ZKG INTERNATIONAL ISSN: 2366-1313

Efficient Image Encryption Architecture Using Reversible Logic

[1] Kasarla Bhanu, [2] Dr.V. Manohar,

[1] M Tech Student, Dept. Of ECE, Vaagdevi Engineering College

Bolikunta, Warangal

[2] Assistant Professor, Dept. Of ECE, Vaagdevi Engineering College

Bolikunta, Warangal

Abstract: *Developing and testing reversible logic is a fascinating new area of research that is crucial to low power architecture and quantum computing. Digital signal processing, bio-information, quantum computing, nanotechnology, and countless more may all stand to gain from reversible computations. To further ensure the safety of users' private data, encryption is an integral part of each of these applications. Efficient encryption techniques raise two important issues: the amount of power used and the amount of storage space needed. This study offered a new approach to cryptography that uses reversible logic gates: RLGCD. The RLGCD is used to build architectural frameworks for encryption and decryption. A Linear Feedback Shift Register is used to produce the keys for both encryption and decryption. One Least Significant Bit (LSB) method for data security is data watermarking. For FPGA implementations, we evaluate the RLGCD architecture. Compared to older, less efficient systems, the RLGCD architecture provides much better performance.*

Within the index, you will find a variety of terms such as watermarking, RLGCD, LFSR, FPGA, and linear feedback shift register.

I. INTRODUCTION

The method of cryptography entails rendering information incomprehensible, guaranteeing data privacy. Data is transformed from plain text to encrypted text during the first phase of encryption. Deciphering cipher text entails restoring the original data as the second step.

Very large scale integration (VLSI) designs face the significant challenge of heat dissipation. For the time being, everything is proceeding according to

ISSN: 2366-1313

Moore's law, which specifies that the number of transistors in an integrated circuit will grow as its size reduces [1]. The amount of heat lost, however, also increases as integration and size do. For every bit of data lost, heat dissipation happens in the range of KTln (2), according to Landaulet's study [2]. When discussing this topic, the Boltzmann constant (K) and the temperature (T) in Kelvin are used interchangeably. According to Bennett's research, heat dissipation may be eliminated if reversible systems are used instead of conventional irreversible ones [3]. Reversible computing does not cause data loss; hence it dissipates relatively little heat. For this system, entropy is not decreasing.

Cryptography is a crucial part of the infrastructure for data and communications since data transfers might happen across unfrosted media, leaving them open to hackers. Any cryptography system worth its salt will have minimal power consumption and top-notch security. Here, the best approach to creating a reversible logic gate cryptography system would be perfect.

The RLGCD, a scheme for reversible logic gate encryption, is presented in this paper. The tremendous gain in energy efficiency when compared to more conventional systems is the main justification for using reversible technology in cryptography. Furthermore, this kind of encryption might have several uses in domains as varied as healthcare, banking, and government, to name a few. It is possible to generate a cryptographic key using LFSR [4]. The RLGCD design has better FPGA performance than competing methods.

Because hackers may be located anywhere in the globe, protecting personal information has never been more important. To make it more secure, the LSB method is used to watermark the input image. Watermarking is a method of authenticating data that entails incorporating a pattern into the initial data. This pattern may be used to verify the information's validity by revealing hidden data. A watermark is a kind of embedded data that is almost undetectable. The rest of the paper is organized like this. A synopsis of the pertinent literature is given in Section II. The suggested RLGCD design is detailed in Section III. In Section IV, we provide the results of the experiments. We provide some concluding remarks in section V.

II RELATED WORKS

A Safe and Easy-to-Use Cryptography System

If you're building an embedded system with sensitive nodes like RFID tags or nano-sensors, you must use lightweight

ISSN: 2366-1313

block ciphers. As mentioned in [5], there need to be lightweight block ciphers that come equipped with built-in error detection. The XTEA (extended TEA) block cipher, widely recognized as the world's fastest and most efficient encryption, is used in this investigation. It is particularly efficient because to its tiny code footprint, minimal memory needs, and dependence on fundamental arithmetic operations such as addition, XOR, and shift. It makes no difference how trustworthy the other techniques are; the XTEA approach will still provide worse results.

A key component of the DES security design is reversible logic gates (B).

The Data Encryption Standard (DES) employs a four-bit counter and a two-way shift register in its reversible logic gate–based security architecture [6]. Using RLG to build DES's security aspect allows for good data security with minimum power consumption in this work. It is frustrating that there is a lack of information on performance evaluations and an exact RLG design.

Section C: Evaluating safety and improving dynamic block cipher

Based on the security requirements, we may dynamically modify the S-box size and the amount of registers required [7]. To make the cipher text more secure, this work uses S-box confusion replacement and four stages of matrix transformation to guarantee that the data blocks' internal structure is disorder-free. Applying the column ambiguity function to cyclic byte displacement allowed us to compute the diffusivity of the encrypted data. In the end, we'll use LFSR to create dynamic. The stochastic feature of the secret key improves with each iteration. In this way, it was able to attain remarkable scalability. The S box gets more elusive when the selected dimension is an odd integer because the time needed for encryption and decryption increases.

(D) Reliable hardware models for encrypted block ciphers

Authenticated encryption techniques include the option of using block ciphers, and this work [8] covers both HIGHT and LED. Because they use a Feistel network design, the former are perfect for basic, low-power embedded computers. The latter is a safe and efficient security method, much like AES. The work is really efficient and accurate. No matter how long or short the time horizon is, it can't tell the difference between issues.

III PROPOSED ALGORITHM

The RLGs, or Reversible Logic Gates, Part

ISSN: 2366-1313

A

Real-life graphs, or RLGs, are circuits with a fixed number of inputs and outputs and an obvious one-to-one mapping. It is feasible to recover the input pattern using the output pattern in the event that data loss happens during processing. For the sake of argument, let's pretend that RLG receives a pattern 110. The outcome will be 001 after the logic procedure is finished. This process is reversible, as shown by input 001 and output 110. Traditional combinational logic circuits lose data at a rate that is directly proportionate to the amount of heat energy they emit. The second law of thermodynamics states that once data is destroyed, it cannot be recovered. The reason for doing it this way is this. The only method to guarantee a logical zero power dissipation is to calculate in a reverse technique. This indicates that the system's entropy has remained unchanged. A few limitations are present in this RLG design: [9] even though RLGs don't provide fan-out, the quantum cost shouldn't be too high.

When designing reversible logic circuits, it's best to minimize the number of gates and minimize trash outputs. The RLG shown in Figure 1 provided the basis for the construction of this groundbreaking encryption system. This class includes the Feynman, Fredkin, Toffol, and SCL gates, among others.

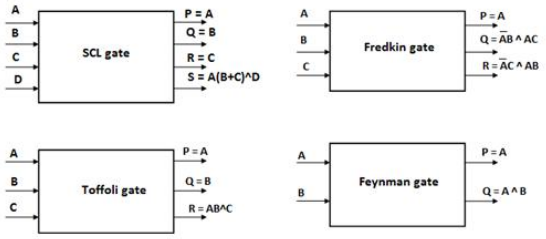


Fig. 1. Block diagram of RLGs.

Make block diagrams simple

Figure 2 shows the whole cryptography process using a block diagram. Described below is the RLGCD's planned operation.

You need to get the picture into MATLAB first before you can add a watermark on it.

The next step is to apply LSB watermarking. The next step is to transform the watermarked input image into binary format.

The last step is to save the binary picture pixel values to a text file. Use MATLAB to do this.

In order to carry out cryptographic procedures, a key is the fourth need. To make this key, the LFSR is essential.

Final point number five: Verilog code is used to carry out cryptographic operations like encryption and decryption. It can read MATLAB text files.

Lastly, the Verilog text files include the outcomes of the encryption and decryption

ISSN: 2366-1313

procedures, which may be used for output verification.

To have MATLAB reconstruct the pixels, the seventh step is to utilize the encrypted and decrypted binary pixel values from the text file. These pixel values may be used to produce encrypted and decrypted pictures.

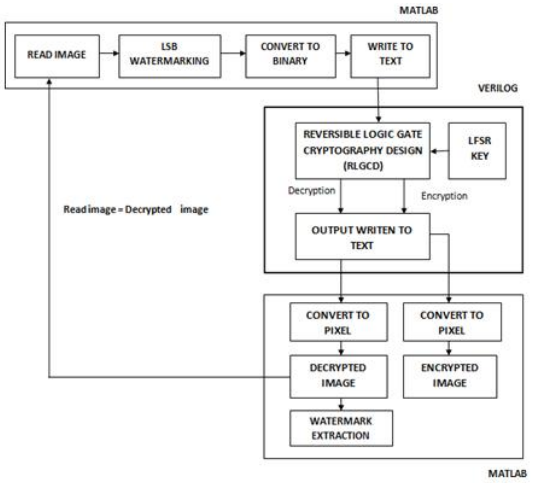
Step 8 ensures that the encrypted version of the input picture is indistinguishable from the original.

Step 9 involves removing the watermark from the encrypted picture.

Step 10 involves testing the FPGA's capabilities using the Verilog code.

Now we're going to talk about LSB watermarking.

An efficient and lightweight method of watermarking is the least significant bit (LSB) method. Last significant bit (LSB) watermarking accomplishes the goal by including the data bits used to create the watermark into the original image's pixel bits. The results of these alterations are invisible to the human eye because of this. Every 128x128 pixel has the potential to store one bit of data, hence any image that fits inside those dimensions might potentially support a 128-bit image.



All secret information totals 163,844 bits. Even if noisy modifications such as lossy compression or cropping could be immune to simple LSB watermarking, a more advanced attacker might still be able to retrieve the altered bits. Therefore, the watermark in this work is added by using the third and fourth LSB of the original picture [10]. Such LSB sites being injected with sensitive information is quite unlikely. The safety of the system would be improved as a result. We begin by bringing the original 128x128 input picture into MATLAB. We next convert the watermark to its binary equivalent. Beginning with the first least significant bit (LSB), the data is subsequently added to the third and fourth LSPs of the image. A 5-pixel space separates the first eight pixels' third and fourth least significant bits (LSBs), which is the starting point for the binary watermark data length. Since 817 is the maximum length for a binary watermark, we will be asked to modify the data if it

ISSN: 2366-1313

exceeds this limit. The data used for the watermark is stored in the third and fourth least significant bits (LSBs), with a five-pixel gap after the data length. The final result is a copy of the input image with a watermark added. Decryption is the first step in watermark extraction, and the last step is to apply the watermark backwards. The hidden data from the third and fourth LSBs is calculated by starting with the first pixel and jumping five pixels. It measures the length starting at the ninth pixel. The third and fourth long short bits (LSBs) are then extracted using the same method. After we convert the binary data to its character counterpart, we will display the watermark that we applied. The quality of the supplied picture, whether in colour or greyscale, is irrelevant. The blue region of a colour image is where the watermark should be placed. This is mostly due to the fact that it downplays the significance of what the human eye detects.

Department D. Approach to Encryption

The encryption procedure is shown in Figure 3. Therefore, the values of pixels are 8-bit binary words, structured as follows: i[0], i[1], i[2], i[3], i[4], i[5], i[6], i[7]. The most significant bit (MSB) input feeds two SCL gates, whereas the least significant bit (LSB) input feeds one.

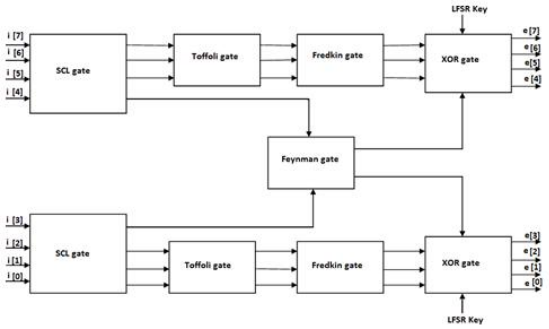


Fig. 3. Encryption block.

Data that has been input. The SCL gate operation requires these four inputs and can only be finished with a four-bit output. The following SCL gate takes its three left-side-bit (LSB) outputs and utilizes the Toffoli gate function to generate three separate bits of output. Similar to how the SCL gate uses the first three MSB values to create its three output bits; the Toffoli gate does the same. The output bit of an above-and-below SCL gate is used to carry out the Feynman gate operation. Because it relies on its outputs, the Fredkin gate follows the two Toffoli gates. The outputs of the Feynman and Fredkin gates are connected to the XOR gates so that an LFSR key may be used to conduct an XOR operation. The output of the XOR gate then provides the encrypted binary values of the image pixels e[0], e[1], e[2], e[3], e[4], e[5], e[6], and e[7].

A. Decryption process

ISSN: 2366-1313

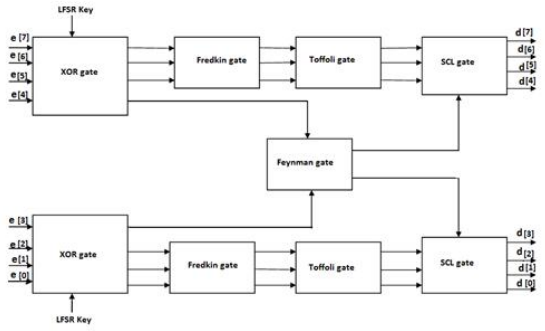


Fig. 4. Decryption block.

The decryption method is shown in Figure 4. Simply put, decryption is the opposite of encryption. Thus, blocks used in decryption operations take as input the results of encryption processes. An XOR operation is performed using the encrypted pixel bits and the key that was created by the LFSR. Following the sequential execution of the four reversible gate operations, the decrypted output is generated at the SCL gate output. The unencrypted eight-bit pixel values are [0], [1], [2], [3], [4], [5], [6], and [7]. The encrypted and decrypted binary values are stored in a text file as output. It is possible to generate encrypted and decrypted pictures using MATLAB's output text files.

Be sure to sign up for Shift F of Linear Feedback.

A linear feedback shift register, often known as a random number generator (RFSG), may be used to produce arbitrary key patterns. This four-bit LFSR is constructed using an XNOR gate in conjunction with four flip-flops. Giving each flip-flop a random value is the first step. "Seed value" refers to the randomly selected word for the first 8 bits. In order to generate a random test pattern, the LFSR employs a feedback loop and bits to alter the seed value when clocked. The production of arbitrary encryption keys is one typical use of LFSRs [11]. Because of this, the LFSR has potential uses in stream ciphers and other low- and high-speed applications. There is a direct correlation between the feedback polynomial and the amount of sequences generated at random. Since it is really just a simple counter with feedback, an LFSR can count up to 2n-1 when built with its maximum length feedback polynomial [12]. The size of a key is an indication of

You may see the 128x128 input pepper images in Figure 6. With MATLAB, we may convert the values of image pixels to a binary format. As shown in Figure 7, the data OUTPUT is converted to a binary value and thereafter used as a watermark. Figure 8 shows the original picture's binary value, whereas Figure 9 shows the input image's binary value with the watermark applied. In Figure 10, you can see the input image that has a watermark.

ISSN: 2366-1313

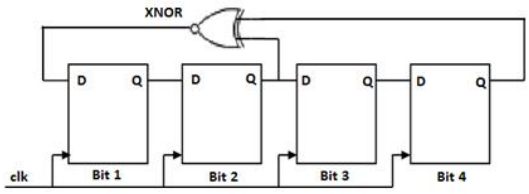


Fig. 5. Linear Feedback Shift Register.

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 watermarking are performed in MATLAB 2018.

Fig. 6. Original input image.

The 128x128 input pepper pictures are shown in Figure 6. The binary representation of picture pixel values is achieved in MATLAB. The data OUTPUT is transformed to a binary value and then used as a watermark, as seen in Figure 7. Figure 8 displays the binary value of the original picture, whereas Figure 9 shows the binary value of the input image with the watermark. The input picture with the watermark is seen in Figure 10.

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

Fig. 7. Binary value of watermark.

Fig. 8. Binary value of original image

Fig. 9. Binary value of watermarked input image

Fig. 10. Watermarked input image.

The RLGCD, which is built in Verilog, receives its input from a text file that contains the values of the binary watermarked images. Shown in Figure 11 is the time diagram of RLGCD, which encompasses both the encryption and decryption processes. According to Xilinx ISE, the

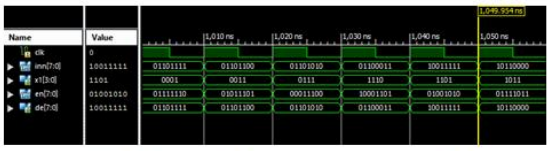


Fig. 11. Timing diagram of cryptography process using RLGCD.

The "readme" function is used to read the file. Due to the input picture size of

ISSN: 2366-1313

128x128, the resultant binary value depth contains 16,384 words. The input is denoted as "inn", and the key that is produced via LFSR is designated as "x1". The end results of an encryption operation are "en" and those of a decryption operation are "de," respectively. The timing graphic makes it very evident that the input and decryption pixel values are identical. In order to display the encrypted and decrypted images, MATLAB reads the "en" and "de" variables. The decrypted picture is identical to the input image, as seen in Figure 13 and Figure 12, which exhibit the encrypted and decrypted images, respectively.

Fig. 14. Watermarked input image

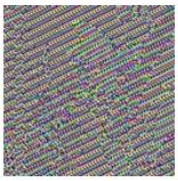


Fig. 15. Encrypted image

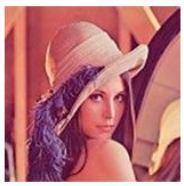


Fig. 16. Decrypted image

TABLE I
FPGA PERFORMANCE OF VARIOUS DESIGNS

Target FPGA	Circuit	LUT	Flip-flop	Slice	Frequency (MHz)
Virtex 7	Isogenies- MC [13]	185,871	218,012	77,425	158.5
Virtex 7	Scalable isogenies[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	1%	
Number of 4 input LUTs	37	9,312	1%	
Number of occupied Slices	42	4,656	1%	
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	69	9,312	1%	
Number used as logic	37			
Number used as a route-thru	32			
Number of bonded IOBs	33	232	14%	
Number of RAMB16s	8	20	40%	
Number of BUFGMUXs	1	24	4%	
Average Fanout of Non-Clock Nets	4.01			

Fig. 18. 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.

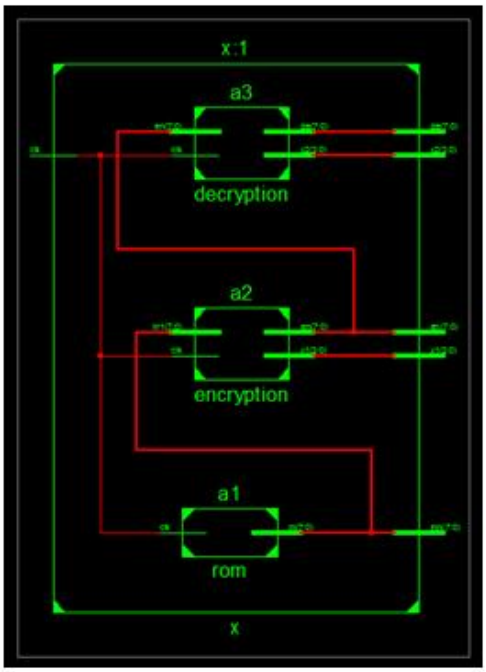


Fig. 19. RTL of RLGCD

V CONCLUSION

An LFSR key with watermarking implemented in a reversible logic gate cryptography design is presented in this

ISSN: 2366-1313

paper. This innovative cryptography system design makes use of reversible gates such as 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. After running the input picture read operation, watermarking, and conversion to binary format in MATLAB, the values of the binary representation are saved to a text file. This data

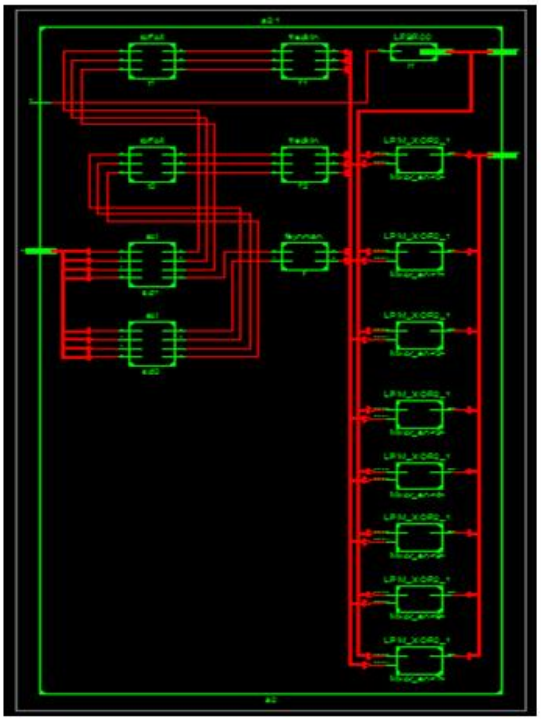


Fig. 20. RTL schematic of encryption block

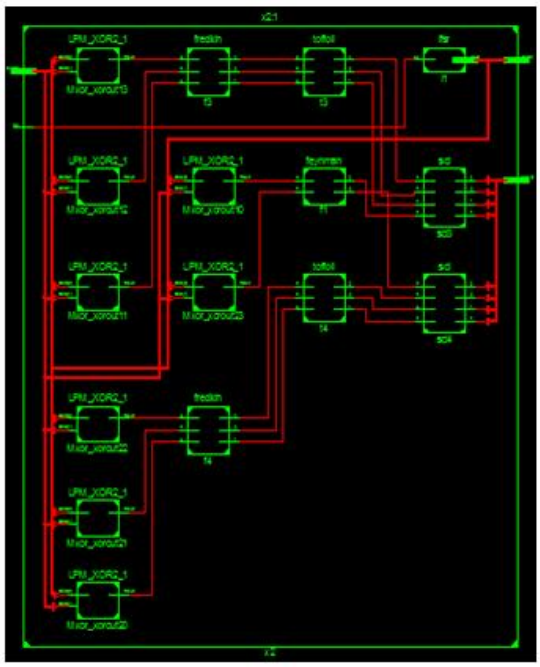


Fig. 21. RTL schematic of decryption block

Xilinx ISE is used to read the pixel values. Xilinx applications include the RLGCD architecture, which comprises an LFSR, an encryption block, and a decryption block. Images in either greyscale or colour may be accommodated by this design. Data security is enhanced with the use of the LSB approach for watermarking. The Spartan3E XC3S500E device's Xilinx performance result outperforms competing systems hands down.

One essential component of the new area of quantum computing is reversible logic gates. Each piece of work that makes use of reversible logic gates contributes to the advancement of quantum logics. Future ASIC deployments should be successful with RLGCD as it has been successfully developed using verilog code.

ISSN: 2366-1313

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 pro- cess, 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, Septem- ber 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 Azarder- akhsh, Mehrdad Nojoumaian, "Reliable hardware architectures for cryp- tographyic 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. Abduullah 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

ISSN: 2366-1313

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.