International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

SECURE VISUAL DATA PROCESSING IMAGE ENCRYPTION AND DECRYPTION THROUGH REVERSIBLE LOGIC GATES

YEDURU VENKATA PRAVEEN KUMAR REDDY [1], M.PRAVEEN RAJU[2]
[1]UG Students, Department Of Ece, Pvkk Institute Of Technology Anantapur.
[2]Assistant Professor, Department Of Ece, Pvkk Institute Of Technology Anantapur

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.

1. INTRODUCTION

Reversible logic has received great attention in the recent year s due to their ability to reduce the power dissipation which is the main requirement in low power VLSI design. It has wide applications in low power CMOS and Optical information processing, DNA computing, quantum computation and nanotechnology. Irreversible hardware computation results in energy dissipation due to information loss. According to Landauer's research, the amount of energy dissipated for every irreversible bit operation The heat generated due to the loss of one bit of information is very small at room temperature but when the number of bits is more as in the case of high speed computational works the heat dissipated by them will be so large that it affects theperformance and results in the reduction of lifetime of the components In 1973, Bennett showed that KTln2 energy would not dissipate from a system as long as the system allows the reproduction of the inputs from observed outputs. Reversible logic supports the process of running the system both forward and backward. This means that reversible computations can generate inputs from outputs and can stop and go back to any point in the computation history. A circuit is said to be reversible if the input vector can be uniquely recovered from the output vector and there is a one to one correspondence between its input and output assignments, i.e. not only the outputs can be uniquely determined from the inputs, but also the inputs can be recovered from the outputs Energy dissipation can be reduced or even

eliminated if computation becomes Information lossless

2. THE CONCEPT

Reversibility in computing implies that no information about the computational states can everbe lost, so we can recover any earlier stage by computing backwards or uncomputing the results. This is termed as logical reversibility. The benefits of logical reversibility can be gained only after employing physical reversibility. Physical reversibility is a process that dissipates no energy to heat. Absolutely perfect physical reversibility is practically unachievable. Computing systems give off heat when voltage levels change from positive to negative: bits from zero to one. Most of the energy needed to make that change is given off in the form of heat. Rather than changing voltages to new levels, reversible circuit elements will gradually move charge from one node to the next. This way, one can only expect to lose a minute amount of energy on each transition. Reversible computing strongly affects digital logic designs. Reversible logic elements are needed to recover the state of inputs from the outputs. It will impact instruction sets and high level programming languages as well. Eventually, these will also have to be reversible to provide optimal efficiency.

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 Moore's law [1]. But with higher integration and scaling the amount of heat that is dissipated also increases. Landauer's work [2] showed that for each bit of data that is lost there will be a heat dissipation in the range of KTln(2). Where, K is the Boltzman 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 parts 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.

Data security is importance in present time as lots of information is being communicated via network. A suitable methodology for privacy transformation is best to make a data protected over network. Different methods are implemented in order to protect the sensitive data. Now a days most of the data is secured by the technique of encryption and certificates. Most of methods are based on cryptography technique. Multi-level encryption is a new concept that is used for making the system more secure than existing cryptosystems. Multi level encryption is the process of encrypting the plain text with one or more time with same of different no of keys. It makes the process more complex and powerful than existing.

2. LITERATURE SURVEY

Architecture Design and VLSI Hardware Implementation of Image Encryption/ Decryption System Using Re-configurable 2-D Von Neumann Cellular Automata by Rong-Jian Chen, Yi-TE Lai, Jui-Lin Lai

The first architecture design and VLSI hardware implementation of image encryption/decryption system using re-configurable two-dimensional (2-D) von Neumann cellular automata (CA) is presented in this paper. Its encryption scheme is based on replacement of the pixel values using a progressive cellular automata (CA) substitution. In our scheme, we used the re-configurable 2-D von Neumann CA to generate high quality random sequence as key stream. To enhance the flexibility of our system, we used 16 x 16 re-configurable 2-D von Neumann CA which produces 1 set (256) CA, or concurrently produces 4 sets (64/set) 8 x8 CA and 16 sets (16/set) 4x4 CA, respectively. We have accomplished simulations of our image encryption/decryption system by using CADENCE tools. We also have completed the circuit synthesis using the SYNOPSYS tools with the T SMC 0.18um cell-library. The area size was 15.6816 MM , and the maximum operation frequency was 100 MHz with 27.74 MW total dynamic power. It shows that the architecture of the proposed image encryption/decryption system is suitable for VLSI realization.

A Nonlinear equation based cryptosystem for image encryption and decryption by Rithmi Mitter, M. Sridevi Sathya Priya

In this paper a new approach for image encryption and decryption using chaotic map and a Non Linear equation known as BB equation is described. Chaotic maps have been widely used in data encryption. Various chaos Mapbased encryption and decryption algorithms are used but are found to be insecure. Hence a new method is implemented based on BB (Brahmagupta-Bhaskara) equation which is combined with chaos to give a no linear dependency and thus improved security. VLSI architecture for the proposed algorithm is designed and realized using Xilinx ISE VLSI software for hardware implementation

VLSI realization of a secure cryptosystem for image encryption and decryption by K. Durgha Rao, Ch. Gangadhar

Chaotic maps have been widely used in data encryption. However, a number of chaos-based algorithms have been shown to be insecure. The application of BB equation for encryption is reported in a recent article. In

this paper, new algorithms based on chaos and BB equation are reported for image encryption and decryption. The algorithms are illustrated through an example. For practical use, VLSI architectures of the proposed algorithms are designed and realized using Xilinx ISE VLSI software for hardware implementation. Further, the hardware complexity of the proposed algorithms is compared with the algorithm reported in [6]

Implementation of encryptionand decryptionAlgorithms for Security of Mobile Devices by :B.V.Varun, Abhishek M.V., Akshay Chanabasappa Gangadhar

Progress of mobile communication and VLSI technology has aided in development of smart devices. These devices process the information of various formats and sizes in a limited amount of time. This information will be either stored in the devices or in cloud, hence there is a need for some kind of methodology to process and secure the data. Implementation of new algorithms to secure the information is always of immense interest. These algorithms will improve the performance of smart devices and helps for better human-machine interaction. Generally, symmetric and asymmetric approaches are used to secure the data from unauthorized users or attacks. Considering the amount of delay and complexity involved in processing the data, various forms of algorithms are used. In this paper, we propose a novel algorithm to secure the data from vulnerable attacks. These algorithms can be implemented on various platforms. The experimental results demonstrate an improvement of 10% for contacts and 15% for the encryption of images as compared to other conventional approaches.

3. EXISTING SYSTEM

DES is a secret-key archetypal block cipher with block size of 64 bits. DES encrypts a block of 64-bit plaintext into 64-bit cipher text using 64-bit secret key (Left most bit of a block is bit one). Block diagram of the DES algorithm is shown in the Figure 1.

DES adopted in 1977 by the National Bureau of Standards now the National Institute of Standards and Technology (NIST), as Federal Information Processing Standard 46 (FIPS PUB 46).

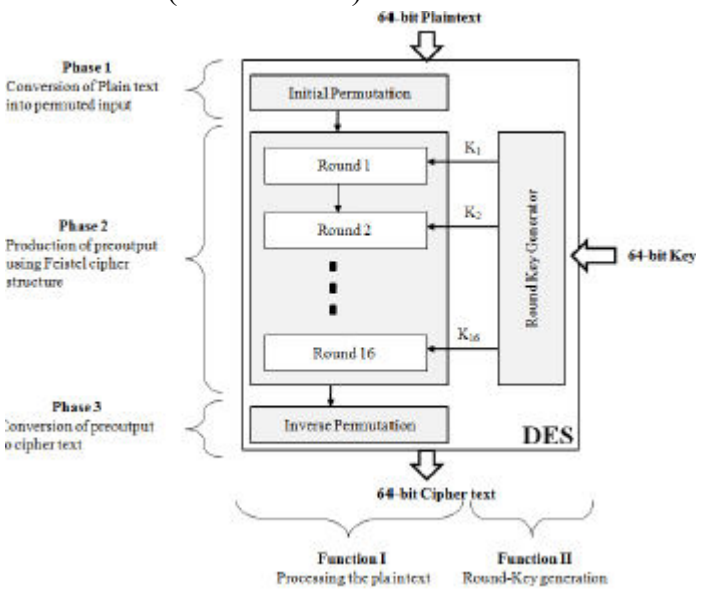

Figure 4: General block diagram of DES algorithm

A DES key consists of 64 binary digits ("0"s or "1"s) of which 56 bits are randomly generated and used directly by the algorithm. The other 8 bits, which are not used by the algorithm, may be used for error detection. The 8 error detecting bits are set to make the parity of each 8-bit byte of the key odd, i.e., there is an odd number of "1"s in each 8-bit byte. A TDEA key consists of three DES keys, which is also

referred to as a key bundle. Authorized users of encrypted computer data must have the key that was used to encipher the data in order to decrypt it. The encryption algorithms specified in this standard are commonly known among those using the standard. The cryptographic security of the data depends on the security provided for the key used to encipher
and decipher the data.

Data can be recovered from cipher only by using exactly the same key used to encipher it. Unauthorized recipients of the cipher who know the algorithm but do not have the correct key cannot derive the original data algorithmically. However, it may be feasible to determine the key by a brute force "exhaustion attack." Also, anyone who does have the key and the algorithm can easily decipher the cipher and obtain the original data. A standard algorithm based on a secure key thus provides a basis for exchanging encrypted computer data by issuing the key used to encipher it to those authorized to have the data.

DES Encryption process has two functions

A. Processing the plaintext

B. Round-Key generation

A. Processing the plaintext

The processing of plaintext proceeds in three phases.

1. Conversion of Plain text into permuted input

2. Production of preoutput using Feistel cipher structure

3. Conversion of preoutput to cipher text

1. Conversion of Plain text into permuted input

The 64-bit plaintext passes through an initial permutation (IP) that rearranges the bits to produce the permuted input, which is split into two 32-bit halves L0 and R0 where first 32 bit is L0 and next 32-bit is R0.

Permutation is keyless and can be predetermined. This has no cryptographic significance but included to facilitate loading blocks in and out of hardware and to make DES run slower in software.

2. Production of preoutput using Feistel cipher structure Most symmetric block encryption algorithms are based on Feistel [14] structure. Feistel proposed the use of a cipher that alternates substitutions and permutations which is a practical application of a product cipher that alternates confusion and diffusion functions producing Substitution-Permutation Network (SP Network) [15].

3. Conversion of preoutput to cipher text The preoutput is passed through a permutation (IP-1) that is the inverse of the initial permutation function, to produce the 64-bit cipher text. This stage has no cryptography significance in DES. The initial and final permutations are straight P-boxes that are inverses of each other.

B. Function 2- Round-Key generation

DES takes 64-bit key as input. Among 64-bit key only 56 bitsare effective and used directly by the algorithm. The other 8 bits, which are not used by the algorithm, may be used for error detection or set arbitrarily or can be ignored [13]. The 8 error detecting bits are set to make the parity of each
8-bit byte of the key odd, i.e., there is an odd number of "1"s in each byte. The round-key generator creates sixteen 48-bit round/sub keys out of a 56-bit cipher key. The round key generation block is shown in Figure 2

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

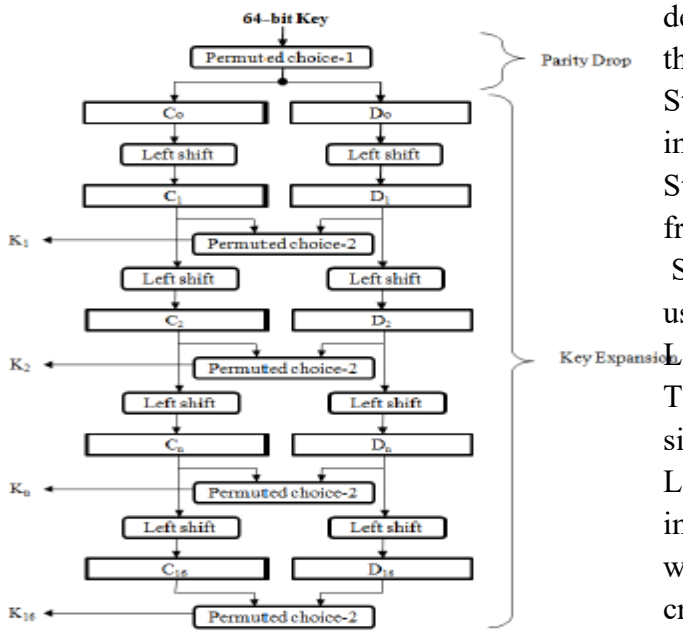

Fig 2 Key Generation

4. PROPOSED SYSTEM

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.

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

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 can easily extract the changed bits. 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 length of 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.

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 original motivation was that reversible gates dissipate less heat (or, in principle, no heat). In a normal gate, input states are lost, since less information is present in the output than was present at the input. This loss of information loses energy to the surrounding area as heat, because of thermodynamic entropy. Another way to understand this is that charges on a circuit are grounded and thus flow away, taking a small quantity of energy with them when they change state. A reversible gate only moves the states around, and since no information is lost, energy is conserved.

The RLG that are used to design this new cryptography system includes Feynman gate, Fredkin gate, Toffoli'sgate and SCL gates and are shown in Fig.1

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

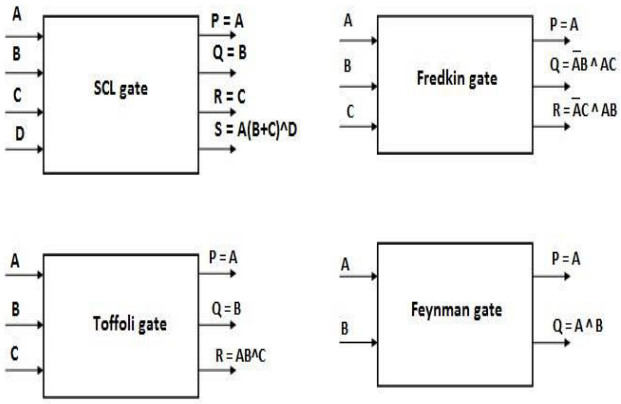

Fig. 3. Block diagram of RLGs

Encryption process:

The encryption process is shown in Fig.2. 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.

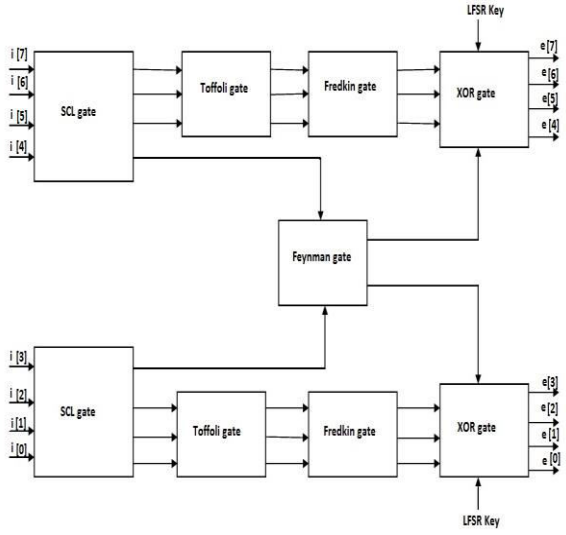

Fig. 4. Encryption block

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 providesthree output bits. One of the output bits 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].

Decryption process:

The process of decryption is shown in Fig.3. 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 outputs are obtained at the SCL gate output. The decrypted output eight bit pixel values ared[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.

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

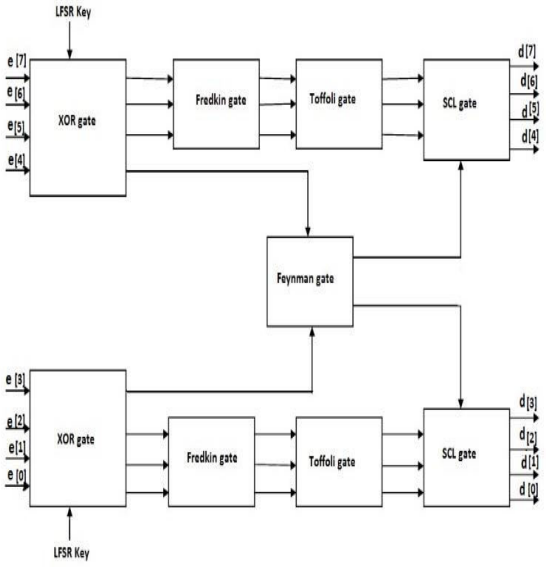

Fig. 5. Decryption block

5. SIMULATION RESULTS

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 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.1. shows the original image and the watermarked input image respectively

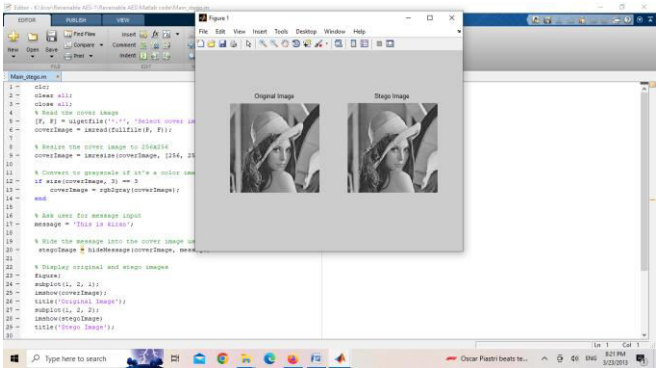

Fig 6 Shows the original image and the watermarked input image

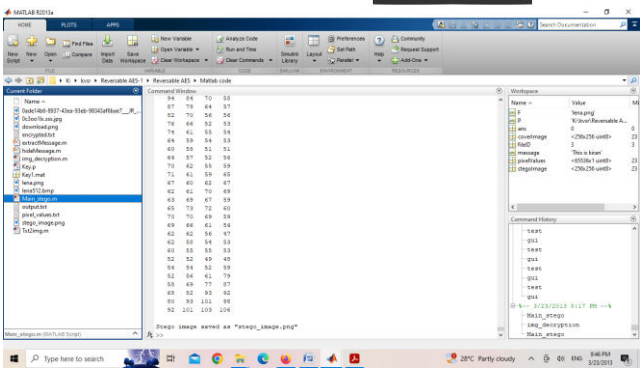

Fig 7 Show the binary values of the input image

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.8 and 9. In Xilinx ISE the text Fig. 2. 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.

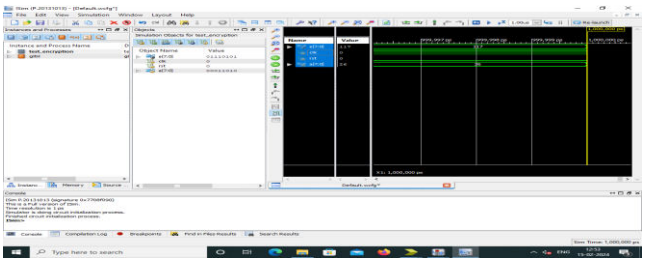

Fig 8 Encryption Simulation Result

Fig 8 shows the simulation result of Encryption,in this we applied input plane test ,clock and reset based on that encrypted output is generated.

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

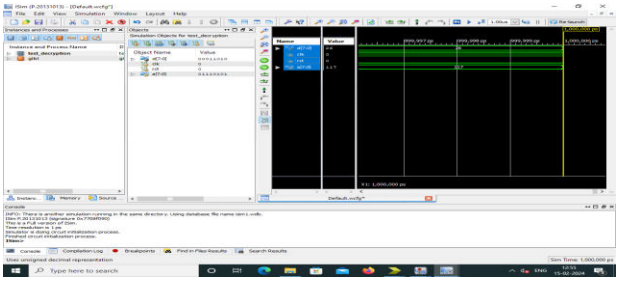

Fig 9 Decryption Simulation Result

Fig 9 shows the simulation result of Decryption, in this we applied encrypted input,clk and rst based on that decrypted output is generated.

The ”en” and ”de” variables are read in MATLAB to present the encrypted image and the decrypted image. Fig.7.3 and Fig.7.4 shows the encrypted and decrypted image respectively and it shows that the decrypted image is as same as the input image.

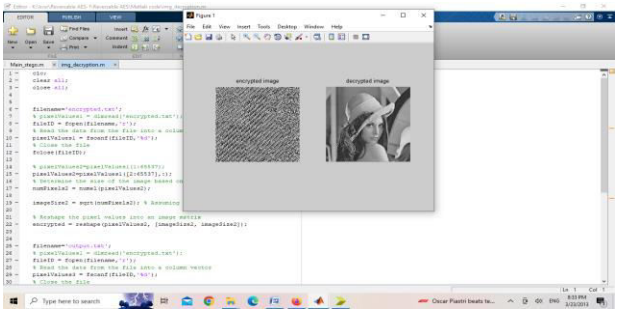

Fig 10 Decrypted Image

6. CONCLUSION AND FUTURE WORK

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.

those binary values are written into a text file. This input pixel values are read using Xilinx ISE. The RLGCD architecture consisting of LFSR, encryption block and decryption block is implemented in Xilinx software. This architecture is suitable for both gray scale images and color images. The watermarking using LSB technique is performed to improve the security of the data. The Xilinx performance result for Spartan3E XC3S500E device gives a far better performance as compared to other existing systems.

The reversible logic gates are the fundamental requirement in the emerging field of quantum computation. Thus each work using the reversible logic gates will help to move forward in the field of quantum logics. Since RLGCD is successfully implemented using verilog code it can be effectively deployed on ASIC in future.

REFERENCES

[1] Gordon E. Moore, “Craming 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] Dr.M.V. Sruthi “Color images authentication and copyright protection in blind dual watermarking” in Journal of engineering sciences

[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 cryptographyic 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 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, "Postquatum 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.