International Journal of Emerging Engineering Research and Technology
Volume 3, Issue 11, November 2015, PP 63-70
ISSN 2349-4395 (Print) & ISSN 2349-4409 (Online)

WG Stream Cipher based Encryption Algorithm

Shrddha N Choudhary[1], K Suresh[2]

[1]*Department of ECE, Malla Reddy College of Engineering & Technology, Hyderabad, India (PG Scholar)*
[2]*Department of ECE, Malla Reddy College of Engineering & Technology, Hyderabad, India (Associate Professor)*

ABSTRACT

This paper presents two new hardware designs of the Welch–Gong (WG) −128 cipher, one for the multiple output WG (MOWG) versions, and the other for the single output version WG based on type−II optimal normal basis representation. The proposed MOWG design uses signal reuse techniques to reduce hardware cost in the MOWG transformation, whereas it increases the speed by eliminating the inverters from the critical path. This is accomplished through reconstructing the key and initial vector loading algorithm and the feedback polynomial of the linear feedback shift register. The proposed WG design multiple output encryption, decryption And single output encryption, decryption are designed and simulated by using xilinx13.2 tools

Keywords: Finite fields, linear feedback shift register (LFSR), pseudorandom key generators, stream ciphers, Welch–Gong (WG) transformation.

INTRODUCTION

SYNCHRONOUS stream ciphers are light weight symmetric-key cryptosystems. These ciphers encrypt a plain-text, or decrypt a cipher-text, by XORing the plain-text/cipher-text bit-by-bit with the generated key-stream bits. The key-stream bits are produced using a pseudorandom sequence generator (PRSG) and a seed (secret key). Stream ciphers are heavily used in wireless communication and restricted in resources applications such as 3GPP LTE-Advanced security suite [1], network protocols (Secure Socket Layer, Transport Layer Security, Wired Equivalent Privacy, and Wi-Fi Protected Access) [2], radio frequency identification (RFID) tags [3], and Bluetooth [4], to name some. Traditionally, many hardware-oriented stream ciphers have been built using linear feedback shift registers (LFSRs) and a filter/combiner Boolean function. However, the discovery of algebraic attacks made such a way of design insecure [5]–[8]. Many nonlinear feedback shift registers-based stream ciphers have been proposed in the eSTREAM stream cipher project [9], which have limited theoretical results about their randomness and cryptographic properties [3], and therefore, their security depends on the difficulty of analysing the design itself [3], [10]. In addition, the arrival of the 4G mobile technology has triggered another initiative for new stream ciphers [11], [12]. The randomness of the key streams generated by the 4G LTE cryptographic algorithms is, however, hard to analyse and, also, some weaknesses have been discovered [13]–[15].The Welch–Gong (WG)(29, 11) [29 corresponds to GF(229) and 11 is the length of the LFSR] is a stream cipher submitted to the hardware profile in phase 2 of the eSTREAM project [9]. It has been designed based on the WG trans-formations [16] to produce key bit-streams with mathematically proved randomness aspects. Such properties include balance, long period, ideal tuple distribution, large linear complexity, ideal two-level autocorrelation, cross correlation with anm -sequence has only three values, high nonlinearity, Boolean function with high algebraic degree, and 1-resilient [10], [17]–[19]. The revised version of the WG (29, 11) [9], [10] does not suffer the chosen initial value (IV) attack [20], [21]. The number of key-stream bits per run is strictly less than the number of key-stream bits required to perform the attack introduced in [22]. In addition, the WG cipher is secure against algebraic attacks [10], [19]. Therefore, the WG (29, 11) is secure and has the randomness properties that cannot be offered by other ciphers and, hence, it has a potential that the WG stream cipher will be adopted in practical applications.

Address for correspondence:

shrddha.choudhary@gmail.com

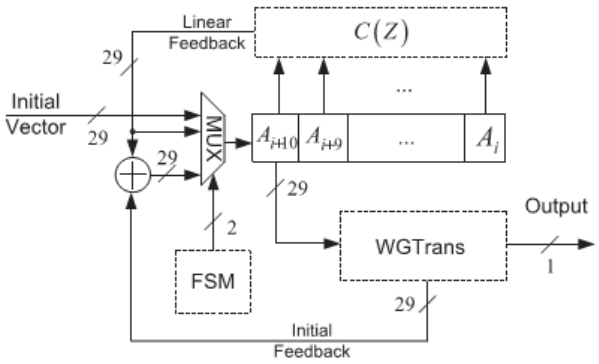
WG STREAM CIPHER

A synchronous stream cipher consists of a key stream generator which produces a sequence of binary digits. This sequence is called the running key or simply the key stream. The key stream is added (XORed) to the plaintext digits to produce the cipher text. A secret key K is used to initialize the key stream generator and each secret key corresponds to a generator output sequence. Since the secret key is shared between the sender and the receiver, an identical key stream can be generated at the receiving end. The addition of this key stream with the cipher text recovers the original plaintext. Stream ciphers can be divided into two major categories: bit-oriented stream ciphers and word-oriented stream ciphers. The bit-oriented stream ciphers are usually based on binary linear feedback shift registers (LFSRs) (regularly clocked or irregularly clocked) together with filter or combiner functions. They can be implemented in hardware very efficiently.

The WG cipher can be used with keys of length 80, 96, 112 and 128 bits. An initial vector (IV) of size 32 or 64 bits can be used with any of the above key lengths. To increase security, IVs of the same length as the secret key can also be used. WG cipher is a synchronous stream cipher which consists of a WG key stream generator. A simple block diagram of the WG key stream generator is shown in Figure the key stream produced by the generator is added bitwise to the plaintext to produce the cipher text. We now describe the WG key stream generator. As shown in figure the key stream generator consists of a 11 stage linear feedback shift register (LFSR) over F2^29. The feedback polynomial of the LFSR is primitive over F2^29 and produces a maximal length sequence (m-sequence) over F2^29 . This m-sequence is filtered a nonlinear WG transformation, F2^29 to produce the key stream. All the elements of F2^29 are represented in normal basis and all the finite field computations are in normal basis as well. The feedback polynomial of the LFSR is given by

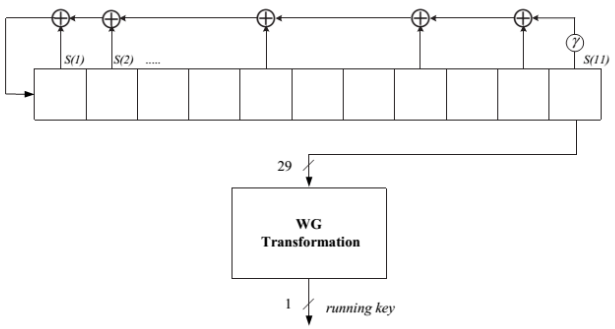
WG Generator

Main block diagram of wg generator

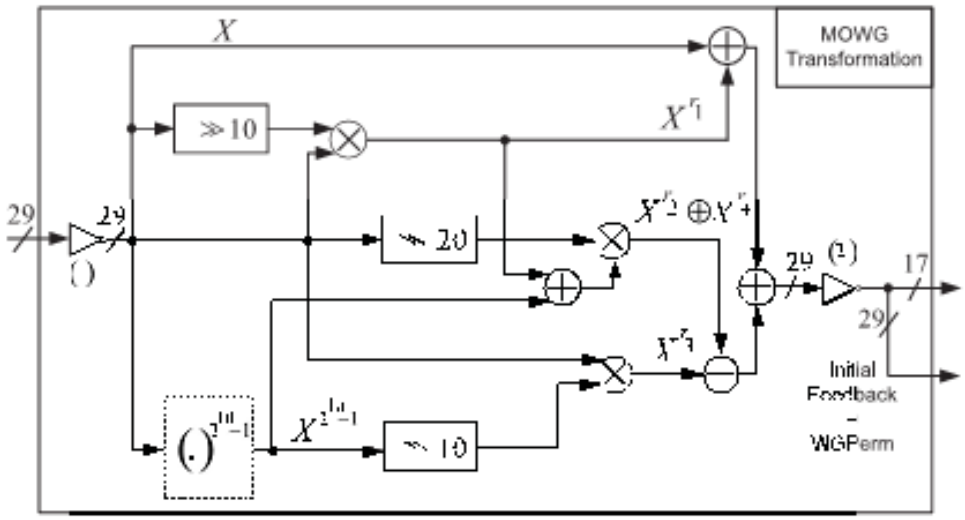


The WG/MOWG ciphers consist of three phases of operations:

1. key and iv loading phase
2. Key initialization phase
3. running phase



WG Transformation

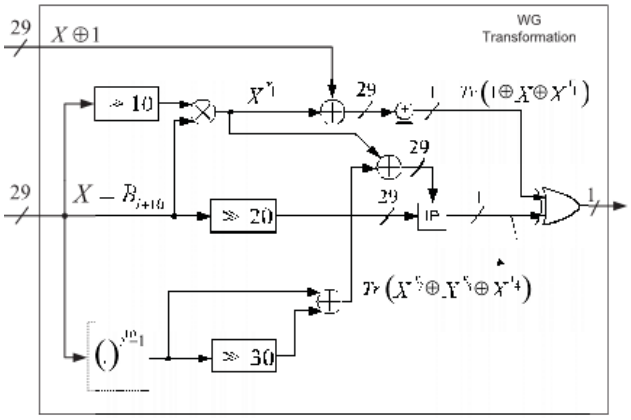


This section presents a hardware design of the MOWG (29, 11, 17) cipher, where 29 corresponds to GF (2^29), 11 is the number of stages in the LFSR, and 17 is the number of output bits. In this design, the MOWG transform uses seven multipliers, compared with eight multipliers in previous paper. In addition, in an attempt to improve the overall speed of the cipher, the LFSR is reconstructed to remove the inverters from the critical paths during the PRSG phase/initialization phase. In what follows, the reduced area MOWG transform design is first introduced, followed by presenting the LFSR/key and initial vector loading algorithm (KIA) algorithm changes for speed improvement. Then, the architecture of the finite-state machine (FSM) is discussed The hardware cost of the MOWG cipher is dominated by its transform's field multipliers. Any decrease in the number of these multipliers would minimize the area of the overall cipher. This subsection presents the architecture of the MOWG transform, where the number of field multipliers is reduced by 1 through signal reuse, compared with those in previous paper

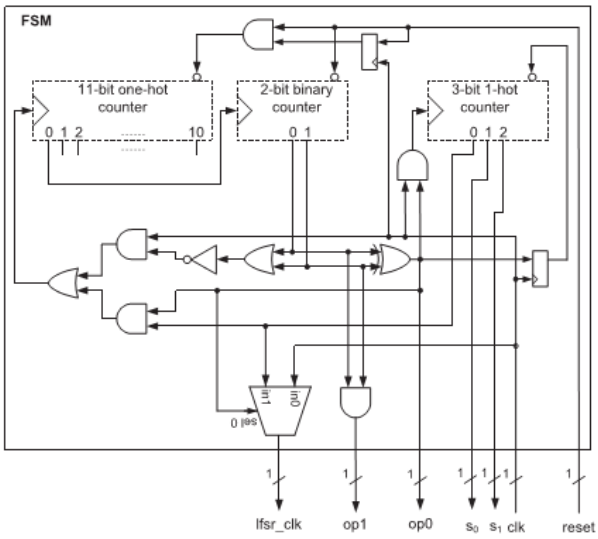
$$\text{WGPerm} = \left(1 \oplus X \oplus X^{2^k+1} \oplus X^{2^k(2^k-1)+1} \oplus X^{2^{2k}}\left(X^{(2^k+1)} \oplus X^{(2^k-1)}\right)\right).$$

Here, the overall proposed architecture of the MOWG (29, 11and 17) cipher is presented, as shown in Fig. In this figure, the FSM controls the input to the LFSR for each phase of operation. In the same figure, because of the bit-wise complement operator denoted by (a), the LFSR receives the complemented IV during the loading phase. Hence, after 11 clock cycles, the initial state of this LFSR,(B0,B1,...,B10), is basically the complement of the initial state of the LFSR in Fig. 1, i.e., Bi =Ai $\oplus$1, 0≤i <11.When the key initialization phase starts, the bit-wise XOR of the initial feedback and linear feedback applies to the input of the LFSR. Note that the Linear Feedback in Fig is generated which is equivalent to Bi =Ai $\oplus$1, 11≤i <33 (complement of corresponding one.. It is clear that the maximum delay of the MOWG transformation is reduced by an amount equivalent to the delay of two inverters, as compared with the one.

This section presents a method for the recovery of the Initial feedback signal through serialized computation. To accomplish the multiplication operations during this serial computation, the existing finite field multiplier that is used in generating the signal X r1 is used. The proposed scheme generates the initial feedback signal by serially computing it over three consecutive clock cycles. Denote this complete round of the serialized initial feedback computation (three clock cycles) as an extended key initialization round. In addition, denote the single clock cycle version of this computation (as in the MOWG design) as a simple round. Therefore, with serialization, the entire key initialization phase requires 3×22=66 clock cycles instead of 22 clock cycles (that is, 22 extended rounds instead of 22 simple rounds). It is noted that this only affects the key initialization phase without increasing the number of cycles required for the run phase



Modified FSM



Here, the new architecture and operation of the FSM are described. The architecture, which is shown in Fig, generates the new set of control signals lfsr_clk, s0, ands1.Theseare required for the serial computation of the initial feedback signal. Before each run of the cipher, the FSM resets its 11-bit one-hot counter to (1,0,...,0) and its 2-bit binary counter to(0,0)(where the leftmost and rightmost bits, within the brackets, denote the lowest output bit and the highest output bit of the corresponding counter, respectively). This is done through pulling down the reset inputs. When the reset signal is released, the 2-bit binary counter becomes ready. At the same time, the 11-bit one-hot counter's reset input stays pulled down for an extra clock cycle. This is due to the 1-bit Register connected to the input of the AND gate that drives its reset input. This assures that the(1,0,...,0) state of the 11-bit one-hot counter consumes a clock cycle at the beginning of the loading phase. After 11 clock cycles, from the release of the reset signal, the 11-bit one-hot counter returns to the (1,0,...,0) state. At this point, it triggers the clock input of the 2-bit binary counter. The 2-bit binary counter changes its state to (1,0), triggering the start of the key initialization phase. Then, the clk signal starts triggering the clock input of the 3-bit one-hot counter. The counting will, however, start one clock cycle later, when the output of the 1-bit Register connected to the 3-bit one-hot counter's reset input pulls up. This in turn assures that the 3-bit one-hot counter consumes one clock cycle, before incrementing its initial state of (1,0,0), at the start of the key initialization phase. During this phase, the first output bit of the 3-bit one hot counter drives the clock input of the 11-bit one-hot counter. Therefore, it takes 33 clock cycles for the 11-bit one-hot counter to complete 11 counts. Hence, it takes 33 clock cycles for the 2-bit binary counter to increment. Therefore, it requires 66 clock cycles for the 2-bit binary counter to increment twice to start the running phase. When the running phase starts, with the 2-bit binary counter's state at (1,1), the 11-bit and the 3-bit one-hot counters stop counting, as their clock inputs become idle. Notice that during the key initialization phase, the lfsr_clk is driven by the first output of the 3-bit one-hot counter. Hence, the LFSR shifts once every three clock cycles. The two signals s0

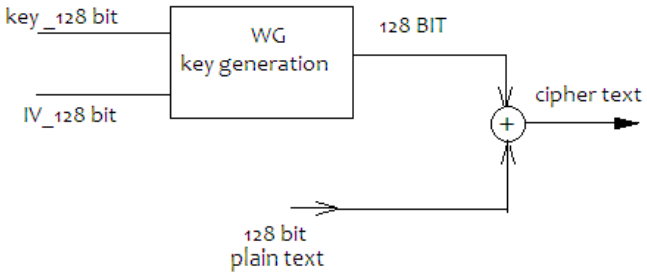
ands1 are derived from the 3-bit one-hot counter's output according to Table III. Notice that this table is realized without any additional hardware by settings0to be the second output and s1 to be the third output of the 3-bit one-hot counter, respectively. Therefore,(s0,s1)produces the three patterns of (0,0), (1,0),and(0,1) during the first, second, and third stages of an extended key initialization round, respectively. During the running phase, (s0, s1) will generate(0,0).The following shows how these patterns are used to accomplish the proper functionality in the key initialization phase as well as in the running phase. This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination

3-bit one-hot counter			s_1	s_0
bit 2	bit 1	bit 0		
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0

PROPOSED ALGORITHM

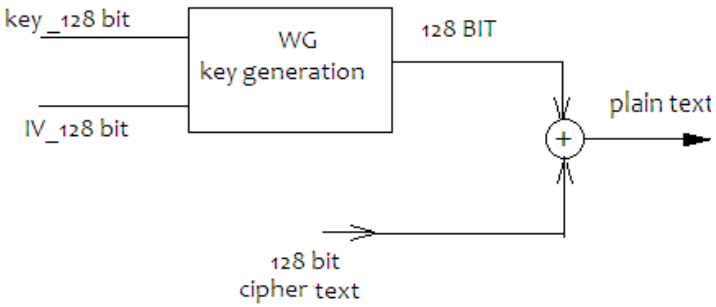
Wg Multiple Output Generation using Encryption and Decryption

Here we are giving key and initial vector to wg block then generated one 128 bit key .plaintext and generated key are xoring then generate ciphertext



ENCRYPTION

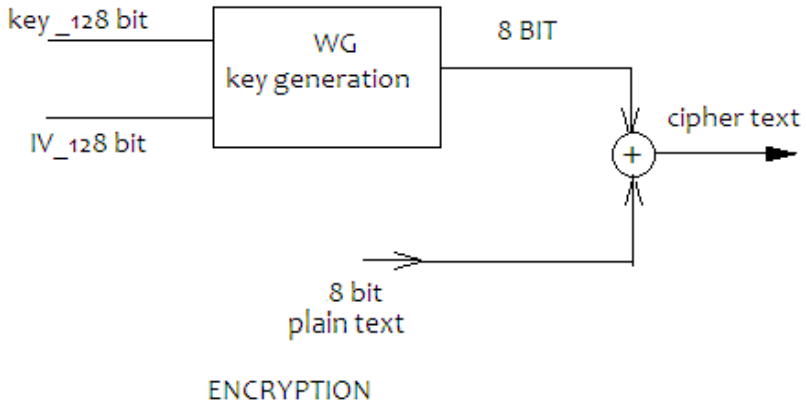
Here we are giving key and initial vector to wg block then generated one 128 bit key .ciphertext and generated key are xoring then generate plaintext



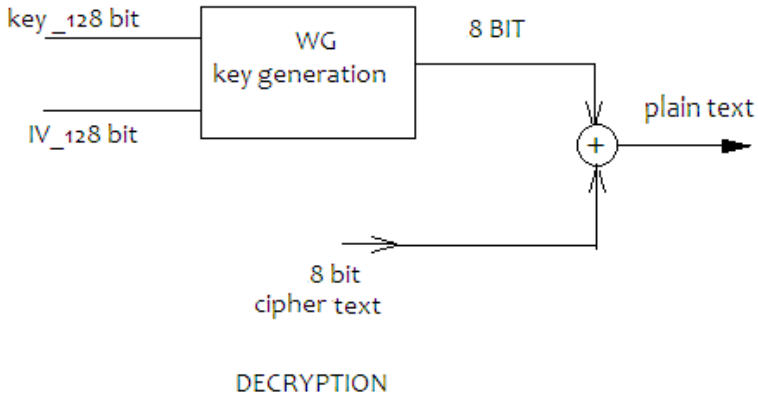
DECRYPTION

Wg Single Output Generation using Encryption and Decryption

Here we are giving key and initial vector to wg block then generated one 8 bit key .plaintext and generated key are xoring then generate ciphertext

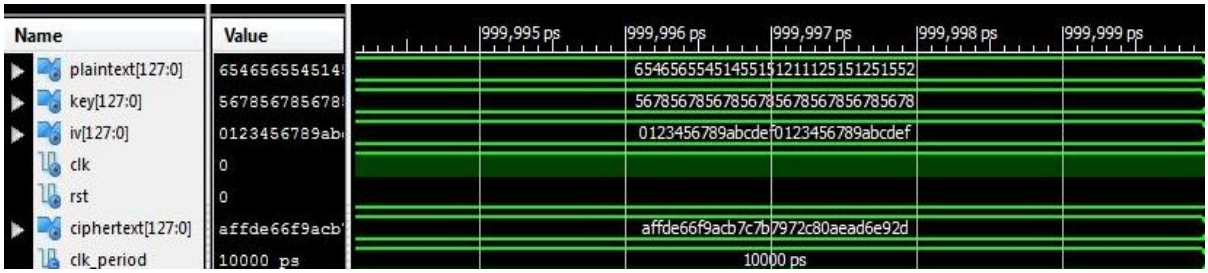


Here we are giving key and initial vector to wg block then generated one 128 bit key .ciphertext and generated key are xoring then generate plaintext

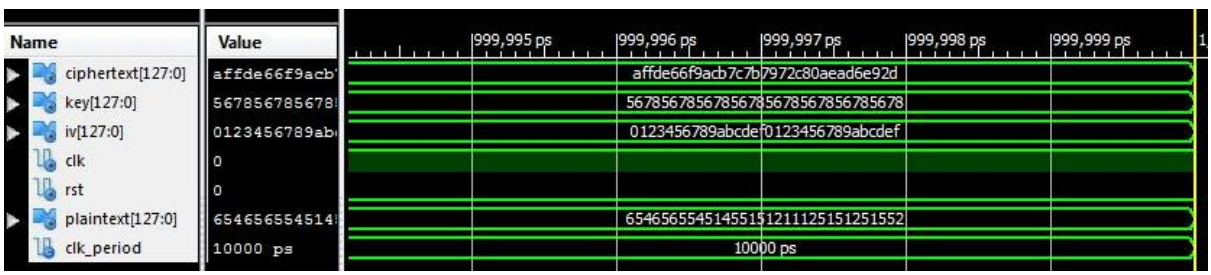


SIMULATION RESULTS

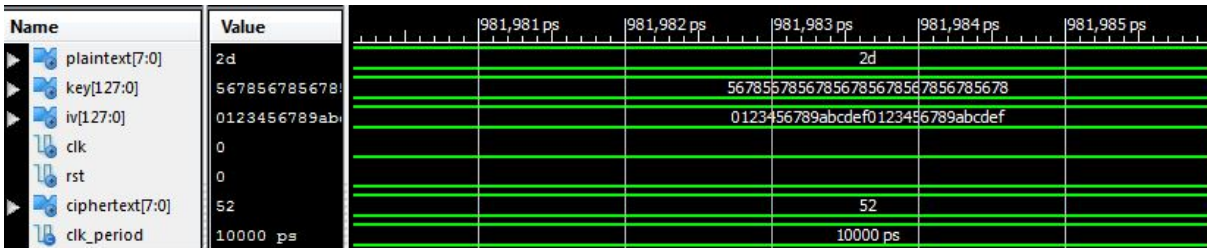
Multiple Output Encryption



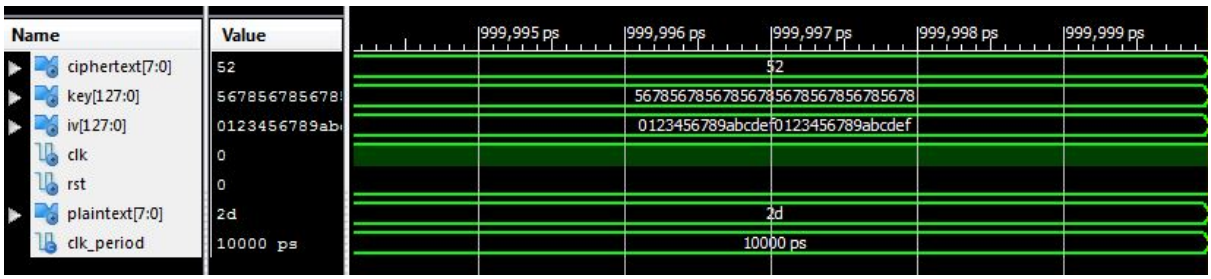
Multiple Output Decryption



Single Output Encryption



Single Output Decryption



CONCULSION

Two new designs for the MOWG (29, 11and 17) and the WG (29, 11) ciphers have been proposed. As compared with the MOWG, the proposed MOWG reduces the number of field multipliers in the transform by one through signal reuse. In addition, it increases the speed by eliminating two inverters delay from the critical path. This is accomplished by reconstructing the KIA and feedback polynomial of the LFSR. The proposed WG is an optimization of the proposed MOWG with trace (WG version).

REFERENCES

[1] S. Sen Gupta, A. Chattopadhyay, and A. Khalid, "HiPAcc-LTE: Anintegrated high performance accelerator for 3GPP LTE stream ciphers, "in Proc. 12th Int. Conf. Cryptol. India, 2011, pp. 196–215.

[2] S. Gupta, A. Chattopadhyay, K. Sinha, S. Maitra, and B. Sinha, "Highperformance hardware implementation for RC4 stream cipher, "IEEETrans. Comput., vol. 62, no. 4, pp. 730–743, Apr. 2013.

[3] Y.Luo, Q.Chai, G. Gong, and X. Lai, A lightweight stream cipherWG-7 for RFID encryption and authentication," in Proc. IEEE Global Telecommun. Conf., Dec. 2010, pp. 1–6.

[4] Bluetooth Special Interest Group. (2010, Jun.). Adopted Bluetooth Core Specifications, Core Version 4.0, Kirkland, WA, USA [Online].Available: https://www.bluetooth.org/

[5] N. Courtois, "Fast algebraic attacks on stream ciphers with linear feedback," in Proc. Advances in Cryptology—CRYPTO (Lecture Notesin Computer Science), vol. 2729. New York, NY, USA: Springer-Verlag, 2003, pp. 176–194.

[6] N. Courtois, "Algebraic attacks on combiners with memory and several outputs," in Information Security and Cryptology—ICISC (Lecture Notesin Computer Science), vol. 3506, C.-S. Park and S. Chee, Eds. New York, NY, USA: Springer-Verlag, 2005, pp. 3–20.

[7] W. Meier, E. Pasalic, and C. Carlet, "Algebraic attacks and decomposition of Boolean functions," in Advances in Cryptology—EUROCRYPT (Lecture Notes in Computer Science), vol. 3027, C. Cachin and J. Camenisch, Eds. New York, NY, USA: Springer-Verlag, 2004,pp. 474–491.

[8] F. Armknecht. (2004). On the Existence of Low-Degree Equations for Algebraic Attacks[Online]. Available: http://eprint.iacr.org/ [9] (2005). eSTREAM—The ECRYPT Stream Cipher Project

[Online]. Available: http://www.ecrypt.eu.org/stream/

[9] Y. Nawaz and G. Gong, "WG: A family of stream ciphers with designed randomness properties," Inf. Sci., vol. 178, no. 7, pp. 1903–1916, 2008.

[10] 3GPP TS 33.401 v11.0.1. 3rd Generation Partnership Project; Technical Specification Group Services and Systems Aspects; 3GPP System Architecture Evolution (SAE): Security Architecture, 3rd Generation Partnership Project (3GPP), France, Jun. 2011, [Online]. Available: http://www.3gpp.org

[11] 3rd Generation Partnership Project; Long Term Evaluation Release 10 and Beyond (LTE-Advanced); Proposed to ITU at 3GPP TSG RAN Meeting, 3rd Generation Partnership Project (3GPP), France, 2009, [Online]. Available: http://www.3gpp.org/.

AUTHORS' BIOGRAPHY

Shrddha N Choudhary, has done M.Tech in VLSI & Embedded Systems from Jawaharlal Nehru Technological University, Hyderabad, Telangala. She has been awarded degree of B.Tech in Electronics & Communication from Birla Institute of Technology, Mesra, Ranchi and Jharkhand. During her studies she has attended workshops related to U/VHF Transceivers at Electronics Corporation of India Limited (ECIL), Hyderabad and Characterization of Nano Thin Films at Defense Metallurgical Research Laboratory (DMRL), Hyderabad.

K Suresh, is presently working at Malla Reddy College of Engineering & Technology, as Associate professor in the Department of Electronics & Communication. His academics degrees include of M.Tech and B.Tech in the field of Electronics and Communication. He is humble and sincere in his work of guiding and teaching students. His time, patience and knowledge is greatly appreciated in this paper