"Design and Implementation of Image Encryption and Decryption Based on Reversible Logic Gates"

M. Lakshmi Prasanna1, T.Anusha2,Y.L Ajay kumar3

1PG Scholar, 2Assistant Professor, 3Professor, Dept of ECE, Anantha Lakshmi institute of technology & sciences, Anantapuramu.

ABSTRACT

The paper presents a new approach to cryptography that tackles the common issues of high area and power consumption found in traditional systems. It uses reversible logic gates for both encrypting and decrypting data, with a Linear Feedback Shift Register (LFSR) to generate the encryption key. To make the system even more secure, it also adds a layer of protection through watermarking, using the Least Significant Bit (LSB) technique. When tested on FPGA hardware, the proposed design shows a significant boost in efficiency compared to traditional methods.

Keywords: Reversible Logic Gate Cryptography Design (RLGCD), Linear feedback shift register (LFSR), Field Programmable Gate Array (FPGA), Watermarking

I.NTRODUCTION

Cryptography is the practice of securing information by transforming it into an unreadable format, ensuring its confidentiality. This involves two main processes: encryption, which converts the original data into a scrambled format, and decryption, which restores the data to its original, readable form. In VLSI design, heat dissipation has become a significant challenge, especially as the size of integrated circuits (ICs) continues to shrink and the number of transistors increases, in line with Moore's Law. While these advancements lead to higher performance and more compact devices, they also result in greater heat generation. Landauer's principle reveals that for every bit of data lost during computation, a specific amount of heat, quantified as KTln(2) (where K is the Boltzmann constant and T is the temperature in Kelvin), is dissipated. This heat dissipation is an inevitable consequence of traditional, irreversible computing systems. However, Bennett's work proposed that this issue could be mitigated or even eliminated by using reversible computing systems, where no information is lost during computation, and the system's entropy remains unchanged. In reversible computation, the energy dissipated is minimal, as there is no decrease in information or increase in heat, making it a potential solution for reducing the heat dissipation challenges in modern VLSI designs.

In data and telecommunications, cryptography is essential for securing communication over untrusted channels, where data is vulnerable to hacking. A cryptographic system needs to provide high security while also ensuring low power consumption. Implementing cryptography using reversible logic gates offers an effective solution to meet both of these requirements. This paper presents a Reversible Logic Gate Cryptography Design (RLGCD), which integrates reversible technologies to enhance energy efficiency, outperforming conventional cryptographic systems. This design is particularly useful for applications in fields like healthcare, banking, and government. The cryptographic key is generated using an LFSR, and the FPGA performance of the RLGCD architecture shows significant improvements over existing methods. In today's world, data privacy is crucial due to the increasing threat of hackers. To enhance security, watermarking using the LSB technique is applied to input images. Watermarking embeds a hidden pattern within the original data to verify its authenticity. The rest of the paper is structured as follows: Section II reviews related works, Section III explains the proposed RLGCD architecture, Section IV presents experimental results, and Section V provides the concluding remarks.

II. EXISTING METHOD

A. Fault resilient light weight cryptography

Embedded systems with sensitive components, such as RFID tags and nano-sensors, require the adoption of lightweight block ciphers. Error detection schemes for these ciphers are explored in [5]. In this work, the XTEA (eXtended TEA) block cipher, known for its speed and efficiency, is utilized. XTEA relies on simple operations like addition, XOR, and shifting, offering minimal code size, reduced memory footprint, and lower computational overhead. While these proposed methods enhance reliability, the trade-off is a relatively high error rate, which limits their overall accuracy when using the XTEA approach.

B. Designing a Secure DES Using Reversible Logic Gates

The security design of the Data Encryption Standard (DES) using Reversible Logic Gates (RLG) incorporates a reversible logic gate shift register and a four-bit counter. By utilizing RLG, this approach enhances data security while reducing power consumption. However, the specific design of the RLG has not been detailed, and performance evaluations for this method have not been conducted, leaving room for further exploration and optimization. This approach holds promise for improving the efficiency and security of DES, but additional research is needed to fully assess its effectiveness and practical implementation. *Security analysis and enhanced dynamic block cipher* In this work, the security of the ciphertext is enhanced by dynamically adjusting the S-box dimension and the number of registers based on the required security level. The encryption process improves safety through confusion substitution in the S-box, followed by four steps of matrix transformation to disorder the data blocks. To increase the diffusion of the ciphertext, cyclic displacement of bytes is performed using a column ambiguity function. Additionally, an LFSR (Linear Feedback Shift Register) is used to generate a dynamic secret key, improving its stochastic characteristics with each iteration. This approach offers high scalability, but it becomes challenging to generate the S-box when the selected dimension is an odd number, leading to longer encryption and decryption times.

A. Dependable hardware designs for encryption algorithms used in block ciphers

This study examines two block ciphers, HIGHT and LED, suitable for use in authenticated encryption algorithms. HIGHT features a Feistel network structure, making it ideal for low-power, low- complexity embedded systems. LED is an efficient AES-like cipher. While the study demonstrates high error coverage and efficiency, it does not address the detection of both permanent and transient faults.

III. PROPOSED METHOD

A. Reversible Logic Gates (RLGs)

Reversible Logic Gates (RLGs) are circuits where the number of inputs equals the number of outputs, with a unique one-to-one mapping between them. This allows for the recovery of the input pattern from the output, ensuring no information is lost during computation. For instance, if an input pattern like 110 produces an output of 001, applying 001 as input will yield the original 110 output, demonstrating a reversible operation. In contrast, traditional combinational logic circuits cause data loss during computation, leading to equivalent heat dissipation due to the second law of thermodynamics, which asserts that once information is lost, it cannot be recovered. Reversible computation, therefore, facilitates zero power dissipation, as it prevents any decrease in system entropy. When designing Reversible Logic Gates (RLGs), several important constraints and optimization goals must be considered to ensure efficient performance. These constraints can be generalized as follows:

No Fanout: In reversible logic design, fanout is not allowed. This means that each output of a gate can only be used once, which prevents any duplication of signals in the circuit. This constraint ensures that the design remains reversible and complies with the basic principle of reversibility, where information is not lost.

Minimize Quantum Cost: The quantum cost of a reversible gate or circuit refers to the number of basic quantum gates required to implement the logic of ooperation. To make the circuit more efficient, the quantum cost should be minimized, as fewer quantum gates typically lead to reduced resource consumption and better performance in quantum computing applications.

1. **Minimize Garbage Outputs**: In reversible logic design, "garbage outputs" are unwanted outputs generated during the computation process, which do not contribute to the final result but are necessary to maintain reversibility. The goal is to minimize these outputs because they increase the complexity of the circuit and consume additional resources.
2. **Minimize Gate Level**: The design of the reversible logic circuit should aim to minimize the number of gate levels (the depth of the circuit). A lower gate level implies faster computation and better overall efficiency, as fewer sequential operations are needed to compute the result. These general principles guide the development of reversible logic circuits, especially in contexts where resource efficiency, such as in quantum computing, is crucial. Reversible logic is important because it theoretically avoids energy dissipation due to the reversibility of computation, making it a promising field for low-power and quantum applications. The RLGs used in the design of this new cryptography system include the Feynman, Fredkin, Toffoli, and SCL gates, as shown in Fig. 1. Reversible logic circuits are essential for resource efficiency, particularly in quantum computing, as they avoid energy dissipation by ensuring computations are reversible. This eliminates information loss, making reversible logic ideal for low-power applications and quantum systems, where minimizing energy consumption is crucial for optimal performance. Reversible logic circuits minimize energy dissipation, prevent information loss, and are crucial for low-power quantum computing systems.

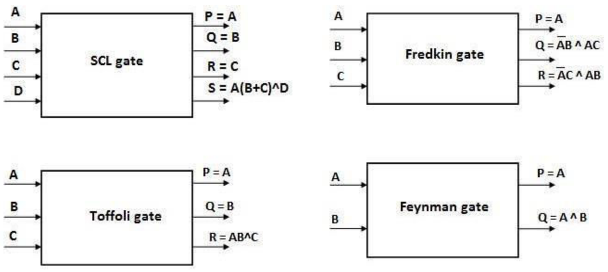

Figure 1. Block diagram of RLGs.

B. Block diagram

Fig.2 shows the block diagram of the entire cryptography process. Below, we explain the working principle of the proposed Reversible Logic Gate-based Cryptography Design (RLGCD).

Generalized Steps for the Cryptography Process: Input Image Processing: The input image is read and prepared for watermarking.

Watermarking: LSB (Least Significant Bit) watermarking is applied to the image, and then the watermarked image is converted into a binary format.

- Text File Conversion: The pixel values of the binary image are written into a text file for further processing.
- Key Generation: A cryptographic key is generated using a Linear Feedback Shift Register (LFSR) for encryption and decryption processes.
- Cryptography Implementation: The text file containing binary image data is input into Verilog code, where encryption and decryption operations are performed.
- Output Verification: The results of the encryption and decryption are stored in separate text files for verification.
- Image Reconstruction: Using MATLAB, the encrypted and decrypted pixel values are reconstructed from the text files, generating the respective encrypted and decrypted images.
- Image Comparison: The decrypted image is compared to the original image to verify that they are identical.
- Watermark Extraction: The watermark is extracted from the decrypted image to ensure the integrity of the process.
- Performance Evaluation: The FPGA performance of the entire cryptographic system is evaluated through the Verilog code. This generalized process outlines the workflow for implementing a cryptography system with watermarking, encryption, decryption, and performance evaluation.

A. LSB watermarking

The Least Significant Bit (LSB) watermarking technique involves modifying the LSBs of an image's pixel values to embed secret data, typically without noticeable changes to the human eye. In this method, the third and fourth LSBs of the original image are used to enhance security, as these positions are less likely to be suspected by attackers. For a 128x128 image, this allows the embedding of up to 16,384 bits of data. The watermark embedding process begins by converting the watermark into binary and embedding it into the third and fourth LSBs of the image, starting from the first pixel with a gap of five pixels between them. The first eight pixels store the length of the watermark, with a maximum length of 817 bits. If the watermark exceeds this length, it prompts the user to rewrite the data. After embedding, the watermarked image is generated. During watermark extraction, the process is reversed. The length of the secret data is extracted from the first eight pixels, and then the watermark data is retrieved from the third and fourth LSBs. This binary data is converted back into the original watermark. If the image is in color, watermarking is performed on the blue component, as it is less sensitive to the human visual system. This method allows watermarking to survive basic transformations but remains vulnerable to more sophisticated attacks.

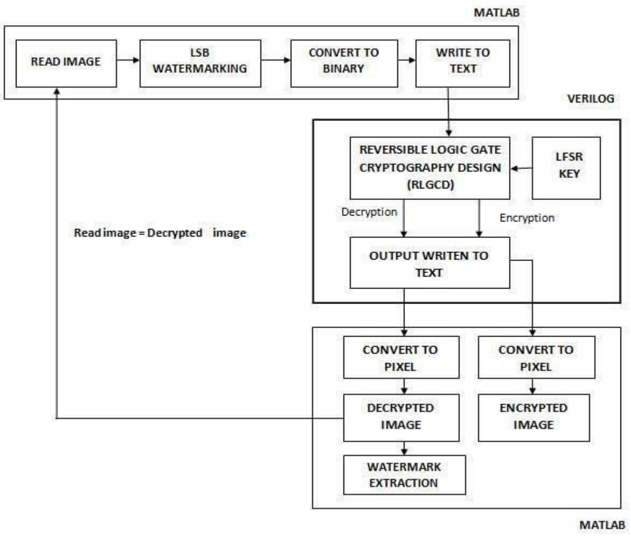

Figure 2. Overall block diagram.

B. Encryption process

The encryption process, as shown in Fig. 3, involves using the 8-bit binary values of the image pixels: i[0], i[1], i[2], i[3], i[4], i[5], i[6], and i[7]. The first four LSB (Least Significant Bits) are fed into a lower SCL gate, while the first four MSB (Most Significant Bits) are input into the upper SCL gate. This setup forms part of the encryption process, utilizing the SCL gates to manipulate the pixel data.

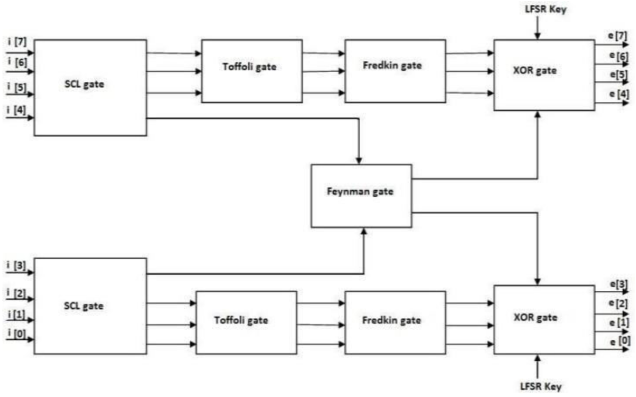

Figure 3. Encryption block

The encryption process involves a series of reversible logic gate operations on the input pixel bits. The 8-bit input pixels are divided into the Most Significant Bits (MSB) and Least Significant Bits (LSB). The first four LSBs are processed through a lower SCL gate, and the first four MSBs are processed through an upper SCL gate. These gates generate four result bits each. Next, the first three LSB outputs from the lower SCL gate are passed through a Toffoli gate, which produces three output bits, and similarly, the first three MSB outputs from the upper SCL gate are processed through another Toffoli gate to generate three more output bits. One output bit from both the upper and lower SCL gates is fed into a Feynman gate for further manipulation. Following this, both Toffoli gate outputs are connected to a Fredkin gate, which performs a controlled swap operation on the bits. The outputs from both the Fredkin and Feynman gates are then passed to XOR gates, where they are XORed with a Linear Feedback Shift Register (LFSR) key. Finally, the result of the XOR operation provides the encrypted 8-bit pixel values, completing the encryption process. This general process ensures that the pixel data is encrypted through a combination of reversible logic gates and key-based transformations, enhancing security.

Decryption process

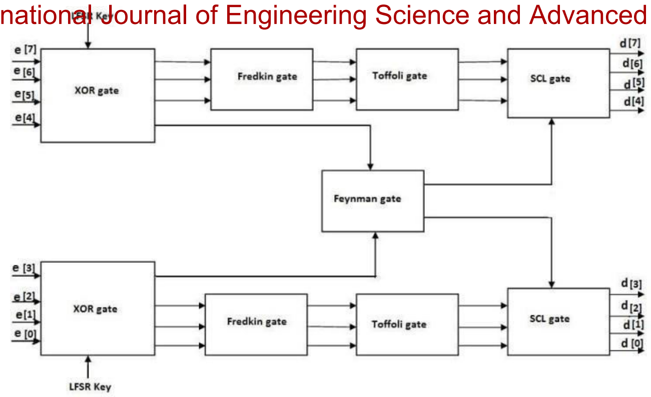

Figure 4: Decryption block.

The decryption process is the reverse of the encryption process. The encrypted pixel values are fed as input to the decryption block, which begins by performing an XOR operation with the key generated by the Linear Feedback Shift Register (LFSR). After the XOR operation, the encrypted bits undergo a series of reversible gate operations, performed one after the other, to reverse the encryption transformations.

The final decrypted output is obtained at the output of the SCL gate, resulting in the 8-bit decrypted pixel values: d[0], d[1], d[2], d[3], d[4], d[5], d[6], and d[7]. Both the encrypted and decrypted binary pixel values are written into a text file for further processing or verification.

This generalized process ensures that the encryption and decryption operations can be effectively reversed, allowing the original image to be recovered from the encrypted version.

C. Linear Feedback Shift Register

Summary of Linear Feedback Shift Register (LFSR) for Random Key Generation

A Linear Feedback Shift Register (LFSR) is a type of pseudo-random number generator commonly used to produce random key patterns in cryptographic systems. A 4-bit LFSR consists of four flip-flops and an XNOR gate for feedback. The process begins by loading the flip-flops with an initial seed value, which determines the starting sequence. When clocked, the LFSR shifts the bits while applying feedback, generating pseudo-random patterns. These properties make LFSRs useful in stream ciphers and various high- and low-speed applications. The length and randomness of the generated sequence depend on the feedback polynomial. Since an LFSR operates like a feedback-based counter, it can produce a maximum of 2^n - 1 unique states (wheren is the number of bits) when using a maximal-length polynomial ensuring efficient and secure key generation.

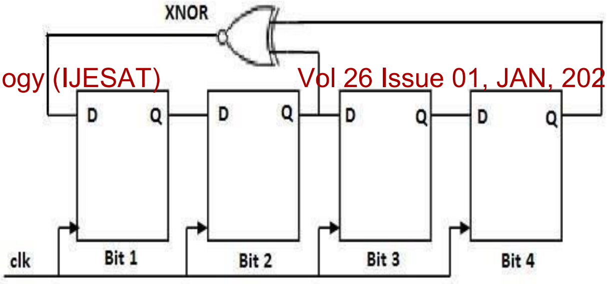

Figure5: Linear Feedback Shift Register.

In power and area-constrained cryptographic systems, using large keys increases network load, making efficiency a challenge. To address this, Linear Feedback Shift Register (LFSR)-based random key generation is employed, reducing overhead while maintaining security. The LFSR design (Fig. 5) enables confidential message transmission even with limited integrity. In this system, an input image undergoes cryptographic processing using dynamically generated LFSR keys. Implemented in Verilog, the system processes each watermarked image pixel individually, assigning a unique key for encryption and decryption. By regenerating the same key at the receiver, the original data is securely reconstructed. This approach ensures a lightweight, efficient, and highly secure cryptographic system, ideal for resource-constrained applications.

IV. RESULTS AND DISCUSSION

The RLG-based cryptography system was simulated using Xilinx ISE 14.7, while MATLAB 2018 handled the input image read operation and watermarking process. This hybrid approach combines hardware-efficient cryptographic key generation (using LFSR/RLG) with MATLAB-based image processing ensuring secure and optimized performance for the system.

Figure 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 shows the binary value of the original image and the watermarked input image respectively. The watermarked input image is shown in All paragraphs must be indented.

Figure:7 Watermarked input image

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.

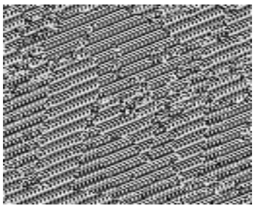

Figure:8 Encrypted Image

figure:9 Decrypted Image

V. CONCLUSION

This work introduces a cryptography system using reversible logic gates, such as Feynman, Fredkin, Toffoli, and SCL gates, combined with an LFSR (Linear Feedback Shift Register) key and watermarking. The system is designed to provide high security while minimizing power consumption. MATLAB is used to handle tasks like reading the input image, applying watermarking, and converting the image into binary format. The resulting binary values are then stored in a text file. The RLGCD architecture, which includes LFSR, encryption, and decryption blocks, is implemented using Xilinx software and is suitable for both grayscale and color images. Watermarking with the LSB technique is used to enhance data security. The performance results on the Spartan3E XC3S500E device show superior performance compared to other existing systems. The use of reversible logic gates, essential in quantum computing, positions this work as a step forward in quantum logic research. Since the RLGCD is implemented in Verilog, it can be effectively deployed on ASICs in the future.

VI. REFERENCE

1. T.Anusha, N. Vasantha, B. Rakshitha, K. Kusuma, M. Yogesh, C. Pranitha. Image encryption and decryption through reversible logic gates, April 2025.
2. Rolf Landauer, Irreversible and heat generation in the computing process, IBM Research and Development, vol.5, pp.183–191, July 1961.
C.H. Bennett, "Logical reversibility of computation" IBM Research and Development, vol.17, pp.525–532, 1973.
3. Saranya Karunamurthi, Vineya kumar Krishnasamy Natarajan, " VLSI implementation of reversible logic gates cryptography with LFSR key," Microprocessors and Microsystems, Elsevier, vol. 69, pp.68–78, September 2019.
4. 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.
Shikha Kuchhal , Rakesh Verma, "Security design of DES using reversible logic," Int. J. Compute. Sci. Netw. Security, vol. 15, no. 9, pp. 81–84, September 2015.
5. 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.
6. Srivatsam Subramanian, Mehran Mozaffari Kermani, Reza Azarderakhsh, Mehrdad Nojoumaian, "Reliable hardware architectures for cryptographyic block ciphers LED and HIGHT," in IEEE Trans. Compute.
7. 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.
8. 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.
9. B. Koziel, R. Azarderakhsh, M. Mozaffari Kermani, D. Jao, "Post- quatIEEE Trans.Circuits Syst.I, vol. 64, no. 1, pp. 86–99, Jan. 2017.