



VLSI Implementation of Image Encryption and Decryption Using Reversible Logic Gates

¹Ganganamudi Tejaswini,²Gobburusrishanth,³Gundu Nikhila,⁴Korra Arun Kumar,⁵N.Pavithra

^{1,2,3,4} Student, Department of ECE, Narsimha Reddy Engineering College, Misammaguda(V), Kompally-500100, Telangana State, India.

⁵ Assistant Professor, Department of ECE, Narsimha Reddy Engineering College, Misammaguda(V), Kompally-500100, Telangana State, India.

Abstract

As a crucial strategy for low power design and quantum computing, reversible logic synthesis and testing is an intriguing field of study. A wide variety of fields may benefit from reversible calculations, including quantum computing, nanotechnology, bio-information, digital signal processing, and many more. To protect sensitive information from prying eyes, a cryptography system is essential for all of these uses. One of the main issues with well-cured cryptography algorithms is their high power and area requirements. To get around these issues, this paper suggests a Reversible Logic Gates Cryptography Design (RLGCD). Architectures for encryption and decryption are both designed using RLGCD. To encrypt and decrypt data, the key is generated using a Linear Feedback Shift Register. The Least Significant Bit (LSB) technique is used for data watermarking in order to further enhance data security. It assesses the RLGCD architecture's FPGA performance. The performance of RLGCD architecture is much better than that of other traditional systems.

Index Terms—Reversible Logic Gate Cryptography Design (RLGCD), Linear feedback shift register (LFSR), Field Programmable Gate Array (FPGA), Watermarking

INTRODUCTION

Protecting information by transforming it into an unintelligible format is known as cryptography. This ensures that the data remains secret. The first step, encryption, involves changing the data from plain text to cipher text. The second step, decryption, includes recovering the original data from cipher text. Heat dissipation is a major obstacle in very large scale integration (VLSI) design. Currently, ICs are becoming smaller and transistor counts are rising, and all of this is occurring in accordance with Moore's law [1]. Dissipation of heat, however, grows in tandem with increasing integration and scalability. Data loss results in heat dissipation in the range of $KT \ln(2)$, as shown by Landauer's study [2]. That is, where T is the temperature in Kelvin and K is the Boltzmann constant. The elimination of heat dissipation may be achieved by transforming typical irreversible systems into reversible ones, according to Bennett's work [3]. As there is no data loss during reversible computing, very little heat is disseminated. In other words, the entropy of the system does not diminish.

Since data transmissions may occur across untrusted media, making them vulnerable to hacking, cryptography is an essential component of the data and telecommunications infrastructure. Both low power consumption and good security are required in a cryptography system. For this, the most effective cryptography system implementation is one that makes use of reversible logic gates.

In this study, we introduce an RLGCD, or Reversible Logic Gate Cryptography Design. Aside from the many potential uses in fields as diverse as medicine, finance, and government, the primary argument in favor of incorporating reversible technology into cryptography is the significant improvement in energy efficiency it provides compared to more traditional methods. The LFSR algorithm is used to produce the cryptographic key [4]. When compared to other techniques, the RLGCD architecture's FPGA performance is superior. With hackers popping up all over the place, protecting personal information has never been more crucial. We apply a watermark to the input picture using the LSB approach to make it more secure. One way to ensure that data is genuine is by watermarking, which involves inserting a pattern into the original data. This pattern acts as concealed data that may be used to confirm the information's legitimacy. A watermark is a kind of embedded, almost invisible, data. Below is the outline for the remainder of the article. In part II, we summarize the relevant literature. Section III explains the proposed RLGCD architecture. Section IV presents the experimental findings. This section concludes with some last thoughts.

RELATED WORKS

Cryptography that is both lightweight and resistant to failure Use of lightweight block ciphers is essential in embedded systems that include sensitive nodes, such RFID tags and nano-sensors. Lightweight block ciphers with built-in error detection are suggested in [5]. This study makes use of XTEA (eXtended TEA), one of the world's quickest and most efficient block ciphers. It requires minimal memory and processing power, has a tiny code footprint, and employs basic arithmetic operations like addition, XOR, and shift. While the suggested techniques are reliable, the XTEA approach has a lower accuracy rate, which is a drawback. Designing DES with reversible logic gates for security A four-bit counter and a two-way shift register based on reversible logic gates make up the security portion design of the Data Encryption Standard (DES) employing RLG [6]. This work features minimal power consumption and strong data security since RLG is utilized to implement DES's security portion. Additionally, no performance assessments were conducted, and no particular RLG design is given. Analysis of security and improved dynamic block cipher We were able to dynamically adjust the S-box size and the number of registers needed based on the security requirements [7]. Based on the confusion replacement of S-box and ensuring that the internal structure of data blocks is disorder-free via four phases of matrix transformation, this study improves the safety of the cipher text. The diffusivity of the encrypted text was then determined by repeatedly shifting bytes in a certain direction using a column ambiguity function. Lastly, LFSR is used to produce dynamic. This means that the secret key's stochastic characteristic becomes better with each repetition. This method was able to scale well. However, with an odd-numbered dimension, achieving the S box becomes more challenging and time-consuming for encryption and decryption. D. Hardware designs for cryptographic block ciphers that are reliable The article delves into two block ciphers, HIGHT and LED, that may be used in authorized encryption techniques [8]. The former are ideal for embedded systems with less power requirements and little complexity due to their Feistel network topology. The latter is an AES-type, which is known for its efficiency. The task is very efficient and has a high mistake coverage rate. However, it can't tell the difference between temporary and permanent problems.

PROPOSED ALGORITHM

Random Logic Gates (RLGs) RLGs are circuits with a unique one-to-one mapping relationship and an equal number of inputs and outputs. So, there is no data loss during calculation since the input pattern can be recovered from the output pattern. Take RLG as an example; let 110 be the pattern it receives as input. The output will be 001 after the logic process is finished. A reversible operation has taken place if, after applying this to the input of 001, we get the output of 110. For every bit of data lost during operation, there will be an equal amount of heat energy dissipated when employing standard combinational logic circuits. This is due to the fact that once information is lost, it cannot be recovered in accordance with the second rule of thermodynamics. Therefore, a logical zero power dissipation may be achieved when the calculation is done in a reversible way. that is, the system's entropy does not diminish. The following are some limitations of RLG design: • RLGs do not permit fanout [9]. Minimizing quantum cost is of the utmost importance.

Reduce the amount of trash produced by optimizing the design. The bare minimum for a reversible logic circuit is one gate level. The Feynman gate, Fredkin gate, Toffoli gate, and SCL gates are the RLGs that were used to create this innovative cryptographic system. You can see them in Fig.1.

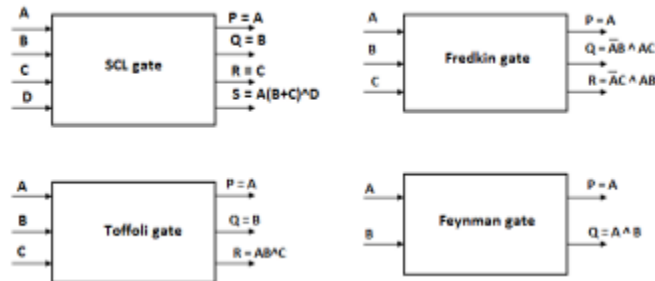


Fig. 1. Block diagram of RLGs.

Simple block diagram The block diagram of the whole cryptography process is shown in Fig. 2. Below is a description of the proposed RLGCD's operating concept. • The first step is to apply a watermark to the supplied picture by reading it into MATLAB. • Step 2: The input picture that has been watermarked is converted into a binary image following the LSB watermarking procedure. 3. In Step 3, MATLAB will be used to write the binary picture pixel values into a text file. • Step 4: A key is necessary to carry out the cryptographic operations. The LFSR is used to generate this key. 5. In Step 5, the text file generated from MATLAB is used as input to the Verilog code, which is responsible for performing cryptographic procedures like encryption and decryption. 6. After that, in order to verify the output, the encryption and decryption results are transferred into text files in Verilog. • Step 7: The text file contains both the encrypted and decrypted binary pixel values; from there, the pixels are reassembled in MATLAB. These pixel values are then used to construct the encrypted and decrypted pictures. At this point, in Step 8, both the input and decrypted images will be identical. • Step 9: The decrypted image is used to retrieve the watermark. The Verilog code is used to assess the FPGA's performances in Step 10. Subject: LSB watermarking One of the easiest ways to include a watermark is using the least significant bit (LSB). The bits of watermarking data are substituted for the original image's LSB bits in LSB watermarking. Because of this, the alterations that have been made are invisible to the human eye. An picture with dimensions of 128 by 128 pixels may so store a 128-bit

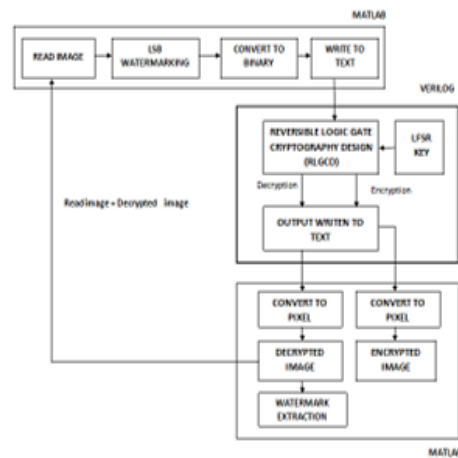


Fig. 2. Overall block diagram.

, in all, 16,384 bits of sensitive information. While lossy compression, cropping, and other noise modifications may withstand simple LSB watermarking, a more advanced attacker can simply retrieve the modified bits. Consequently, the original image's third and fourth LSB are used to insert the watermark in this work [10]. Secret data injection into these LSB places is unlikely to be anticipated. The safety of the system will be improved by this. After entering the watermark into MATLAB, the original 128x128 input picture is read and converted to a binary value. The data is then embedded into the third and fourth least significant bits (LSBs) of the picture, beginning with the first. To begin, a 5-pixel gap separates the first eight pixels' third and fourth least significant bits (LSBs), which include the binary watermark data length. Because 817 is the maximum length for a binary watermark, we will be asked to modify the data if it exceeds this limit. The watermark data is put into the third and fourth LSB with a five-pixel gap after the data length. Therefore, the input picture with the watermark is acquired. Following decryption, the watermark extraction method involves inserting the watermark in the opposite direction. Starting with the first pixel and jumping five pixels, the length of the secret data is derived from the third and fourth LSBs until it reaches the eighth pixel. The embedded data from the third and fourth LSBs is then obtained in the same manner. We revert the acquired binary data to its character counterpart in order to reveal the watermark we applied. You may use either a color or grayscale picture as the input. The blue part of a color picture is where the watermark is applied. This is due to the fact that it is not as perceptible to the human eye. Method of encryption You can see the encryption procedure in Fig. 3. Because of this, the values of the pixels are 8-bit binary words: $i[0]$, $i[1]$, $i[2]$, $i[3]$, $i[4]$, $i[5]$, $i[6]$, $i[7]$. One SCL gate is supplied by the first four MSB input bits, while the other SCL gate is fed by the first four LSB input bits.

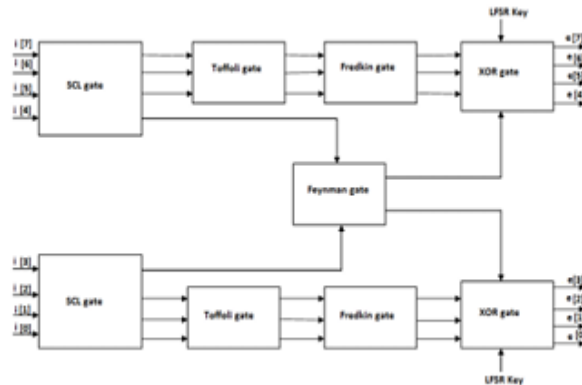


Fig. 3. Encryption block.

image bitstream. The SCL gate operation requires four of these inputs to finish, and the output is four bits. Using the Toffoli gate operation, the first three left-side-bit (LSB) outputs of the SCL gate below provide three distinct bits of output. Similarly, the Toffoli gate receives its input from the first three MSB values of the SCL gate and generates three bits of output. The above and below SCL gates each have an output bit that may execute a Feynman gate operation. Its outputs execute the Fredkin gate since it follows both Toffoli gates. By connecting the outputs of the Fredkin and Feynman gates to the XOR gates, an XOR operation using the LFSR key may be executed. Then, the encrypted binary image pixel values $e[0]$, $e[1]$, $e[2]$, $e[3]$, $e[4]$, $e[5]$, $e[6]$, and $e[7]$ are provided by the XOR gate output.

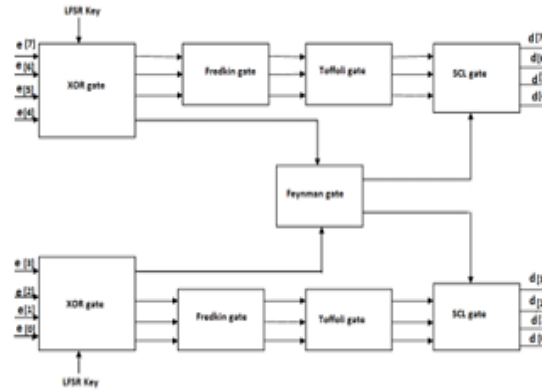


Fig. 4. Decryption block.

Fig.4 shows the decryption procedure. Decryption is essentially the same as encrypting, but in reverse. As a result, the decryption process block receives its input from the encryption process. The LFSR-generated key is first used in an XOR operation with the encrypted pixel bits. At the SCL gate output, the encrypted data is acquired after four reversible gate operations, one after the other. Values of the eight-bit pixels in the decrypted output are $d[0]$, $d[1]$, $d[2]$, $d[3]$, $d[4]$, $d[5]$, $d[6]$, and $d[7]$. The binary output values, both encrypted and decrypted, are saved to a text file. From the output text file, MATLAB generates both the encrypted and decrypted images. Section F. Sequential Feedback Shift Register You may create random key patterns using a linear feedback shift register, which is also called a random number generator (LFSR). It is an XNOR gate and four flip-flops that make up this four-bit LFSR. The first step is to randomly assign a value to each flip-flop. The seed value is this arbitrary first-bit word. When clocked, the LFSR uses a feedback loop and bits to change the seed value to create a random test pattern. The generation of random encryption keys is a common use of LFSRs [11]. Hence, it may be used in stream ciphers, and the LFSR is well-suited for several additional low- and high-speed applications. How many random sequences are produced is proportional to the degree of its feedback polynomial. Given that LFSR is really just a feedback-based simple counter, its highest value that can be achieved by designing it with its maximum length feedback polynomial is $2n-1$ [12]. A key's size is a factor in

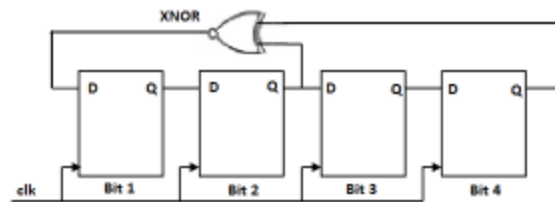


Fig. 5. Linear Feedback Shift Register.

create a cryptography system that is both space and power limited. Using a higher key will cause the network load to increase. This study uses an LFSR architecture to generate random keys in an effort to circumvent this issue. Figure 5 shows the LFSR block diagram. Even with lower integrity, cryptographic processes may employ LFSR to achieve secret message delivery. The input is a picture, and the output is encrypted with the LFSR key. The watermarked input picture pixels are handled independently in Verilog, much like individual blocks. It uses a different key for decrypting each pixel value and a different key for each pixel value. The receiver is then able to decode all of the data. Consequently, the LFSR method may be used to create a cryptography system that is both more secure and more effective.

EXPERIMENT AND RESULT

This study utilizes Xilinx ISE 14.7 to simulate an RLG-based cryptography system. In MATLAB 2018, the input picture is read and watermarking is applied.



Fig. 6. Original input image.

In Fig.6, we can see the 128x128 input pepper picture. A binary representation of an image's pixel values is generated in MATLAB. The data OUTPUT is transformed to a binary value and then used as a watermark, as seen in Figure 7. See Figure 8 for the original picture's binary value and Figure 9 for the input image with the watermark. Figure 10 shows the input picture with the watermark.

```
'1001111'
'1010101'
'1010100'
'1010000'
'1010101'
'1010100'
```

Fig. 7. Binary value of watermark

01101111	01100011
01101100	01101100
01101010	01101010
01100011	01100011
10011111	10011111
10110000	10110000
10101101	10101101
10110011	10110011
10110110	10110110
10111100	10111100
10111010	10111010
10110111	10110111
10111011	10111011
10110011	10110011
01110110	01110110
01110100	01110100

Fig. 8. Binary value of original im age, Fig. 9. Binary value of watermarked input image.



Fig. 10. Watermarked input image.

The verilog-designed RLGCD receives its input from a text file containing the binary watermarked picture values. Fig.11 shows the time diagram of RLGCD, which encompasses the encryption and decryption processes. Xilinx ISE displays the

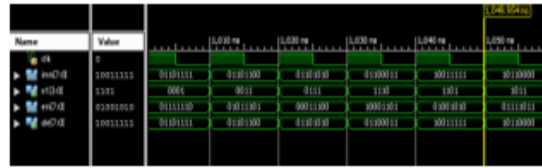


Fig. 11. Timing diagram of cryptography process using RLGCD.

The "readmemb" function is used to read the file. With an input picture size of 128 by 128 pixels, the resultant binary value depth contains 16,384 words. The "inn" is the input representation. A key called "x1" is produced using the LFSR algorithm. The final outputs are denoted as "de" after decryption and "en" after encryption, respectively. You can see that the input and decryption pixel values are identical in the timing diagram. In order to display the encrypted and decrypted images, MATLAB reads the "en" and "de" variables. The encrypted and decrypted images are shown in Figures 12 and 13, respectively. It is evident that the decrypted picture is identical to the input image.

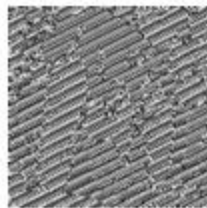


Fig. 12. Encrypted image Fig. 13. Decrypted image

You may use this RLGCD with color images as well. The input color picture, encrypted image, and decrypted image are shown in Figures 14, 15, and 16, respectively, after the watermarking process.

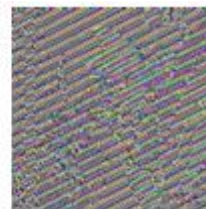


Fig. 14. Watermarked input image, Fig. 15. Encrypted image

As shown in Figure 18, the RLGCD outperforms competing systems when it comes to device usage, according to data collected from the Xilinx Spartan3E XC3S500E. The table displays a comparison of the performance of several existing systems with the RLGCD utilizing the Spartan3E device. When compared to previous systems, I and which demonstrate a significant improvement in device usage. A tool for estimating power usage is the Xilinx Power Estimated. The projected power of the RLGCD is 85 mW.

TABLE 1 FPGA PERFORMANCE OF VARIOUS DESIGNS

Target FPGA	Circuit	LUT	Flipflop	Slice	Frequency (MHz)
Virtex 7	Isogenies- MC [13]	185,871	218,012	77,425	158.5
Virtex 7	Scalable isogeny[14]	18,820	24,908	4791	202.1
Virtex 7	AES-NPL[15]	19,547	53,478	4089	495.32
Spartan 3E	RLGCD	37	40	42	175.047

Device utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	40	9,312	3%	
Number of 4-input LUTs	37	9,312	3%	
Number of occupied Slices	42	4,858	3%	
Number of Slices containing only related logic	42	42	100%	
Number of Slices containing unrelated logic	0	42	0%	
Total number of 4-input LUTs	39	9,312	3%	
Number used as logic	37			
Number used as a route-thru	32			
Number of bonded I/Os	33	232	14%	
Number of BUFGs	8	23	40%	
Number of BUFGMUXs	1	29	4%	
Average Fanout of Non-Clock Nets	4.31			

Fig. 18. FPPGA result of RLGCD for Spartan 3E device

Both the RTL perspective of the main module RLGCD (Fig.19) and the RTL schematic of the encryption and decryption blocks (Fig.20 and Fig.21) are shown in the corresponding physical images. You may check the design's functionality using these RTL diagrams.

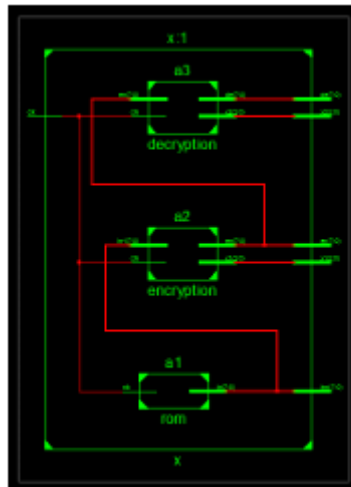


Fig. 19. RTL of RLGCD

CONCLUSION

This paper details an LFSR key-watermarking reversible logic gate cryptography design. Among the reversible gates used in this novel cryptographic system design are the Feynman, Fredkin, Toffoli, and SCL gates. This study is among the finest of its kind since cryptography systems need both low power consumption and great security. You may use MATLAB to read the input images, add watermarks, and convert them to binary format. Then, you can write the binary data to a text file. This data

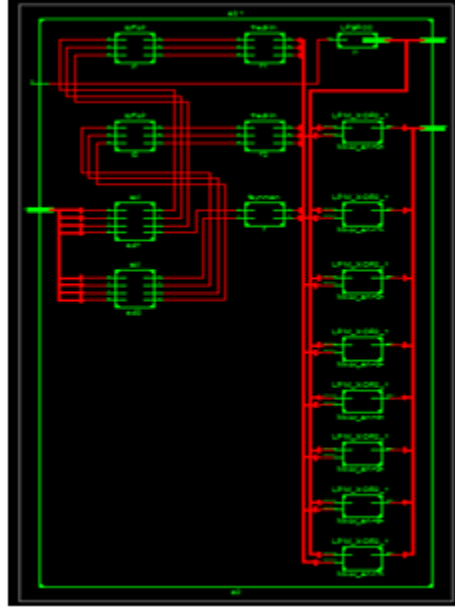


Fig. 20. RTL schematic of encryption block

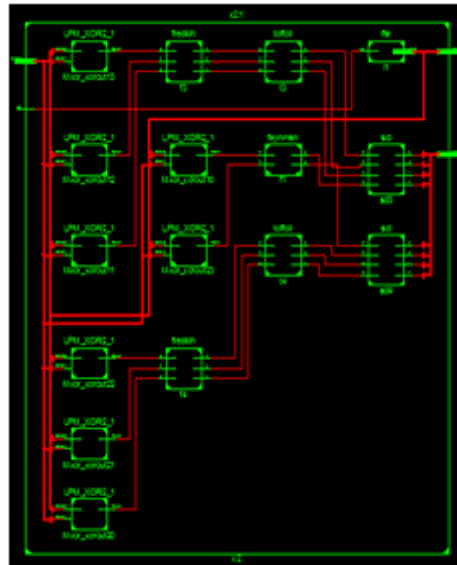


Fig. 21. RTL schematic of decryption block

The Xilinx ISE interprets the pixel values. The Xilinx program contains the RLGCD architecture, which includes the LFSR, encryption block, and decryption block. Images in either grayscale or color may be accommodated by this design. To make the data more secure, the LSB approach is used for watermarking. When compared with competing systems, the Xilinx performance result for the Spartan3E XC3S500E device is much superior. A prerequisite for the new area of quantum computing is the use of reversible logic gates. Therefore, the area of quantum logics will advance thanks to all of the works that use reversible logic gates. The future effective deployment of RLGCD on ASIC is within reach, since it has been successfully developed using verilog code.

REFERENCES

- [1]. Gordon E. Moore, "Cramming more components onto integrated circuits," Electronics, pp.114-117, April 1965.

- [2]. Rolf Landauer, Irreversible and heat generation in the computing process, IBM Research and Development, vol.5, pp.183–191, July 1961.
- [3]. C.H. Bennett, “Logical reversibility of computation” IBM Research and Development, vol.17, pp.525–532, 1973.
- [4]. Saranya Karunamurthi, Vineyakumar Krishnasamy Natarajan, “ VLSI implementation of reversible logic gates cryptography with LFSR key,” *Microprocessors and Microsystems*, Elsevier, vol. 69, pp.68–78, September 2019.
- [5]. Mehran Mozaffari Kermani, Kaj Reza Azarderakhsh, Siavash Bavat Sarmadi, “Fault resilient lightweight cryptography block cipher for secure embedded systems,” in *IEEE Embedded System Letters*, vol. 6, no. 4, pp.89–92, Dec. 2014.
- [6]. Shikha Kuchhal , Rakesh Verma, “Security design of DES using reversible logic,” *Int. J. Comput. Sci. Netw. Security*, vol. 15, no. 9, pp. 81–84, September 2015.
- [7]. Z. H. A. O. Guosheng, W. A. N. G. Jain, “Security analysis and enhanced design of a dynamic block cipher,” *China Commun.*, vol. 13, pp. 15–160, January 2016.
- [8]. Srivatsam Subramanian, Mehran Mozaffari Kermani, Reza Azarderakhsh, Mehrdad Nojoumaian, “Reliable hardware architectures for cryptographic block ciphers LED and HIGHT,” in *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, vol. 36, no.10, pp. 1750-1758, Oct. 2017.
- [9]. Raghava Garipelly, P. Madhu Kiran, A. Santhosh Kumar, “A review on reversible logic gates and their implementation,” in *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no. 3, March 2013.
- [10]. Abduallah Bamatraf, Rosziati Ibrahim, Mohd. Najib. B, Mohd. Salleh, “Digital watermarking algorithm using LSB,” in *2010 International Conference on Computer Applications and Industrial Electronics*, Kuala Lumpur, pp. 155-159, 2010.
- [11]. Meenal Dadhe, Prof. Anup. R. Nage, “Design of high speed VLSI architecture for LFSR with maximum length feedback polynomial,” in *International Journal for Scientific Research & Development*, vol .3, no. 5, 2015.
- [12]. Y. G. Praveen Kumar, B. S. Kriyappa, M. Z. Kurian, “Implementation of power efficient 8-bit reversible linear feedback shift register for BIST,” in *2017 International Conference on Inventive Systems and Control*, Coimbatore, 2017.
- [13]. B. Koziel, R. Azarderakhsh, M. Mozaffari Kermani, D. Jao, “Post quantum cryptography on FPGA based on isogenies on elliptical curve,” in *IEEE Trans.Circuits Syst.I*, vol. 64, no. 1, pp. 86–99, Jan. 2017.
- [14]. B. Koziel, R. Azarderakhsh, M. Mozaffari Kermani, “A high performance and scalable hardware architecture for isogeny based cryptography,” in *IEEE Trans.Comput.*, vol. 67, no. 11, pp. 1594–1609, Nov. 2018.
- [15]. H. Zodpe, A. Sapkal, “An efficient AES implementation using FPGA with enhanced security features,” in *J.King Saud Univ.Eng.Sci.*, 2018, in press