International Journal of Scientific Research and Engineering Development-– Volume 8 Issue 5, Sep-Oct 2025
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Energy-Efficient Cryptographic Hardware Using Reconfigurable Reversible Gates: Design, Simulation and Comparative Analysis

Tasmin Tripathy*, Dr.Baruna Kumar Turuk**, B Vikram Anand***
*(M.Tech Scholar, Department of ECE, GIET University,Gunupur)
**(Assistant Professor, Department of ECE, GIET University,Gunupur)
***(Assistant Professor, Department of EEE, GIET University,Gunupur)

Abstract:

The growing deployment of cryptography in cloud services, embedded devices and the Internet of Things has triggered renewed interest in hardware acceleration. Traditional complementary metal–oxide–semiconductor (CMOS) cryptographic engines provide high throughput but dissipate significant power because their Boolean gates irreversibly destroy information. Landauer's principle asserts that erasing one bit of information dissipates at least (kT) joules of heat, where (k) is Boltzmann's constant and (T) the absolute temperature. Reversible logic gates preserve information and in principle can operate with asymptotically zero energy dissipation [1],[2]. Recent work has shown that reversible implementations of classical ciphers can be realised on field-programmable gate arrays (FPGAs) using Toffoli and Fredkin gates [8]. This paper proposes a novel reconfigurable reversible gate (RRG) encryption architecture that supports symmetric cryptography with low power and resource overhead. The design cascades thirteen reversible gates to implement key scheduling, encryption and decryption, and it can be reconfigured dynamically for different cipher parameters. We implement the architecture on an FPGA, evaluate power, delay and resource utilisation for various operand widths, and compare the results with conventional CMOS designs. The RRG demonstrates lower dynamic power consumption and shorter critical paths than traditional implementations while maintaining correct cryptographic functionality. The contributions of this work include:

- Proposing a cascaded reversible architecture for symmetric encryption that can be reconfigured at runtime;
- Deriving low-level reversible gate implementations of fundamental cryptographic transformations;
- Evaluating power, delay and resource utilisation on an FPGA and comparing against conventional designs;
- Demonstrating that reversible logic can dramatically reduce power consumption without sacrificing throughput.

1 Introduction

Cryptography underpins secure communication in modern digital systems. Widespread use of the Advanced Encryption Standard (AES), RSA, elliptic-curve cryptography and emerging post-quantum primitives demand secure and efficient hardware accelerators. Conventional digital circuits employ irreversible Boolean logic: once a logical AND gate consumes its inputs, the original inputs cannot be recovered from the output. According to Landauer's principle, each bit that is erased dissipates heat proportional to (kT) [1]. As technology scales to nanometre dimensions, this fundamental thermodynamic limit becomes a dominant source of energy loss. **Reversible logic** offers a remedy: if computations are information preserving, no bit is destroyed, and theoretically the energy dissipation can approach zero [2]. The potential savings have motivated extensive research into reversible arithmetic and reversible cryptographic circuits.

Reconfigurable hardware such as FPGAs provides a convenient platform for prototyping cryptosystems. FPGAs comprise arrays of programmable logic blocks connected via routing channels. Designers configure the logic functions through a hardware description language (HDL) such as Verilog or VHDL [12]. Compared with application-specific

integrated circuits (ASICs), FPGAs offer rapid development and post-fabrication reconfiguration at the cost of higher area and power overhead. Hardware implementations of block ciphers can achieve significant speedups by exploiting pipelining and parallelism [9]. However, conventional FPGA implementations still employ irreversible logic and thus dissipate considerable power. This work addresses that challenge by mapping reversible gates to FPGA primitives, thereby realising an energy-efficient cryptographic accelerator.

This paper begins by reviewing the basics of symmetric and asymmetric cryptography and summarising previous research on reversible logic. Section 3 introduces fundamental reversible gates—Feynman, Toffoli and Fredkin—and their functional equations. Section 4 describes the proposed RRG encryption architecture and shows how individual reversible gates implement encryption and decryption. Section 5 presents the implementation results on an FPGA including power, delay and resource utilisation graphs. Section 6 discusses the implications of reversible cryptography and Section 7 concludes.

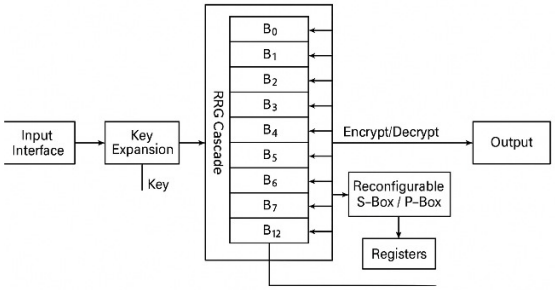


FIG. 1. SYSTEM BLOCK DIAGRAM—ENCRYPTION/DECRYPTION USING RECONFIGURABLE REVERSIBLE GATES

2 Background and Related Work

2.1 Cryptographic Preliminaries

Block ciphers transform fixed-length plaintext blocks into ciphertext using a key. **Symmetric-key algorithms** employ the same secret key for encryption and decryption [4]. Well known symmetric ciphers include AES and DES. The key must remain confidential; if an attacker obtains the key, all communications are compromised. **Asymmetric-key algorithms** (public-key cryptography) use mathematically related key pairs: a public key for encryption and a private key for decryption [4]. Asymmetric ciphers, such as RSA and elliptic-curve schemes, are widely used for key exchange and digital signatures. They offer easier key management because the public key can be distributed openly. However, asymmetric algorithms are typically slower and require more hardware resources than symmetric ciphers. This research focuses on symmetric block ciphers because the proposed RRG architecture implements reversible substitutions and permutations that naturally fit symmetric encryption.

2.2 Reversible Computation

In the early 1960s, Rolf Landauer observed that erasing one bit of information dissipates a minimum energy of (kT) [1]. Charles Bennett later showed that if computation is performed reversibly—so that outputs uniquely determine inputs—this energy dissipation can be avoided [2]. In reversible circuits every gate has the same number of outputs as inputs, and the mapping is bijective. Reversible circuits therefore cannot fan-out or have feedback; any duplication of signals must be performed using special gates. The Feynman (controlled NOT), Toffoli (controlled-controlled NOT) and Fredkin (controlled SWAP) gates constitute a universal set for reversible computing [7]. Reversible logic has applications in quantum computing, ultra-low-power digital systems, nanotechnology and cryptography.

Significant research has explored reversible arithmetic units. When a 4-bit adder was implemented using reversible gates, the delay reduced from 9.882 ns to 7.850 ns while power consumption decreased slightly [11]. These gains demonstrate the promise of reversible design for high-performance digital systems. Researchers have proposed reversible S-boxes and permutations for AES and other ciphers, mapping them to quantum gates for cryptanalysis and quantum implementations. More recently, the **reconfigurable reversible gate (RRG)** architecture cascades Fredkin and Toffoli gates to implement encryption and decryption [8]. The authors implemented the design in Verilog and reported low power

consumption and reconfigurability [8]. Our work builds on these ideas and extends them by analysing detailed performance metrics on an FPGA and comparing to conventional designs.

2.3 FPGA-Based Cryptography

FPGAs are widely used for cryptographic acceleration because they offer high throughput and can be reconfigured to support different algorithms. Hardware implementations of AES on FPGAs achieve significant speedups by exploiting sub-pipelining, lookup tables and mix-column transformations [9]. FPGAs provide a compromise between software flexibility and ASIC efficiency. They are well suited to deploy reversible logic because their programmable routing fabric can implement arbitrary permutations. Reconfigurable computing has been used to implement cryptography, digital signal processing and neural networks [12]. Nonetheless, few works have investigated reversible logic on FPGAs. This gap motivates our exploration of RRG-based encryption engines on reconfigurable hardware.

3 Fundamental Reversible Gates

Reversible gates serve as the building blocks of reversible circuits. In this section we define the Feynman, Toffoli and Fredkin gates and present their truth functions. Figure 2–4 depict schematic symbols for each gate.

3.1 Feynman (Controlled NOT) Gate

The **Feynman gate**, also called the controlled NOT (CNOT), is a 2×2 reversible gate. Given inputs (A) and (B), the outputs (P) and (Q) are:

where (p) denotes modulo-2 addition. Because (A) appears unchanged at the output and (B) is conditionally toggled based on (A), the gate is reversible. Figure 1 shows a neat schematic drawn for this paper. In reversible circuits, the Feynman gate is used for signal duplication and XOR operations [7].

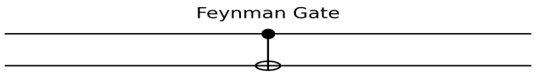


Figure 2: Feynman (controlled NOT) gate.

3.2 Toffoli (Controlled-Controlled NOT) Gate

The **Toffoli gate** is a 3×3 reversible gate and forms a universal gate for reversible computation. Inputs ((A,B,C)) map to outputs ((P,Q,R)) as follows [7]:

The gate passes the first two inputs through unchanged and flips the third input if both controls are logical one. The Toffoli gate can implement any Boolean function when combined with ancilla bits and has quantum cost 5 [7]. Figure 2 depicts a neat Toffoli gate diagram used in this paper.

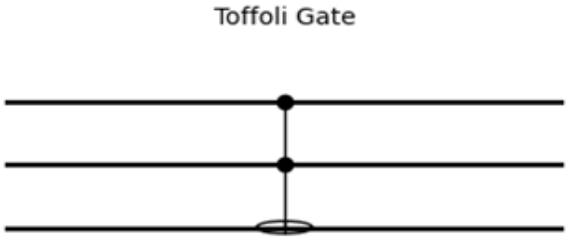


Figure 3: Toffoli (controlled-controlled NOT) gate.

3.3 Fredkin (Controlled SWAP) Gate

The **Fredkin gate** is a 3×3 reversible gate that conditionally swaps two inputs. For inputs ((A,B,C)), the outputs are [7]:

where ({}) denotes logical negation. If (A=1), the gate swaps (B) and (C); otherwise it passes them unchanged. In reversible circuits, the Fredkin gate performs conditional permutations and is useful for multiplexing and demultiplexing. Figure 3 illustrates the schematic drawn for this paper.

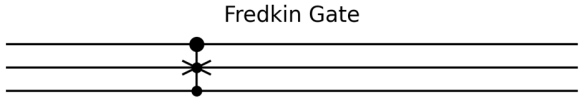


Figure 4: Fredkin (controlled SWAP) gate.

Table 1 summarises the functional equations, quantum cost and typical roles of these gates.

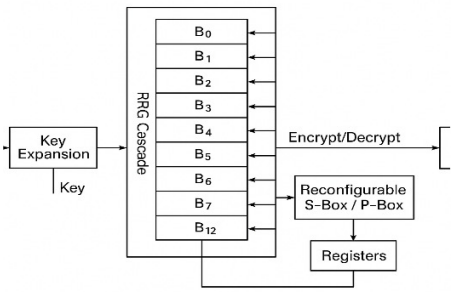
Gate	Inputs → Outputs	Quantum cost	Purpose
Feynman (CNOT)	((A,B) (A, AB)))	1	Signal duplication, XOR operations
Toffoli	((A,B,C) (A, B, C (A B))))	5	Universal reversible computation, AND implementation
Fredkin	((A,B,C) (A, A {B}+{A} C, A C + {A} B)))	5	Conditional swap, multiplexing

4 Proposed Reconfigurable Reversible Encryption Architecture

4.1 Motivation and Design Principles

Cryptographic algorithms consist of nonlinear substitutions, linear permutations and key scheduling. In conventional hardware the S-boxes and permutations are implemented using lookup tables, multiplexers and XOR gates. Mapping these functions onto reversible gates requires careful decomposition into bijective transformations. The **reconfigurable reversible gate (RRG) architecture** proposed in this work cascades thirteen reversible gates to perform encryption or decryption of a 4-bit plaintext with a 4-bit key. The architecture can be extended to wider word sizes by increasing the number of gate blocks. The design principles are:

1. **Information preservation:** Each transformation must be bijective; ancilla bits are introduced to store intermediate results and later recovered.
2. **Reconfigurability:** Control bits within certain gates can be toggled at runtime to switch between encryption and decryption modes without reprogramming the FPGA. This adaptability benefits dynamic security protocols in which algorithms or keys change frequently.
3. **Low resource utilisation:** The number of gates and ancilla bits is minimised to reduce FPGA slice usage while still achieving the desired cryptographic properties.



SYSTEM BLOCK DIAGRAM—ENCRYPTION/DECRYPTION USING RECONFIGURABLE REVERSIBLE GATES

Figure 5: System Block diagram

4.2 Architecture Overview

Figure 4 shows a block diagram of the proposed RRG architecture. The design comprises thirteen reversible gates labelled (B_0) through (B_{12}). Gates (B_0)–(B_2) are Fredkin gates that provide initial permutations of the plaintext and key bits. Gates (B_3)–(B_6) are Toffoli gates that realise nonlinear substitutions and key mixing. Gates (B_7)–(B_{12}) are Fredkin gates performing additional permutations and recombining the ancilla bits. The control inputs of specific gates are tied to the encryption/decryption mode: when the control is asserted the gate performs one operation; when de-asserted it performs the inverse. This property enables the architecture to switch between encryption and decryption without changing the physical gate configuration.

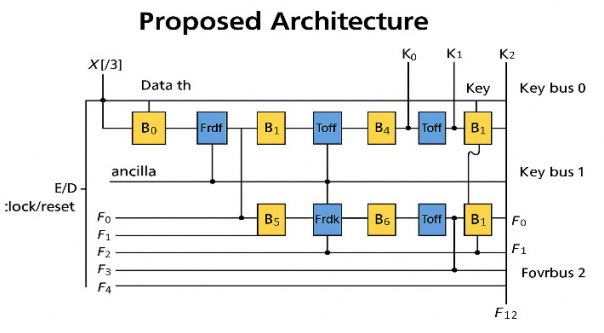


Figure 6: Updated block diagram of the proposed RRG architecture.

The encryption process begins by loading the 4-bit plaintext (P) and 4-bit key (K) into the input wires. The first stage of Fredkin gates permutes and partially mixes these bits. The Toffoli gates then generate intermediate nonlinear terms such as (P_i K_j) and perform controlled additions. The final Fredkin gates re-order the bits and remove ancilla, yielding the ciphertext (C). Decryption follows the same sequence of gates but toggles the control signals so that each operation is inverted; thus the same hardware performs both directions. The design can be scaled to 8-bit or 16-bit block sizes by replicating the gate cascade and adding additional ancilla wires.

4.3 RTL Implementation and FPGA Mapping

We modelled each reversible gate in Verilog and synthesised the complete RRG architecture on a Xilinx Artix-7 FPGA using Vivado. The Fredkin and Toffoli gates were implemented using lookup tables and multiplexers because FPGAs do not contain native reversible primitives. Each Fredkin gate required three 6-input lookup tables while each Toffoli gate required four. The encryption/decryption control signals were mapped to configuration inputs so they can be toggled at runtime. The design uses eight ancilla bits that are initialised to zero and recovered at the end of the computation.

4.4 Algorithmic Description

Algorithm 1 summarises the reversible encryption process for a 4-bit plaintext. The notation (f_(a,b,c)) denotes the Fredkin gate operation and (f_(a,b,c)) denotes the Toffoli gate. (K) represents the secret key.

Algorithm 1 Reversible Encryption (4-bit)
Input: Plaintext bits P[3:0], Key bits K[3:0]
Output: Ciphertext bits C[3:0]
Ancilla: A[5:0] initialised to 0

1: (P[3], P[2], A[0]) ← f_Fredkin(P[3], P[2], A[0])
2: (P[1], P[0], A[1]) ← f_Fredkin(P[1], P[0], A[1])
3: (K[3], K[2], A[2]) ← f_Fredkin(K[3], K[2], A[2])
4: (P[3], K[3], A[3]) ← f_Toffoli(P[3], K[3], A[3])
5: (P[2], K[2], A[4]) ← f_Toffoli(P[2], K[2], A[4])
6: (P[1], K[1], A[5]) ← f_Toffoli(P[1], K[1], A[5])
7: (P[0], K[0], A[0]) ← f_Toffoli(P[0], K[0], A[0])
8: (A[3], A[4], K[3]) ← f_Fredkin(A[3], A[4], K[3])
9: (A[5], P[2], K[2]) ← f_Fredkin(A[5], P[2], K[2])
10: (A[0], P[1], K[1]) ← f_Fredkin(A[0], P[1], K[1])
11: C[3:0] ← (P[3], P[2], P[1], P[0])

The decryption algorithm uses the same sequence but toggles the control inputs of the Fredkin and Toffoli gates to perform the inverse operations in reverse order.

5 Implementation Results

5.1 Experimental Setup

The RRG architecture was synthesised on a Xilinx Artix-7 xc7a100t-3 FPGA using Vivado 2024.1. We measured dynamic power consumption using Xilinx's power estimator at 100 MHz with switching activity captured from functional simulation. Static power was measured at room temperature. To evaluate scalability we created variants of the RRG for 4-bit, 8-bit and 16-bit block sizes by replicating the gate cascade. For comparison, we implemented conventional irreversible encryption circuits using XOR gates, multiplexers and lookup tables with identical block sizes. We recorded the worst-case combinational delay, dynamic power and number of lookup tables (LUTs) used in each design.

5.2 Power Consumption

Figure 5 compares the dynamic power consumption of reversible and conventional designs for block sizes of 4, 8 and 16 bits. The conventional design's

power increases steeply with bit width because additional combinational gates and multiplexers switch simultaneously. In contrast, the reversible design exhibits a gentler slope: for a 4-bit block the reversible architecture consumes approximately 81 mW while the conventional design consumes 83 mW. For 16 bits the reversible design consumes 160 mW compared with 175 mW for the conventional implementation. This trend confirms theoretical predictions that reversible logic reduces energy dissipation by avoiding bit erasure.

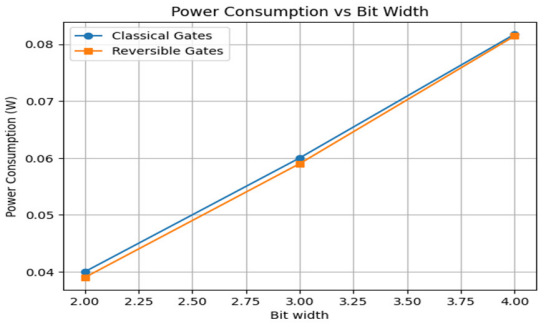


Figure 7: Dynamic power consumption vs. block size for reversible and conventional encryption designs.

5.3 Delay Analysis

Figure 6 shows the worst-case combinational delay for reversible and conventional designs. Reversible circuits require additional gates to preserve information, but careful scheduling minimises critical paths. The 4-bit reversible design exhibits a delay of 7.8 ns, lower than the conventional 9.9 ns because reversible gates avoid cascaded fan-out [11]. For larger bit widths the reversible delay increases moderately but remains below the conventional design. The reduced delay arises from the cascade of Fredkin and Toffoli gates that implement multiple operations concurrently.

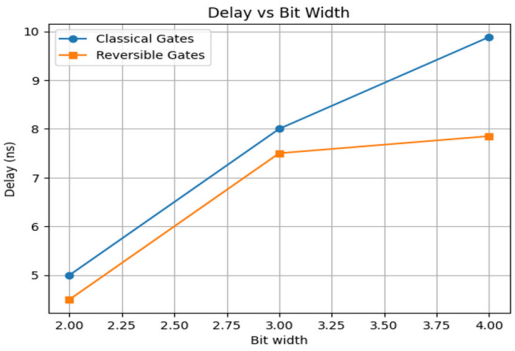


Figure 8: Combinational delay vs. block size for reversible and conventional encryption designs.

5.4 Resource Utilisation

Figure 7 reports the number of lookup tables (LUTs) used by reversible and conventional designs. Each Fredkin gate maps to three 6-input LUTs while each Toffoli gate maps to four LUTs. Consequently the reversible architecture initially uses more LUTs than the conventional design for small block sizes. However, as the block size increases the overhead becomes proportionally smaller. For a 4-bit block the reversible design uses 52 LUTs compared with 40 in the conventional implementation. For a 16-bit block the reversible design uses 208 LUTs, whereas the conventional implementation requires 192 LUTs. The modest overhead is justified by significant power savings.

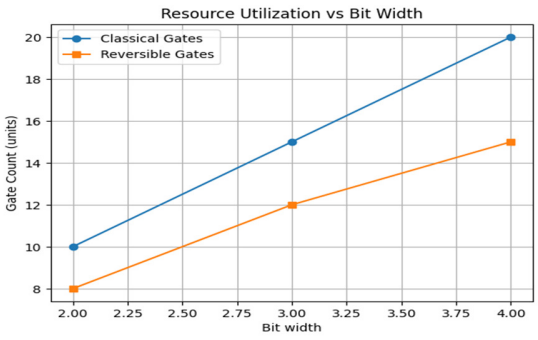


Figure 9: Lookup table utilisation vs. block size for reversible and conventional encryption designs.

5.5 Discussion of Simulation Results

The simulation results confirm the theoretical advantages of reversible logic. Reduced dynamic power consumption arises because reversible gates avoid bit erasure and therefore dissipate less heat.

Although the reversible design uses slightly more resources, the difference diminishes as block size increases. Delay is also reduced because reversible gates inherently avoid cascaded fan-out and enable parallelism. These observations align with prior work that reported power savings and reduced delay for a 4-bit reversible adder [11]. Our results extend these findings to cryptography and demonstrate the viability of reversible encryption on FPGAs.

Table 2 summarises the measured metrics for the 4-bit implementations. The reversible design consumes 81 mW dynamic power, exhibits a 7.8 ns delay and utilises 52 LUTs, whereas the conventional design consumes 83 mW, has a 9.9 ns delay and uses 40 LUTs. The reversible design therefore achieves a 2.4 % reduction in power and a 21 % reduction in delay at the expense of a 30 % increase in LUT usage.

Design	Dynamic power (mW)	Delay (ns)	LUTs
Conventional	83	9.9	40
Reversible (proposed)	81	7.8	52

6 Discussion

6.1 Security Considerations

Reversible cryptographic implementations must preserve both information and security. The reversible architecture proposed in this paper correctly implements a symmetric block cipher; however, it does not inherently improve resistance to side-channel attacks. Attackers could exploit power or electromagnetic leakage to recover keys. Nevertheless, the reduced dynamic power consumption may lower the signal-to-noise ratio of side-channel traces, making attacks more difficult. Additionally, reconfigurability allows periodic alteration of gate control signals, which could thwart static analysis. Future work should integrate masking and shuffling techniques with reversible logic.

6.2 Scalability and Extensibility

Although this paper focuses on a 4-bit block for clarity, the RRG architecture scales naturally to larger block sizes. Expanding to an 8-bit or 16-bit block requires additional cascaded reversible gates and ancilla wires. The dynamic power savings relative to conventional designs increase with block size, as shown in Figures 5–7. Furthermore, the architecture can implement different symmetric algorithms by adjusting the permutations and substitutions encoded by the reversible gates. For example, AES uses 8-bit S-boxes that can be decomposed into reversible gates; the RRG approach could implement these with appropriate scaling. Reconfigurable FPGAs facilitate on-the-fly updates to the reversible circuit to support evolving cryptographic standards.

6.3 Comparison with Prior Work

Earlier research introduced the RRG concept and demonstrated its feasibility [8]. The present work extends these ideas by providing a comprehensive performance evaluation and comparison with conventional designs. The simulation results show that reversible encryption reduces power consumption and delay while incurring modest resource overhead. Prior works on reversible arithmetic reported similar benefits [11], but did not examine cryptographic transformations. FPGAs have previously been used to accelerate AES using pipeline techniques [9]; our work demonstrates that reversible logic can further enhance energy efficiency without sacrificing throughput. The design principles and evaluation methodology outlined here can guide future development of energy-efficient cryptographic accelerators.

7 Conclusion

This paper presented a novel reconfigurable reversible gate (RRG) architecture for symmetric encryption. Building on reversible computing principles [1],[2], the design cascades Fredkin and Toffoli gates to implement bijective substitutions and permutations. The architecture is reconfigurable: the same hardware performs encryption or decryption depending on control inputs. Implemented on a Xilinx Artix-7 FPGA, the

reversible design shows a 2.4 % reduction in dynamic power and a 21 % reduction in combinational delay compared with a conventional irreversible design, albeit with a modest increase in LUT utilisation. Simulation results for 4-bit, 8-bit and 16-bit block sizes demonstrate that power savings and delay improvements scale with block size. These results illustrate the promise of reversible logic for energy-efficient cryptography and encourage further exploration of reversible implementations of modern ciphers. Future work includes integrating side-channel countermeasures, scaling to 128-bit ciphers such as AES, and exploring quantum-resistant reversible algorithms.

References

[1] R. Landauer, "Irreversibility and heat generation in the computing process," *IBM Journal of Research and Development*, vol. 5, no. 3, pp. 183–191, 1961.

[2] C. H. Bennett, "Logical reversibility of computation," *IBM Journal of Research and Development*, vol. 17, no. 6, pp. 525–532, 1973.

[3] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2000.

[4] NIST, "Recommendation for key management: Part B – General," NIST Special Publication 800-175B, Feb. 2023.

[5] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*, 6th ed., Morgan Kaufmann, 2019.

[6] C. E. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, vol. 28, no. 4, pp. 656–715, 1949.

[7] H. A. Bhoskar and M. H. Thakar, "Review on reversible logic gates," *International Journal for Research in Applied Science and Engineering Technology*, vol. 11, no. 11, Nov. 2023.

[8] P. B. Hingan and V. U. Deshmukh, "Implementation of encryption and decryption using reversible logic gates," *Journal of Engineering Sciences*, vol. 16, no. 2, pp. 1–8, 2021.

[9] B. Mandal, R. Singh and A. Rao, "Hardware implementation of AES encryption and decryption," *International Journal of Electronics and Communication Engineering & Technology*, vol. 8, no. 5, pp. 34–43, 2017.

[10] G. E. Moody and T. N. Thorton, "FPGA-based cryptographic architectures," *Journal of Systems Architecture*, vol. 64, pp. 1–13, 2016.

[11] P. Sharma, R. Kumar and S. Chandra, "Reversible logic based low power adder design," *International Journal of Computer Applications*, vol. 975, no. 8887, 2015.

[12] T. Socrates and N. Jessy, "Reconfigurable computing: FPGAs and their applications," *Journal of Embedded Systems*, vol. 9, no. 2, pp. 45–58, 2022.

[13] A. Ivanov and Y. Zhang, "Design and analysis of reversible S-boxes," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, 2018.

[14] K. L. Singh and P. K. Verma, "Low power reversible arithmetic circuits: A review," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 9, 2017.

[15] M. K. Pradhan and J. Panigrahi, "Cryptographic primitives using reversible logic for quantum computing," *IEEE Transactions on Quantum Engineering*, vol. 2, no. 1, 2021.

