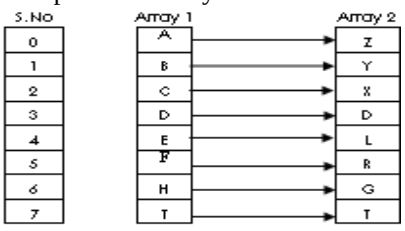
Step II
Take input of keystream from Attack file and store it in array of size 8.

Attack File
ZYXDLRGT
KGJTEADW
BNXCFSRT

Array 2

Z	Y	X	D	L	R	G	T

Step III
Compare both arrays index wise.



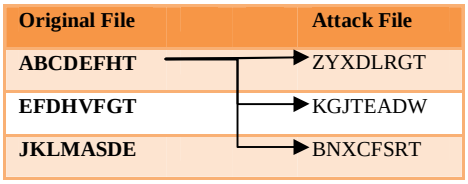
Comparison is done in a way that if the value at index 1 of original array is similar to the value of index 1 of attacking array, it will store '1'in resulting file otherwise '0' will be stored. It is shown in Table 2.

Table 2: Comparison of Key streams

Index	Value at index of array 1	Matching (Equal/not equal)	Value at index of array 2	Result
0	A	!=	Z	0
1	B	!=	Y	0
2	C	!=	X	0
3	D	= =	D	1
4	E	!=	L	0
5	F	!=	R	0
6	H	!=	G	0
7	T	= =	T	1

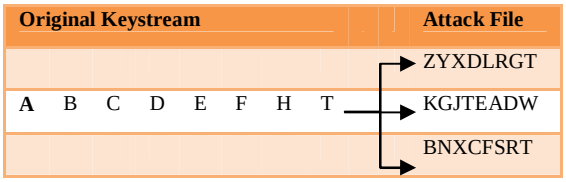
Step IV

Read all the key streams from Attack file one by one and compare them with the original keystreram which is taken as input in step I.

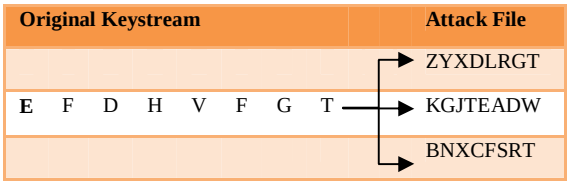


Step V

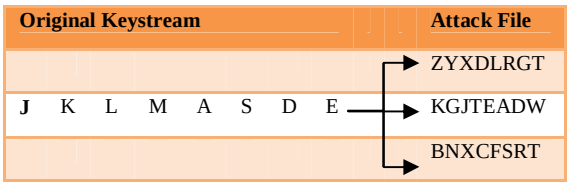
Store the resulting bits in another file for further calculations. This procedure will continue until all attacking key streams are being compared with first keystream of original file.



Comparison of first keystream with all attacking key streams



Comparison of second keystream with all attacking key streams



Comparison of third keystream with all attacking key streams

Step VI

Read the next keystream from original file when first keystream of Original file get compared with all Attacking key streams.

The algorithm will execute in this fashion until all key streams of Original file is being compared with each and every keystream of Attack file.

EXPERIMENTAL ANALYSIS

The whole experimental analysis is divided into two phases. The parameters of both phases are described in Table 3.

Table 3: Parameters of Phase I and Phase II

No.of Experiments	n	10
No.of Attacks in each Exp	m	10
No.of Guesses in each Attack	g	50
No.of Keystreams in Phase I	k	5
No.of Keystreams in Phase I	K1	16

In each phase 'n' experiments are performed. Each experiment has 'm' attacks, thus 'n' experiments have n*m attacks (i.e. 10 *10=100). Since one attack contains 'g' guesses. Therefore 'm' attacks will have g*m guesses (i.e. 50*10=500). And 'n' experiments have n*g*m guesses in total (i.e.10*50*10=5000).

As number of keystreams generated in Phase I against each guess is 'k' so total number of keystreams generated in Phase I are k*n*g*m (i.e. 5*10*50*10=25000)

And in Phase II 'k1' keystreams are generated against each guess thus total number of keystreams generated in Phase II are k1*n*g*m (i.e.16*10*50*10=80000)

Results of Phase I: The experimental results show that in n*m attacks 250442 similarities in Original SNOW 2.0 and 248719 similarities in Modified SNOW 2.0 have been trapped. And the average percentage of attack on Original SNOW 2.0 and Modified SNOW 2.0 is 6.26105 and 6.21795 respectively. This evaluates that Original SNOW 2.0 has been more affected by Guess and Determine attack.

Results of each experiment of Phase I are shown in Table 4.

Table 4: Results of Phase I

Experiments	**Original SNOW 2.0**	**Modified SNOW 2.0**
	No of Similarities	**No of Similarities**
1	25181	24980
2	25191	24524
.	25136	24893
.	24903	24982
.	25257	25014
.	25333	24962
.	24785	24875
.	24958	25221
.	24766	24607
n	24932	24661
SUM	250442	248719

The graphical representation of results of Phase I are shown in Fig 4.

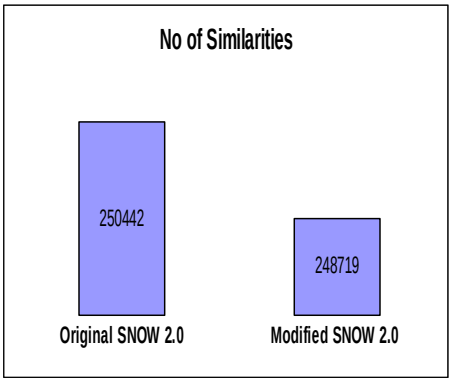


Fig 4: Graphical representation of Phase I

The average percentage of each experiment of Phase I is shown in Table 5.

Table 5: Average Percentage of Attack in Phase I

Experiments	Original SNOW 2.0	Modified SNOW 2.0
	Percentage of Attack	Percentage of Attack
1	6.29525	6.245
2	6.29775	6.131
.	6.284	6.22325
.	6.22575	6.2455
.	6.31425	6.2535
.	6.33325	6.2405
.	6.19625	6.21875
.	6.2395	6.30525
.	6.1915	6.15175
n	6.233	6.16525
Average Percentage	6.26105	6.217975

The graphical representation of average percentage of Phase I is given in Fig 5.

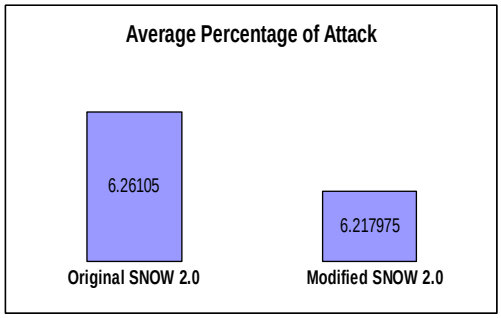


Fig 5: Average Percentage of Phase I

Results of Phase II: Experimental results illustrate that in 'n' experiments 2557588 and 2560859 similarities have been found in Original SNOW 2.0 and Modified SNOW 2.0 respectively. Also the average percentage of Original SNOW 2.0 is 6.244111 and of Modified SNOW 2.0 is 6.248638.

It is obvious that Original SNOW 2.0 has low average percentage. Hence Original SNOW 2.0 has more resistance against Guess and Determine attack.

Results of each experiment of Phase II are shown in Table 6.

Table 6: Results of Phase II

Experiments	Original SNOW 2.0	Modified SNOW 2.0
	No of Similarities	No of Similarities
1	256549	256627
2	255765	256492
.	255025	255293
.	256403	256068
.	255282	256534
.	255094	256007
.	255572	255759
.	255881	256171
.	256231	255547
N	255786	256361
SUM	2557588	2560859

The graphical representation of results of Phase I are shown in Fig 6.

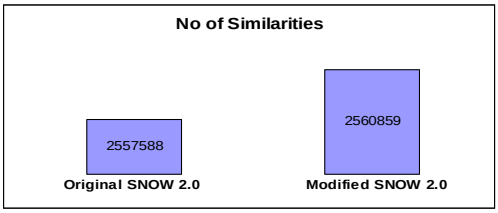


Fig 6: Graphical representation of Phase I

The average percentage of each experiment of Phase II is shown in Table 7.

Table 7: Average Percentage of Attack in Phase I I

Experiments	Original SNOW 2.0	Modified SNOW 2.0
	Percentage of Attack	Percentage of Attack
1	6.263403	6.265308
2	6.244263	6.262012
.	6.226196	6.232739
.	6.259839	6.25166
.	6.232471	6.263037
.	6.227881	6.250171
.	6.239551	6.244116
.	6.247095	6.254175
.	6.25564	6.204346
N	6.244775	6.258813
Average Percentage	6.244111	6.248638

The graphical representation of average percentage of Phase II is given in Fig 7.

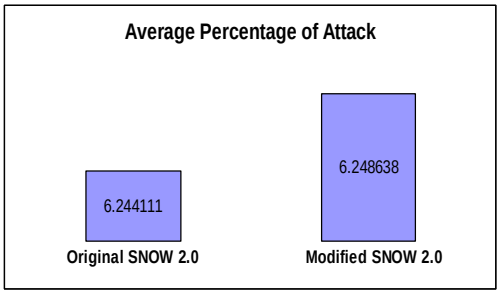


Fig 7: Average Percentage of Phase II

CONCLUSION

Guess and Determine attack has been applied on two versions of SNOW (i.e. Original SNOW 2.0 and Modified Version of SNOW 2.0). The attack consists of two phases. Phase I conclude that, for small number of keystreams modified version of SNOW 2.0 has more resistance against Guess and Determine attack. And Phase II concluded that, for large number of keystreams, SNOW 2.0 becomes more secure. As a result of the experimental work and analysis, it is concluded that, if the plaintext that has to be encrypted is of small amount, modified Version of SNOW 2.0 should be used. And if large data set has to be encrypted, original SNOW 2.0 should be recommended.

REFERENCES

1. Menezes,.A., P. van Oorschot, and S. Vanstone,1996.Handbook of Applied Cryptography CRC Press.For further information,see www.cacr.math.uwaterloo.ca/hac
2. Phil Zimmermann, 0000. Introduction to Cryptography. Available at, www.eie.fceia.unr.edu.ar/ftp/Comunicaciones/AN%20INTRODUCTION%20TO%20CRYPTOGRAPHY.PDF
3. Ekdahl,.P.,2003.On LFSR based Stream Ciphers - Analysis and Design, Ph.D. Thesis, Dept. of Information Technology, Lund University.
4. Wikipedia, 2008.Standard Error - Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/wiki/Cryptanalysis
5. Ekdahl,P. and T. Johansson, 2000.SNOW-A new stream cipher, Proceedings of First Open NESSIE Workshop, KU-Leuven.
6. Ekdahl. P. and T. Johansson, 2002. A new version of the stream cipher SNOW, Proceedings of Selected Areas in Cryptography (SAC) 2002, Springer, 2002.Available at, citeseer.ist.psu.edu/ekdahl02new.html
7. Hawkes, P. and G. G. Rose, 2002. Guess-and-determine attacks on SNOW. In K. Nyberg and H. Heys, editors, *Selected Areas in Cryptography*- SAC 2002, volume 2595 of *Lecture Notes in Computer Science*, pages 37-46. Springer Verlag, 2002.
8. Coppersmith.D. and S.Halevi, C.Jutla. 2002.Cryptanalysis of Stream Ciphers with Linear Masking. In M. Yung, editor, Advances in Cryptology- CRYPTO 2002, volume 2442 of *Lecture Notes in Computer Science*, pages 515-532. Springer Verlag, 2002.
9. Ullah, S.I: Tabassam, A.A,; Khiyal, S.H,0000.Cryptographic weaknesses in modern stream ciphers and recommendations for improving their security levels.
10. Alexander Maximov, 0000.Some Words on Cryptanalysis of Stream Ciphers, Ph.D. Thesis, Lund University, 2006.
11. U.S. National Institute of Standards and Technology. 12 May 2008. Available from: http://www.nist.gov/dads/HTML/finiteStateMachine.html
12. Erdem Yilmaz, 2004.Two Versions of the Stream Cipher SNOW, A Thesis Submitted to the Graduate School of Natural And Applied Sciences of Middle East Technical University, in partial fulfillment of the Requirement for Degree of Master of Science, December 2004.
13. Watanabe,D. A.,Biryukov, and C. de Canni`ere. 2003 A distinguishing attack on SNOW 2.0 with linear masking method. In M. Matsui and R. Zuccherato, editors, Selected Areas in Cryptography – SAC 2003, Lecture Notes in Computer Science, pages 222–233. Springer-Verlag, 2003.
14. Ahmadi,H. and T. Eghlidos, 2005. Advanced Guess and Determine Attacks on Stream Ciphers, IST 2005, pp. 87-91.
15. Ullah, S.I: Tabbasum, Naz : Tarranum, Khiyal, S.H,. Traceable Bit Streams in SNOW 2.0 Using Guess and Determine Attack, World Applied Sciences Jounal 11 (2): 190-195 2010. ISSN 1818-4952.

Tarranum Naz is a graduate from Dept. Computer Science, Fatima Jinnah Women University Pakistan. She is an active researcher in the field of information security and cryptography in Computer Sciences Dept. Fatima Jinnah Women University, Pakistan. She can be contacted at tarranumpk@yahoo.com, Fatima Jinnah Women University, Rawalpindi Pakistan.

Muhammad Tahir works in Department of Computer Engineering Sir Syed University of Engineering and Technology, Karachi. Pakistan. His research interests are in the field of information security and cryptography. He can be contacted tahirfattani@gmail.com, Sir Syed University of Engineering and Technology, Karachi. Pakistan

Aihab Khan works in Department of Computer Sciences Fatima Jinnah Women University, Pakistan. His research interests are in the field of Data mining, Data Warehousing as well as Information Security. He can be contacted at aihabkhan@yahoo.com, Fatima Jinnah Women University, Rawalpindi Pakistan.