

VLSI Implementaion of Image Encryption and Decryption Using Reversible Logic Gates

Ms.V.Vijayalakshmi¹, R.Likitha², S.Babavali³, G. Saketh⁴, M. Chandini⁵

¹Assistant Professor, Dept.of ECE, AITS.Tirupati

^{2,3,4,5}Student, Dept.of ECE, AITS.Tirupati

Abstract

Reversible logic synthesis and testing is a fascinating research area as it is an important approach for low power design and quantum computing. Reversible computations have different applications such as quantum computing, nanotechnology, digital signal processing, bio-information etc. All these applications require a cryptography system to restrict the unauthorized access and thus maintain the confidentiality of data. High area and power requirements are some of the major problems of well secured cryptography algorithms. In this work, a Reversible Logic Gates Cryptography Design (RLGCD) is proposed to overcome these problems. RLGCD is used to design both encryption and decryption architectures. Linear Feedback Shift Register is used to generate the key for encryption and decryption processes. To further improve the security of data watermarking is done using Least Significant Bit (LSB) method. The FPGA performance of RLGCD architecture is evaluated. There is a great improvement in the performance of RLGCD architecture when compared to other conventional systems.

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

I. INTRODUCTION

Cryptography is the process of protecting the information by converting it in to unreadable format and thus maintains the confidentiality of the data. This process involves the conversion of plain text into cipher text by the process called encryption and the process by which the original data that is the plain text is recovered back called decryption.

One of the major challenge in VLSI design is the heat dissipation. Now reducing the size of ICs and increasing the number of transistors are happening day by day and up to now all these obeys Moores law [1]. But with higher integration and scaling the amount of heat that is dissipated also increases. Landauers work [2] showed that for each bit of data that is lost there will be a heat dissipation in the range of $KT\ln(2)$. Where, K is the Boltzmann constant and T is the temperature in Kelvin scale. The work done by Bennett presented that this heat dissipation can be eliminated if the traditional irreversible systems are converted in to reversible systems [3]. Reversible computation is the operation in which there is no loss of information and thus scatters only a small amount of heat. That is, there is no decrease in the entropy of the system. In data and telecommunications, cryptography is one of the most necessary part since the communication even take place over untrusted mediums where the data can be easily hacked out. A cryptography system not only demands high security but also low power consumption. The cryptography system implementation using reversible logic gates offers the best solution for this.

A Reversible Logic Gate Cryptography Design (RLGCD) is presented in this paper. The biggest motivation of including reversible technologies in to cryptography includes, it gives energy efficiency much better than other conventional systems and such a cryptography system is useful for different applications such as medical field, banking, government organization etc. The key for cryptography is generated by using LFSR [4]. The FPGA performance of the RLGCD architecture is better as compared to existing methods.

II. RELATED WORKS

Authorized licensed use limited to: Rutgers University. Downloaded on May 17, 2021 at 03:50:36

UTC from IEEE Xplore. Restrictions apply.

A. Fault resilient light weight cryptography

Embedded systems having sensitive nodes such as RFID tags and nano-sensors necessitate the use of lightweight block ciphers. Error detection schemes for lightweight block ciphers are proposed in [5]. One of the fastest and most efficient block cipher in existence, XTEA (eXtended TEA) is used in this work. It uses simple addition, XOR, and shift functions, and has a very small code size, less memory requirement and less computational power. These proposed methods suitable for providing reliability but less accuracy in error rate is one of the demerit while using XTEA method.

B. Security design of DES using reversible logic gates

Security part design of the DES (Data Encryption Standard) using RLG [6] comprises of a reversible logic gate based two way shift register and four bit counter. Since RLG is used to implement the security part of DES, this work has good data security and low power consumption. But a specific RLG design is not provided and performance evaluations were not carried out.

C. Security analysis and enhanced dynamic block cipher

The S-box dimension and number of registers required can be dynamically varied with respect to the security requirements that we required [7]. In this work the safety of the cipher text was improved based on the confusion substitution of S-box and so that the internal structure of data blocks disorder by four steps of matrix transformation. Then by cyclic displacement of byte using column ambiguity function, the diffusivity of cipher text was obtained. Finally LFSR is used to generate dynamic. Thus the stochastic characteristic of secret key is improved in each round of iteration. This technique achieved high scalability. But it is difficult to achieve the S box when the dimension selected is an odd number and requires more time for encryption and decryption.

D. Reliable hardware architectures for cryptographic block ciphers

In this work, two block ciphers such as HIGHT and LED which can be employed in authenticated encryption algorithms are discussed [8]. The former have a Feistel network structure and it is good for low power and low complexity embedded applications. The latter is of an efficient Advanced Encryption Standard (AES) type. This work has high error coverage and high efficiency. But it is not able to detect the permanent and transient faults.

III. PROPOSED ALGORITHM

A. Reversible Logic Gates (RLGs)

RLGs are the circuits that having equal number of inputs and outputs with a unique one to one mapping relationship. Thus it is possible to recover the input pattern from the output pattern, so that there is no information loss during computation. For example let 110 is the pattern which is given as input to RLG. Then after completing the logic operation it produce 001 as output. If we apply this 001 as input and obtained 110 as output then it depicts the occurrence of a reversible operation. While using traditional combinational logic circuits, for every bit of data that is lost during operation there will be an equivalent heat energy dissipation. The reason behind this is according to the second law of thermodynamics there is no way to reproduce the information once lost. So when the computation is performed in a reversible manner then it is possible to achieve a logical zero power dissipation. i.e., there is no decrease in the entropy of the system. Constraints for designing RLGs include [9]

- RLGs do not allow fanout.
- Quantum cost should be minimum as possible.
- Optimize the design to make garbage outputs minimum.
- A reversible logic circuits should have least gate level. The RLG that are used to design this new cryptography system includes Feynman gate, Fredkin gate, Toffoli gate and SCL gates and are shown in Fig.1

B. Block diagram

Fig.2 presents the block diagram of the overall cryptography process. The working principle of the proposed RLGCD is described below.

- Step 1: MATLAB is used to read the input image and on this image watermarking is performed.
-

- Step 2: The LSB watermarking is used and after watermarking process the watermarked input image is converted into binary image.
- Step 3: The binary image pixel values will be written into a text file with in the MATLAB.
- Step 4: Inorder to perform the cryptography processes a key is required. This key is created using the LFSR.
- Step 5: The input to Verilog code is the text file output from MATLAB and in Verilog the cryptography processes such as encryption and decryption are performed.
- Step 6: Then, both the encryption output and decryption output are copied in to text files in Verilog for output verification.
- Step 7: In MATLAB the pixels are reconstructed from the encrypted binary pixel values and decrypted binary pixel values in the text file. The encrypted and decrypted images are then generated from these pixel values.

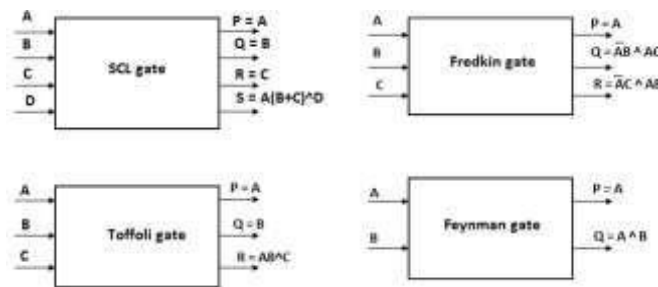


Fig. 1. Block diagram of RLGs.

- Step 8: Then, input image and decrypted image will be the same.
- Step 9: The watermark is extracted back from the decrypted image.
- Step 10: FPGA performances are evaluated using the Verilog code

c. LSB watermarking

The least significant bit (LSB) is one of the simplest watermark embedding technique. In LSB watermarking the LSB of the original image pixel bits are changed by the bits of watermarking data. The changes thus created cannot be find out by human visibility system. Since an LSB can hold 1 bit of data, an image of size 128x128 can thus able to store a total of 16384 bits of secret data. Simple LSB watermarking can survive transformations like lossy compression, cropping or other additions of noise but a more sophisticated attacker caneasily extract the changed bits.

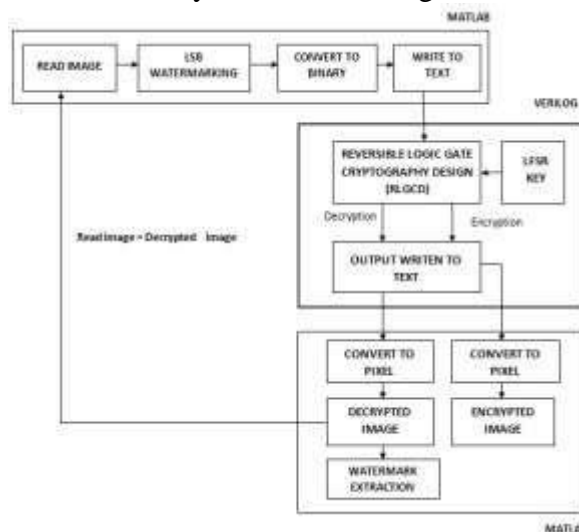


Fig. 2. Overall block diagram.

So in this work watermark embedding is performed into the third and forth LSB of the original image [10]. There is less probability that anyone expects insertion of secret data in these LSB positions. This will helps to enhance the security of the system. First an the original input image of size 128x128 is read in MATLAB and transfer the watermark into binary value after typing it. The data is then embed into the third and then forth LSBs of the image starting from the first. First the length of the binary watermark data is embedded in the third and forth LSBs of the first eight pixels with a gap of 5 pixels. So the maximum lengthof binary watermark is 817 and thus it will ask us to rewrite the data if the length is more than 817. After the length of the data, the watermark data is written in to the third and forth LSB with a gap of five pixels. Thus the watermarked input image is obtained. In watermark extraction process after decryption, the reverse operation of embedding is performed. First of all the length of secret data is extracted from the third and forth LSBs starting from the first pixel and jump by five until it get it from the eight pixels. Then in the same way we get the embedded data from the third and forth LSBs. The obtained binary data is then converted back to the character which will give the watermark that we applied. The input image can be gray scale image or a color image. For color image watermarking is performed on the blue component of the image. This is because it is less sensitive to human visual system.

D. Encryption process

The encryption process is shown in Fig.3. The pixel values are thus 8 bit binary word: $i[0], i[1], i[2], i[3], i[4], i[5], i[6], i[7]$. The first four LSB input bits is applied to the below SCL gate and the above SCL gate is fed by the first four MSB

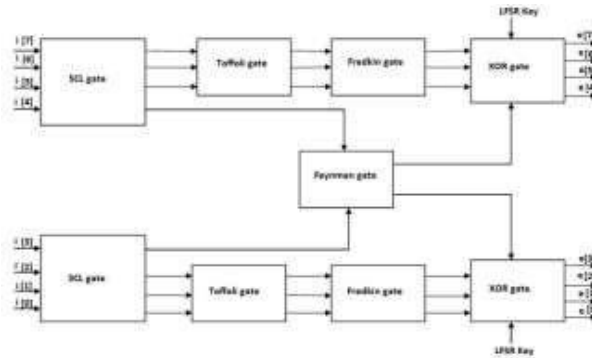


Fig. 3. Encryption block.

input pixel bits. Four of these inputs complete the SCL gate operation and thus produce four result bits. The first three LSB outputs from the below SCL gate perform Toffoli gate operation and provides three different output bits. Similarly the first three MSB value outputs of SCL gate feed Toffoli gate and provides three output bits. One of the output bit from the above and below SCL gates perform Feynman gate operation. Both Toffoli gates are followed by Fredkin gate and thus its outputs perform Fredkin gate. The Fredkin gate outputs and the Feynman gate outputs are connected to the XOR gates and thus perform XOR operation with LFSR key. Then, the XOR gate output provides the encrypted binary image pixel value $e[0], e[1], e[2], e[3], e[4], e[5], e[6], e[7]$.

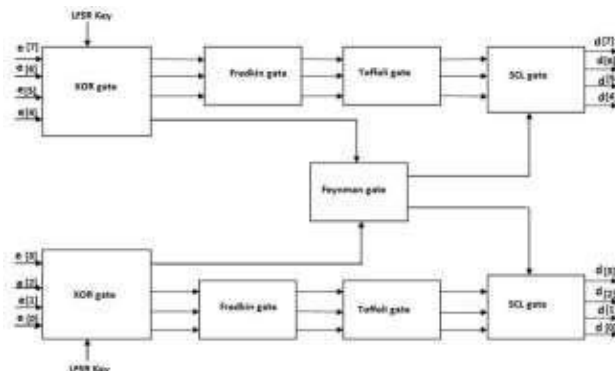




Fig. 4. Decryption block.

is used. In order to get rid of this problem, in this work random keys are generated by using a LFSR design.

Block diagram of the LFSR is shown in Fig.5. The use of LFSR helps cryptography process to achieve confidential message transmission even with a lesser integrity. An image is taken as input, on which the cryptography is performed by using the LFSR key. In Verilog the watermarked input image pixel are processed one by one like separate blocks. For each pixel value a separate key is used and for decrypting that the same key is used. In this way the entire data is decrypted on receiver part. Thus, a more secure and effective cryptography system is achievable by using the LFSR technique.

IV. EXPERIMENT AND RESULT

In this work, RLG based cryptography system is simulated in Xilinx ISE 14.7. The read operation of the input image and the process of decryption is shown in Fig.4. The decryption process is just the reverse operation of the encryption. Thus encryption process output is fed as input to decryption process block. First, the encrypted pixel bits perform XOR operation with the key generated by the LFSR. After performing the four reversible gate operation one followed by the next the decrypted output are obtained at the SCL gate output. The decrypted output eight bit pixel values are $d[0]$, $d[1]$, $d[2]$, $d[3]$, $d[4]$, $d[5]$, $d[6]$, $d[7]$. The encrypted as well as the decrypted binary output values are written into a text file. In MATLAB encrypted image and decrypted image are generated from the output text file.

E. Linear Feedback Shift Register

Linear Feedback Shift Register (LFSR) is one also known as random number generator and thus can be used for producing random key patterns. This is a 4 bit LFSR and it consists of 4 flipflops and an XNOR gate. First of all the flipflops are loaded with a random value. This random first bit word is called the seed value. The LFSR then generates random test pattern when it is clocked, by shifting the seed value bits and by using a feedback. LFSR are widely used for generating random encryption key [11]. Therefore, we can use it in stream ciphers and the LFSR is also suitable for other high and low speed applications. The total number of random sequence generated will depend on its feedback polynomial. Since LFSR is basically a simple counter with feedback, it counts up to a maximum value of $2^n - 1$ when designed using its maximum length feedback polynomial [12]. The size of key is one of

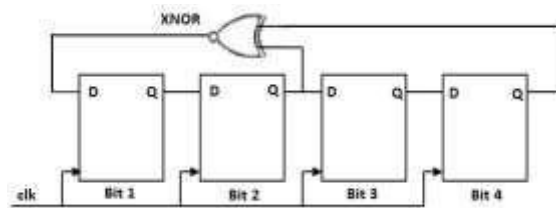


Fig. 5. Linear Feedback Shift Register.

the major factor in a power and area constrained cryptography system. The network load will become high when a higher key watermarking is performed in MATLAB 2018.



Fig. 6. Original input image.

The input pepper image is shown in Fig.6 which is a 128x128 image. In MATLAB, image pixel values are converted to binary values. The data OUTPUT is used as the watermark and is also converted to

binary value as shown in Fig.7. Fig.8 and Fig.9 shows the binary value of the original image and the watermarked input image respectively. The watermarked input image is shown in Fig.10.

```
'10011111'  
'1010101'  
'1010100'  
'1010000'  
'1010101'  
'1010100'
```

Fig. 7. Binary value of watermark.

The binary watermarked image values are written to a text file which is the input for the RLGCD designed in verilog. The timing diagram of RLGCD which include both encryption and decryption process is shown in Fig.11. In Xilinx ISE the text



Fig. 8. Timing diagram of cryptography process using RLGCD.

file is read using the function "readmemb". Since a 128x128 image is taken as input, the resulting binary value depth have a 16,384 words. The input is represented as "inn", The key generated by LFSR is mentioned as "x1". After completing both encryption and decryption operations, the final outputs are represented as "en" and "de" respectively. From the timing diagram it is clear that the decrypted pixel value is as same as the input pixel value. The "en" and "de" variables are read in MATLAB to present the encrypted image and the decrypted image. Fig.12 and Fig.13 shows the encrypted and decrypted image respectively and it shows that the decrypted image is as same as the input image.



Fig. 9. Encrypted image Fig. 13. Decrypted image

This RLGCD is suitable for color image also. A sample color image input after performing watermarking, encrypted image and decrypted images are shown in Fig.14, Fig.15 and Fig.16.



Fig. 10. Watermarked input image Fig. 15. Encrypted image

The watermark data is then extracted from the decrypted image in MATLAB and is shown in Fig.

'O'
'U'
'T'
'E'
'U'
'T'

Fig. 12. Extracted watermark

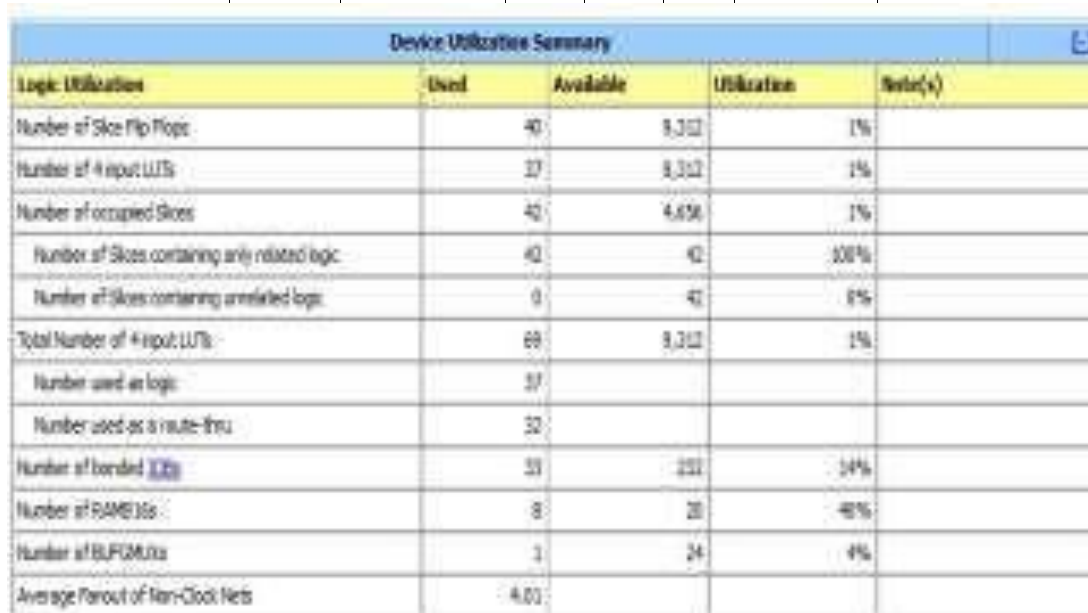
Fig.18. presents the device utilization result of RLGCD taken from Xilinx for Spartan3E XC3S500E device and the result shows that the RLGCD have a better performance in comparison with other existing systems.

The performance comparison of different existing systems and the RLGCD using Spartan3E device is presented in the Table.I and which shows that there is a good improvement in the device utilization as compared with other systems. The Xilinx Power Estimator is used to estimate the power consumption. The RLGCD has an estimated power of 85mW.

TABLE I

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	Notes(s)
Number of Slice Flip Flops	40	3,312	1%	
Number of 4 input LUTs	37	3,312	1%	
Number of occupied Slices	42	4,656	1%	
Number of Slices containing any related logic	42	42	100%	
Number of Slices containing unrelated logic	0	42	0%	
Total Number of 4 input LUTs	69	3,312	1%	
Number used as logic	37			
Number used as a route thru	32			
Number of bonded I/Os	59	232	24%	
Number of RAMB16s	8	20	40%	
Number of BRAM16s	1	24	4%	
Average Fanout of Fan-Clk Nets	4.03			

Fig. 13. FPPGA result of RLGCD for Spartan 3E device

Fig.19 shows the RTL view of the main module RLGCD and the RTL schematic of encryption and decryption blocks are shown in Fig.20 and Fig.21. These RTL schematics help to verify that the design is functionally correct.

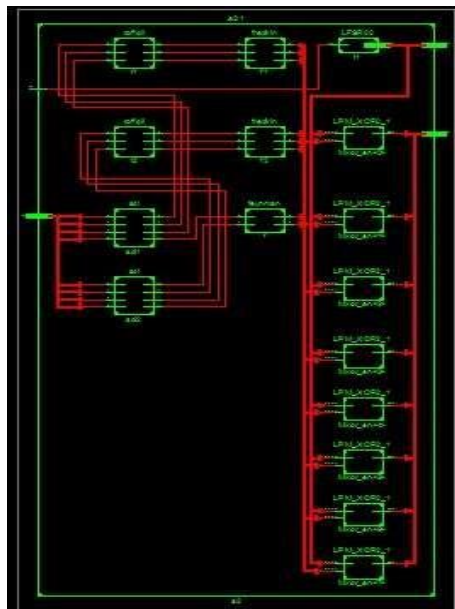
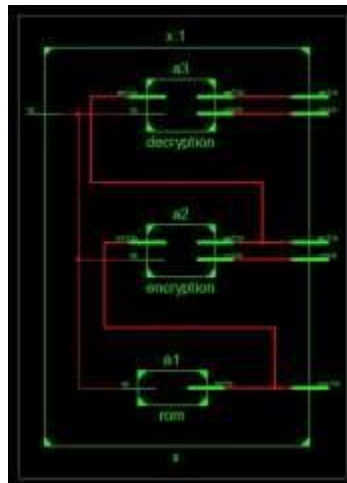


Fig. 14. RTL schematic of encryption block

V. CONCLUSION

This work presents a Reversible Logic Gate Cryptography Design using LFSR key with watermarking. The reversible gates like Feynman, Fredkin, Toffoli and SCL gates are used in this new cryptography system design. Since a cryptography system demands not only high security but low power consumption this work is one of the best among existing systems. The input image read operation, watermarking and conversion to binary format are carried out in MATLAB and those binary values are written into a text file. This input

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, “Postquantum 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.