Modifying LFSR of ZUC to Reduce Time for Key-Stream Generation

Raja Muthalagu[1] and Subeen Jain[2]

[1]*Assistant Professor, Department of Electrical and Electronics Engineering, BITS, Pilani, Dubai campus, Dubai, UAE*
[2]*Student, Department of Electronics and Communication Engineering, BITS, Pilani, Dubai campus, Dubai, UAE*
E-mail: raja.m@dubai.bits-pilani.ac.in; jainsub9@gmail.com

Received 15 June 2017; Accepted 3 July 2017;
Publication 4 August 2017

Abstract

ZUC is stream-cipher which generates 32-bit key-stream by using 128-bit initial key and 123-bit initial vector. It encrypts the plaintext data to produce cipher-text data. The 128-EEA3 encryption and 128-EIA3 authentication algorithms are based on ZUC which are specified for use in 3GPP cellular communications systems. The algorithm is divided in three stages: LFSR (Linear Feedback Shift Register), Bit Reorganization (BR) and, Non-Linear Function. In this paper, we are going to discuss about our modifications proposed for LFSR along with small change in operation of Non-linear Function which can reduce time for generating key-stream. Many attacks based on weakness of LFSR due to its linearity are proposed which in turn makes ZUC susceptible to various attacks based on LFSR. As we know in the structure of LFSR, non-linearity is provided in last block of LFSR by feedback operation while all other being clocked with previous value of LFSR, so we have introduced bit-shifting and circular shift operations on few blocks of LFSR output of which will be taken as input to other blocks of LFSR.

Keywords: LFSR, Key-Stream, ZUC, Security, NIST Statistical Test.

Journal of Cyber Security, Vol. 5_4, 257–268.
doi: 10.13052/jcsm2245-1439.541

1 Introduction

The rapid growth of mobile communications has increased the requirement of having secure network/communication between the users. Multiple ways of attacking or hacking a network are used by an attacker. As the wireless mode of communications provides feasibility and ease to the users, the security of information being exchanged within two users or group of users is always at threat. Many algorithms have been proposed which are used for different encryption purposes of which some have proved resistant towards attacks while some are attacked and hence, proved weaker by attackers. For having secure network, encryption services and algorithms involved need to be robust to provide end-to-end secure transmission of data among various users. It poses a challenge for designer to design highly secure and attack-resistant algorithm for encryption of data. Besides designing a highly-secure algorithm, it is generally desired to have its simple hardware implementation and less complex structure which can encrypt plain-text quickly.

Algorithms like SNOW-3G and ZUC are seen as one among the most robust algorithms designed for mobile encryption services. These are used in 4G/LTE services for providing encryption of data and integrity to it. Both of these produce 32-bit key-streams which are used for encrypting the plaintext data and hence, generating cipher-text. The two algorithms have different stages for functioning but have LFSR as the common feature in both algorithms. Since the feedback in LFSR is given to s_{15} register only and all remaining 15 shift registers being clocked with values of previous registers, attack could possibly be made. Many attacks related to LFSR are proposed in [4–6, 9–11] which are scan-based attack, power analysis attack, algebraic attack, fast correlation attack, posing threat to algorithms especially based on producing stream ciphers. Attack related to cache-timing attack [3] is also there which have stream-ciphers as the target. According to it, on all word-based LFSR implementations that use lookup tables to speed up multiplications a full attack can be done making threat to algorithm. In some of the works and papers, it is suggested to look for introducing non-linear functions/operations (non-linear combiners or, non-linear filters) in LFSR or different alternatives to reduce chances of attack on algorithm having LFSRs in them. As feedback is provided to only last LFSR, non-linearity can be observed only in that register. In this paper, we have focused towards introducing feedback operations in some blocks of the LFSRs of ZUC algorithm while making small modification in non-linear function (F) of chosen algorithm. Besides this, we have also reduced time taken by algorithm to generate key-stream, after modifying LFSR, compared to original ZUC algorithm.

This paper is organized in following manner: Section 1 provides brief overview of ZUC. The three stages of ZUC are discussed along with Initialization Mode and Work Mode of LFSR. Section 2 discusses about our proposed work on modifications in LFSR. Section 3 discusses about results obtained after making modifications in LFSR. It shows comparison of our work with original ZUC in terms of time and speed for generation of 32-bit key-stream. Section 4 gives conclusion of our work done in this paper.

2 Overview of ZUC

ZUC is word-oriented stream cipher which takes 128-bit initial key and 128-bit initial vector (IV) as input to produce 32-bit key-stream. The generated key-stream is then used for encrypting the plaintext and hence, produces cipher-text. It contains two stages for generation of key-stream which are Initialization Mode and work Mode. In Initialization stage 32-bit word produced is discarded and then algorithm goes in work mode to generate key-stream as an output. Figure 1 shows structure of ZUC.

As given in [1], structure of ZUC is divided in 3 parts which are explained briefly below:

2.1 LFSR (Linear Feedback Shift Register)

It comprises of 16 registers each holding capacity to store 31-bit of data. With each clock pulse data from one register is shifted to next register and feedback is provided to last register s_{15}. Tapping of five shift registers is done for providing feedback operation using addition modulo $(2^{32} - 1)$. It has 2 modes: Initialization Mode and Work Mode.

In Initialization Mode, input is given from Non-Linear function stage (F) and, no output is generated. After LFSR is initialized, it goes in work mode where it generates 32-bit key-streams. This output or key-streams produced are then used for encrypting plain-text data. Figure 1 shows Initialization mode and Figure 2 [1] shows Work Mode of LFSR.

2.2 Bit Reorganization (BR)

It comprises of 4 32-bit blocks and values in it come from different LFSRs as shown in Figure 2. Each 32-bit block is made from concatenation of 2 16-bit data from LFSRs as shown in Figure 2. These blocks are then used for different operations in next stage of algorithm.

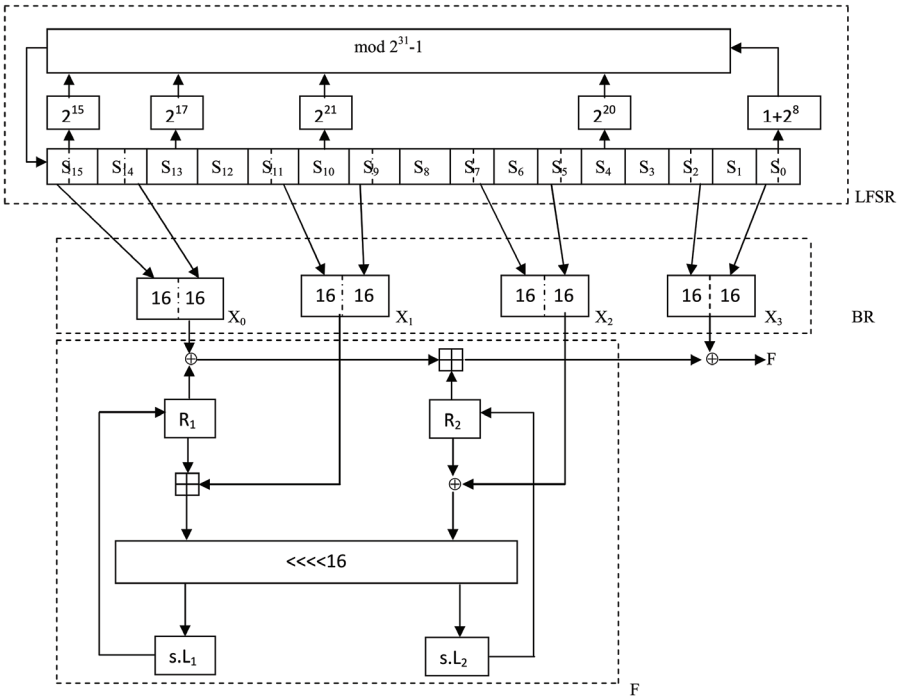

Figure 1 ZUC algorithm [1].

2.3 Non-Linear Function (F)

It has 2 32-bit registers R_1 and R_2. Non-Linear function receives input from 2 of the 4 BR blocks which is shown in Figure 2. Input to registers is given from S (L_1) and S (L_2) where S is two different S-boxes and, L_1 and L_2 are Linear Transforms. Values of S (L_1) and, S (L_2) are given below:

$$R_1 = S\left(L_1\left(W_{1L} \,||\, W_{2H}\right)\right) \tag{1}$$

$$R_2 = S\left(L_2\left(W_{2L} \,||\, W_{1H}\right)\right) \tag{2}$$

These values serve as input values to two registers R_1 and R_2.

3 Proposed Work

In our work, we have made few changes in LFSR used in ZUC algorithm which is shown in Figure 2. In the modified structure of LFSR, we have

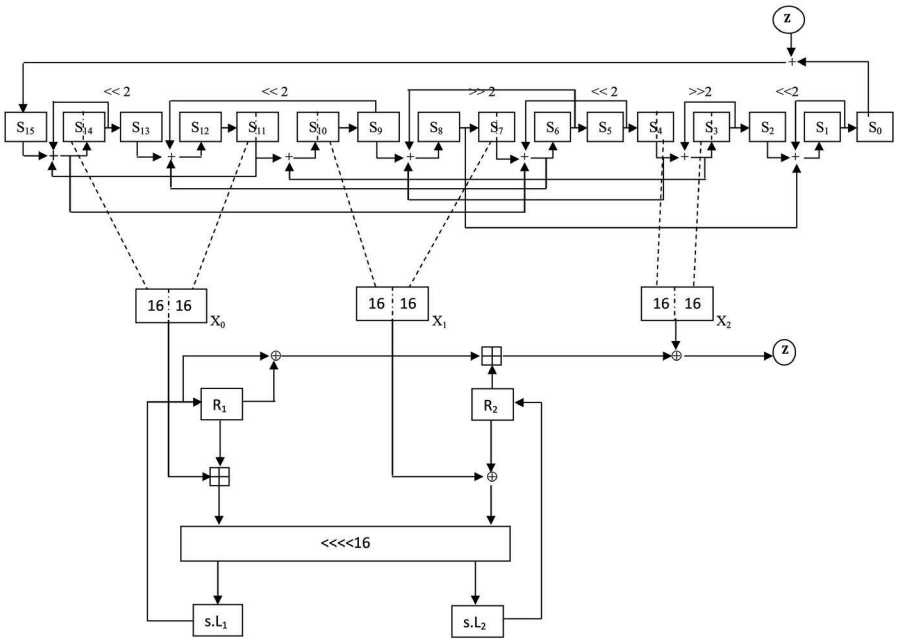

Figure 2 Proposed structure of LFSR in Initialization Mode in ZUC algorithm.

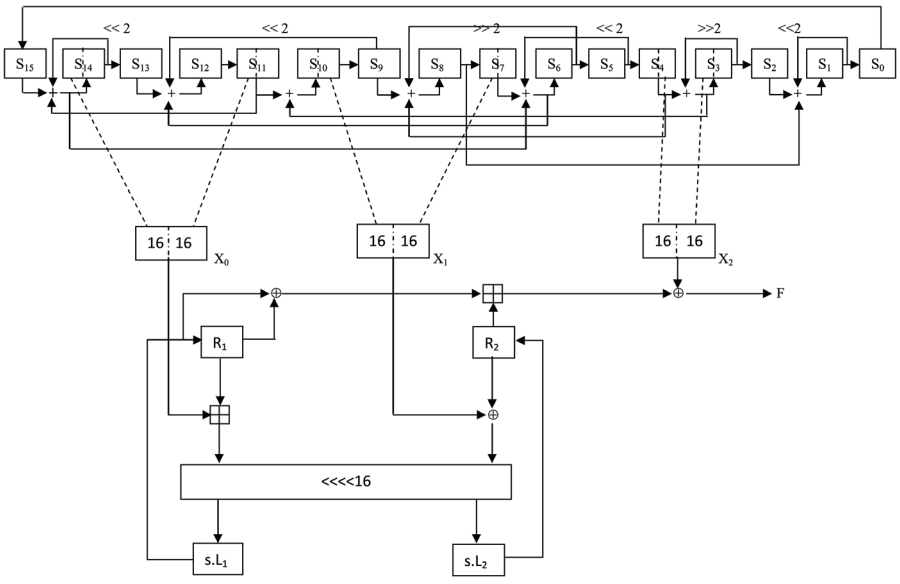

Figure 3 Proposed structure of LFSR in Work Mode in ZUC algorithm.

provided feedback operations to seven Shift Registers by using addition, bit shift operations of s-blocks and, right and left circular shifts of bits coming from different s-blocks. The number of 32-bit s-blocks is kept same as in ZUC algorithm. The process of initialization of LFSR is same as 32 times after which it goes in work mode and hence, generates 32-bit key-streams. In our feedback operations, we have used right circular shift operations four times when data is taken as input from s_{11}, s_8, s_4, s_3 s-blocks to s_{14}, s_1, s_8 and, s_{10} s-blocks while left circular shift operations are used three times when input comes from s_{14}, s_9, s_6 s-blocks to s_6, $s_4$and, s_{12} s-blocks. Apart from this, normal left and right shifting of bits is also done which is shown in Figure 2 (result of right shift of s-blocks is added between [s_{12}, s_{13}] and, [s_9, s_8] s-blocks while result of left shift of input data is added between [s_{15}, s_{14}], [s_{13}, s_{12}], [s_7, s_6] and, [s_2, s_1] s-blocks).

The 32-bit data blocks used in Bit Reorganization is same as in ZUC constructed by bit shifting operations but inputs are used from different 32-bit s-boxes. We have used here three 32-bit blocks. Data for each block is shown below:

$$X_0 = (upper\ 16\text{-}bits\ of\ LFSR\_S_{14}) \mid (upper\ 16\text{-}bits\ LFSR\_S_{11}) \quad (3)$$

$$X_1 = (lower\ 16\text{-}bits\ of\ LFSR\_S_{10}) \mid (upper\ 16\text{-}bits\ of\ LFSR\_S_7) \quad (4)$$

$$X_2 = (lower\ 16\text{-}bits\ of\ LFSR\_S_4) \mid (upper\ 16\text{-}bits\ of\ LFSR\_S_3) \quad (5)$$

Input to non-linear function box (F) is provided by two Bit Reorganization boxes, X_0 and, X_1. Functioning of this non-linear stage is same as in ZUC only small change in use of S-box in registers R_1 and R_2 and, output from non-linear function, W, is made.

$$W = (R_1 \oplus s.L_1) \boxplus R_2 \quad (6)$$

$$\text{and, } R_1 = MAKEU32\ (S21\ [(u >> 24)\ \&\ 0xFF],\ S20\ [(u >> 16)\ \&\ 0xFF],\ S21\ [(u >> 8)\ \&\ 0xFF],\ S20\ [u\ \&\ 0xFF]) \quad (7)$$

$$R_2 = MAKEU32\ (S21\ [(v >> 24)\ \&\ 0xFF],\ S20\ [(v >> 16)\ \&\ 0xFF],\ S21\ [(v >> 8)\ \&\ 0xFF],\ S20\ [v\ \&\ 0xFF]) \quad (8)$$

By making changes in it our focus was to reduce time consumed by ZUC algorithm in generation of key-streams by introducing any possible modifications. Results related to modified algorithm are shown through table and

graph and, comparison has been made with original ZUC algorithm. As we can see that non-linearity is produced only in last s-block (s_{15}) while all other s-blocks being updated with previous s-block value, we made changes in values clocked in LFSR by providing feedback operations in few blocks so as to introduce some non-linearity to those LFSRs. Since the process of bit shifting is faster than using any other operation, we considered to make change in LFSR by using bit-shifting and circular bit-shift operations. The new structure has become little complex but it can produce key-streams faster than original ZUC. Attacks related to LFSR are reported in [6–8] which implies that it can lead to breakdown of algorithm. Reports and theoretical work discuss the strength of LFSR and suggest that LFSR is susceptible to attacks due to its linearity. Since LFSR is also called as linear recurrence generator, it means values generated initially will occur after certain period. If a certain feedback polynomial P is used for generating a particular sequence, same sequence could be regenerated by another polynomial multiple of P. This makes a threat to algorithms that uses LFSRs for producing key-streams and, attacks like divide-and-conquer attack, fast-correlation attack, distinguishing attack, scan-based attack can be implemented. For this, we came up with new design in LFSR which can reduce chances of linear attacks and with this modification generation of key-stream will be faster. The proposed design of LFSR with small other modifications is shown in Figure 2.

4 Results

After making modifications, we made a comparison of Modified and Original ZUC in terms of time taken to generate key-stream and, Speed for generating the key-stream. Values are provided in respective tables and graphs showing comparison of both are given below. The values listed in tables for Modified and Original ZUC are best possible obtained one. Testing was software-based (MobaXterm linux application), done on Intel (R) Core(TM) i7-4770 CPU @ 3.4 GHz.

From the Tables 1, 2 and Figures 4, 5, we can see that modified structure of LFSR when used in ZUC consumes less time and generates key-streams with more speed than original ZUC. Besides making comparison in terms of time and speed, the modified algorithm was tested using NIST [2] statistical test suite where all 15 tests were passed. The test consists of Frequency (Mono-bit) Test, Frequency Test within a Block, Runs Test, Tests for the Longest-Run-of-Ones in a Block, Binary Matrix Rank Test, Discrete Fourier Transform (Spectral) Test, Non-overlapping Template Matching Test,

Table 1 Original ZUC algorithm

Key-Stream Length	Time for Key-Stream Generation	Speed for Key-Stream Generation
1. $4*10^7$	1 second	40 Mbps
2. $5*10^7$	2 seconds	25 Mbps
3. $6*10^7$	2 seconds	30 Mbps
4. $7*10^7$	2 seconds	35 Mbps
5. $8*10^7$	3 seconds	26.6667 Mbps
6. $9*10^7$	3 seconds	30 Mbps
7. $1*10^8$	4 seconds	25 Mbps
8. $2*10^8$	8 seconds	28.5714 Mbps
9. $3*10^8$	11 seconds	27.2727 Mbps
10. $4*10^8$	16 seconds	25 Mbps
11. $5*10^8$	19 seconds	26.31578 Mbps
12. $6*10^8$	24 seconds	25 Mbps
13. $7*10^8$	28 seconds	25 Mbps
14. $8*10^8$	32 seconds	25 Mbps
15. $9*10^8$	37 seconds	24.3243 Mbps

Table 2 Modified structure of LFSR in ZUC algorithm

Key-Stream Length	Time for Key-Stream Generation	Speed for Key-Stream Generation
1. $4*10^7$	1 second	40 Mbps
2. $5*10^7$	1 second	50 Mbps
3. $6*10^7$	1 second	60 Mbps
4. $7*10^7$	1 second	70 Mbps
5. $8*10^7$	1 second	80 Mbps
6. $9*10^7$	2 seconds	45 Mbps
7. $1*10^8$	2 seconds	50 Mbps
8. $2*10^8$	5 seconds	40 Mbps
9. $3*10^8$	7 seconds	42.8571 Mbps
10. $4*10^8$	9 seconds	44.4444 Mbps
11. $5*10^8$	12 seconds	41.6667 Mbps
12. $6*10^8$	14 seconds	42.8571 Mbps
13. $7*10^8$	17 seconds	41.1764 Mbps
14. $8*10^8$	19 seconds	42.1052 Mbps
15. $9*10^8$	22 seconds	40.9091 Mbps

Overlapping Template Matching Test, Universal Statistical Test, Linear Complexity Test, Serial Test, Approximate Entropy Test, Cumulative Sums Test, Random Excursions Test, and Random Excursions Variant Test. This statistical suite is used for testing of randomness produced in designed/modified algorithm. The p-value generated, as per NIST rules and guidelines, for each test, was greater than 0.01.

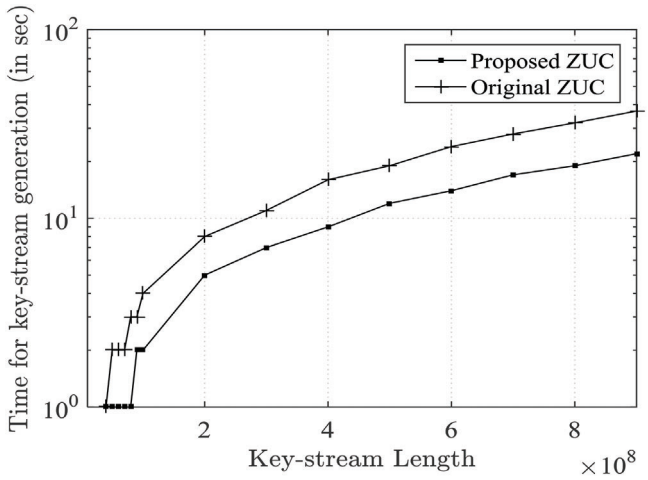

Figure 4 Comparison of time (in sec) Modified and Original ZUC.

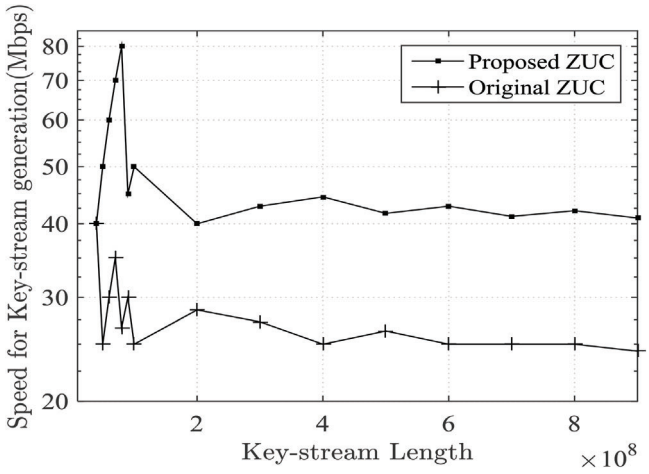

Figure 5 Speed comparison (in Mbps) for Modified and Original ZUC.

5 Conclusion

In this paper, we have proposed changes in LFSR which could introduce non-linearity in it and it is shown above that the new structure consumes less time and has high speed of key-stream generation compared to original ZUC. We can further propose more modification in ZUC by working again on LFSR which can produce non-linearity in it and the one which can produce

from LFSR longer number of runs with more random values having larger time period of recurrence. We can implement the same proposed structure of LFSR in other stream ciphers based on LFSR which can not only introduce non-linearity but can also reduce time consumed for key-stream generation.

References

[1] ETSI/SAGE (2011). *Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3. Document 2: ZUC Specification, Version: 1.6*. Sophia Antipolis: ETSI.

[2] National Institute of Standards and Technology [NIST] (2010). *Special Publication 800-22*. Gaithersburg, MD: National Institute of Standards and Technology.

[3] Leander G., Zenner, E., Hawkes, P. (2009). "Cache timing analysis of lfsr-based stream ciphers, cryptography and coding," in *Proceedings of the 12th IMA International Conference, Cryptography and Coding 2009*, (Cirencester, UK: IMA).

[4] Zou, M. H., Ma, K., Wu, K. J., Sha, H. M. (2014). Scan-based attack on stream ciphers: a case study on eSTREAM finalists. J. Comput. Sci. Technol. 29, 646.

[5] Liu Y., Wu K., and Karri R. (2011) Scan-based attacks on linear feedback shift register based stream ciphers. ACM Trans. Des. Autom. Electron. Sys. 16, 1–15.

[6] Burman, S., Mukhopadhyay, D., Veezhinathan, K. (2007). "LFSR based stream ciphers are vulnerable to power attacks," in *International Conference on Cryptology in India, Indocrypt 2007: Progress in Cryptology – Indocrypt*, eds K. Srinathan, C. P. Rangan, M. Yung (Springer: Berlin), 384–392.

[7] Muller, F., Peyrin, T. (2005). "Linear cryptanalysis of the TSC family of stream ciphers," in *Poceedings of the International Conference on the Theory and Application of Cryptology and Information Security ASIACRYPT 2005: Advances in Cryptology* (Berlin: Springer), 373–394.

[8] Lee, D. H. (2005). Algebraic attack on stream cipher. *Trends in Math.* 8, 133–143.

[9] SarbaniPalit, Bimal K. Roy, Arindom De, (2003). *A Fast Correlation Attack for LFSR-Based Stream Ciphers, International Conference on Applied Cryptography and Network Security, ACNS 2003: Applied Cryptography and Network Security* (Berlin: Springer), 331–342.

[10] Golić, J. (1994). "Linear cryptanalysis of stream ciphers," in *Fast Software Encryption – 1994, volume 1008 of Lectures Notes in Computer Science*, ed. B. Preneel (Springer: Berlin), 154–169.
[11] Courtois, N. (2003). "Fast algebraic attack on stream ciphers with linear feedback," *Advances in Cryptology – Crypto 2003, LNCS 2729*, ed. D. Boneh (Berlin: Springer-Verlag), 176–194.

Biographies

Raja Muthalagu received his B.Eng. Degree in Electronics and Communication Engineering from Anna University, Chennai, India, in 2005, the M.Eng. Degree in Digital Communication and Networking from Anna University, Chennai, India, in 2007, and Ph.D. in Wireless Communication from National Institute of Technology (NIT), Tiruchirappalli, India in 2014. He joined the Department of Electrical and Electronics Engineering, BITS, Pilani, Dubai Campus, in 2015, where he is currently a full Assistant Professor. He was a postdoctoral research fellow at ATMRI, Nanyang Technological University (NTU), Singapore during 2014–2015. He was a recipient of Canadian Commonwealth Scholarship Award-2010 for Graduate Student Exchange Program in the Department of Electrical and Computer Engineering, University of Saskatchewan, Saskatoon, SK, Canada and also he is a Visiting Scholar in the Department of Electrical and Computer Engineering, University of Saskatchewan, Saskatoon, SK, Canada during January 2011–June 2012. His research interests include orthogonal frequency division multiplexing (OFDM), multiple-input and multiple-output (MIMO) systems, and network security. He published his research papers in refereed international journals, and international and national conferences.

Subeen Jain is B.E. (Honors) student of Electronics and Communications engineering in BITS-Pilani Dubai campus. His areas of interest include security algorithms mainly related to mobile security and networking and, areas related to telecommunications.