

# FPGA IMPLEMENTATION OF OPTIMIZED AES ALGORITHM FOR MULTIMEDIA MESSAGES

A DISSERTATION  
SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE AWARD OF DEGREE  
OF  
MASTER OF TECHNOLOGY  
IN  
SIGNAL PROCESSING AND DIGITAL DESIGN

Submitted by:

**JAIDEEP KALA**  
**2K21/SPD/04**

Under the supervision of:

**PROF. JEEBANANDA PANDA & Ms. LAVI TANWAR**



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION  
ENGINEERING**

**DELHI TECHNOLOGICAL UNIVERSITY**

(Formerly Delhi College of Engineering)

Bawana Road, New Delhi – 110042

MAY 2023

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, New Delhi – 110042

**CANDIDATE’S DECLARATION**

I, Jaideep Kala, Roll No. 2K21/SPD/04, student of MTech (Signal Processing and Digital Design), hereby declare that the project dissertation titled “FPGA Implementation Of Optimized AES Algorithm For Multimedia Messages” which is submitted by me to the Department of Electronics and Communication Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirements for the award of degree of Master of Technology in Signal Processing and Digital Design, is original and not copied from any source without citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi

JAIDEEP KALA

Date: 29 May 2023

(2K21/SPD/04)

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, New Delhi – 110042

**CERTIFICATE**

I hereby certify that the project dissertation titled “FPGA Implementation of Optimized AES Algorithm for Multimedia Messages”, which is submitted by Jaideep Kala, Roll No. 2K21/SPD/04, Department of Electronics and Communication Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirements for the award of degree of Master of Technology in Signal Processing and Digital Design is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this university or elsewhere.

Place: Delhi

Date: 29 May 2023

JEEBANANDA PANDA

SUPERVISOR

PROFESSOR

(Department of Electronics and  
Communication Engineering)Delhi Technological  
University

LAVI TANWAR

SUPERVISOR

ASST. PROFESSOR

(Department of Electronics and  
Communication Engineering)Delhi Technological  
University

## ABSTRACT

Ensuring secure communication of multi-media messages is crucial for social networking and data sharing platforms. Prevention of data manipulation and theft has led to the development of various encryption techniques, but scope remains for a fast and efficient multi-media encryptor. Advanced Encryption Standard (AES) is mathematically one of the most complex cipher algorithms to crack and has been widely deployed in the banking sector.

The algorithm's mathematical framework and the implementation of numerous iterations of encryption procedures augment its security. AES has undergone exhaustive examination and scrutiny by the cryptographic community, unveiling no significant vulnerabilities. AES implementation for battery operated devices requires an algorithm with low power consumption and high-speed encryption/decryption of digital data. This dissertation proposes an FPGA implementation of a high throughput parallel pipelined 128-bit AES algorithm with a low power key expansion mechanism for iterative stages. A 128-bit symmetric key has been used for undertaking 10 rounds of transformations. All the encryption and decryption transformations are simulated using iterative design methodology in order to minimize hardware consumption.

Xilinx Artix-7 FPGA device is used for hardware evaluation and Verilog HDL for programming. Simulation and synthesis task has been performed on Xilinx Vivado v2021.1 IDE.

The results exhibit high-rate encryption of 68 Gb/s and low energy consumption of 7 pJ/bit. Detailed study of the synthesized design has been undertaken to highlight the power consumption and performance of the algorithm at various operating voltages and temperature levels. The results have further been compared with existing work to substantiate its unwavering dependability and formidable efficacy. AES is integrated into a multitude of applications and systems, encompassing secure communication protocols, virtual private networks (VPNs), disk encryption software, and various other domains.

A high throughput implementation of 256-bit AES cipher has also been carried out for encrypting digital images and explore its practicality in peer-to-peer communication. Pre-processing of images has been performed to make them suitable for encryption. A detailed study of the encryption results and histogram analysis has been carried out. The proposed algorithm achieved a Peak Signal to Noise Ratio (PSNR) of 61 dB for the decrypted image. Correlation between the input and the decrypted image was found to be 0.994 while the Mean Square Error (MSE) was calculated to be 0.0030. AES-256 has gained wide acceptance and standardization, making it compatible across different platforms, domains, operating systems, and devices.

This compatibility facilitates interoperability and seamless integration into multimedia systems including video streaming, image protection, digital rights management, and secure multimedia communication.

## **ACKNOWLEDGEMENT**

I, Jaideep Kala, Roll No. 2K21/SPD/04, student of MTech (Signal Processing and Digital Design), hereby thank and express my sincere gratitude to my supervisors, Prof. Jeebananda Panda and Ms. Lavi Tanwar, with whose continuous support and insightful guidance, this project titled “FPGA Implementation of Optimized AES Algorithm for Multimedia Messages” was successfully undertaken by me.

Place: Delhi

Date: 29 May 2023

**JAIDEEP KALA**

(2K21/SPD/04)

## CONTENTS

CANDIDATE’S DECLARATION .....	ii
CERTIFICATE .....	iii
ABSTRACT.....	iv
ACKNOWLEDGEMENT .....	vi
CONTENTS .....	vii
LIST OF TABLES .....	ix
LIST OF FIGURES .....	x
LIST OF SYMBOLS, ABBREVIATIONS AND NOMENCLATURE .....	xi
CHAPTER 1 .....	12
1.1.    NEED FOR ENCRYPTION .....	12
1.2.    HISTORY OF ENCRYPTION.....	13
1.3.    ADVANCED ENCRYPTION STANDARD .....	14
1.4.    ADVANCEMENT IN ENCRYPTION.....	15
1.5.    GENERAL STRUCTURE OF ENCRYPTION ALGORIHTM.....	17
1.6.    STRENGTH OF ENCRYPTION.....	19
1.7.    POWER AND PERFORMANCE OF ENCRYPTION TECHNIQUES .....	20
1.8.    IMPLEMENTATION OF ENCRYPTION TECHNIQUES .....	22
CHAPTER 2 .....	27
2.1.    LITERATURE REVIEW.....	27
CHAPTER 3 .....	31

3.1. SHANNON’S THEORY OF CONFUSION AND DIFFUSION .....	31
3.1.1 CONFUSION .....	31
3.1.2 DIFFUSION .....	31
3.2. FRAMEWORK OF ENCRYPTION ALGORITHM .....	32
CHAPTER 4 .....	34
4.1. ALPHANUMERIC DATA CIPHER .....	34
4.1.1. KEY EXPANSION .....	35
4.1.2. SUBSTITUTE BYTES.....	38
4.1.3. SHIFT ROWS.....	38
4.1.4. MIX COLUMNS.....	39
4.1.5. ADD ROUND KEYS.....	39
4.1.6. PIPELINED ARCHITECTURE .....	40
4.2. MULTIMEDIA DATA CIPHER .....	40
4.2.1. IMAGE ENCRYPTION USING AES-256 .....	40
CHAPTER 5 .....	42
5.1. FPGA SYNTHESIS RESULTS FOR ALPHANUMERIC TEXT ENCRYPTIONE.....	42
5.2. SIMULATION RESULTS FOR IMAGE ENCRYPTION.....	45
CHAPTER 6 .....	48
6.1. CONCLUSIONS AND FUTURE SCOPE .....	48
REFERENCES .....	51
APPENDIX A (LIST OF PUBLICATIONS) .....	61
APPENDIX B (PLAGIARISM REPORT) .....	66



## LIST OF TABLES

Table 4.1.1 Round constant table for 10 rounds of transformation in AES.....	37
Table 5.1.1. Energy/bit variation with temperature and input voltage .....	44
Table 5.1.2. Input voltage vs dynamic and leakage power .....	44
Table 5.1.3. Result comparison of presented method with existing research. ....	45

## LIST OF FIGURES

Figure 4.1: Block diagram of N rounds 128-bit AES algorithm.....	35
Figure 4.2: Key Expansion process for 128-bit symmetric key. ....	36
Figure 4.3: Key expansion operation for obtaining next 4 key words.....	36
Figure 4.4: XOR of expanded key with each round of AES. ....	37
Figure 4.5: Substitution byte transformation using 16x16 S-box .....	38
Figure 4.6: Shift rows operation on 128-bit block size data.....	39
Figure 4.7: Multiplication of state matrix with constant matrix.....	39
Figure 4.8: Flow chart for encryption process of RGB Image.....	41
Figure 5.1: Encrypted 128-bit hexadecimal output. ....	42
Figure 5.2: Shows the Encrypted 128-bit hexadecimal results and decrypted message. ....	42
Figure 5.3: Simulation runtime and status of encryption. ....	43
Figure 5.4: AES top module with input data, key and clock pulse and encrypted output. ...	43
Figure 5.5 Round transformation module with its input and output vectors .....	44
Figure 5.6: RGB Image and Gray scale image with their Histogram plots. ....	46
Figure 5.7: Input image matrix to the AES-256.....	47
Figure 5.8: 4×4 encrypted image matrix .....	47
Figure 5.9: Encrypted & Decrypted images with their Histogram plot.....	47

## **LIST OF SYMBOLS, ABBREVIATIONS AND NOMENCLATURE**

1. **AES** – Advanced Encryption Standard
2. **DES** – Data Encryption Standard
3. **FPGA** – Field Programmable Gate Array
4. **HDL** – Hardware Description Language
5. **ECC** – Elliptic Curve Cryptography
6. **ECB** – Electronic Code Book
7. **IV** – Initialization Vector
8. **CBC** – Cipher Block Chaining
9. **QKD** – Quantum Key Distribution
10. **MPC** – Multi-Party Computation
11. **LUT** – Look Up Table
12. **FHE** – Fully homomorphic encryption
13. **SSL** – Secure Sockets Layer
14. **MMSE** – Minimum Mean Square Error
15. **TLS** – Transport Layer Security
16. **IoT** – Internet of Things
17. **S-Box** – Substitution Box
18. **MSE** – Mean Square Error
19. **PSNR** – Peak Signal-to-Noise Ratio
20. **SNR** – Signal-to-Noise Ratio

## CHAPTER 1

### INTRODUCTION

#### 1.1. NEED FOR ENCRYPTION

Encryption ensures sensitive information remains confidential by converting it into an unreadable form. Only authorized individuals possessing the decryption key can decode the encrypted data, preventing unauthorized access. It safeguards data against unauthorized modification, tampering, or corruption. Even if intercepted, encrypted data cannot be altered without the decryption key. This guarantees the integrity and authenticity of the information [1]. In an increasingly digital world, safeguarding privacy has become crucial. Encryption enables secure and private communication, ensuring that personal conversations, emails, financial transactions, and other sensitive information remain confidential. Numerous industries, including finance, healthcare, and government, have strict regulations pertaining to sensitive data protection. Encryption often serves as a requirement to comply with these regulations, helping organizations meet legal obligations and avoid penalties or legal consequences [2].

Encryption is vital for secure communication across networks. It thwarts unauthorized parties from eavesdropping on conversations, intercepting messages, or accessing sensitive data in transit. This is particularly important for activities such as online banking, e-commerce, and confidential business communications [3]. Governments and intelligence agencies rely on encryption to protect sensitive information, secure communications among officials, and safeguard national security interests. Encryption helps prevent unauthorized access to classified or confidential data, mitigating potential threats [4]. Overall, encryption techniques are vital for safeguarding data, preserving

privacy, ensuring compliance, and establishing trust in the current digital landscape. They form the foundation for secure communication, protect sensitive information, and support various aspects of our personal and professional lives [5].

## **1.2. HISTORY OF ENCRYPTION**

In the early 1970s, DES was developed by the U.S. National Bureau of Standards (now NIST) as the first widely used encryption standard [6]. DES utilized a symmetric key algorithm, where the same key was employed for both encryption and decryption [7]. With a key length of 56 bits, DES gained widespread adoption in various applications. Public key cryptography emerged in the late 1970s, independently introduced by Whitfield Diffie and Martin Hellman, and subsequently enhanced by Rivest, Shamir, and Adleman (RSA) [8]. Public key cryptography involves a pair of mathematically similar keys: a public key for encrypting and a private key for decrypting. This breakthrough enabled secure communication without requiring a shared secret key [9]. SSL, developed by Netscape in the mid-1990s, and its successor TLS, are cryptographic protocols deployed for securing internet communication. They provide encryption and authentication mechanisms, ensuring secure connections between web browsers and servers. SSL and TLS have undergone multiple iterations and updates to address vulnerabilities and enhance security. Elliptic Curve Cryptography (ECC) is a contemporary technique of public key cipher based on the algebraic structure of elliptic curves over finite fields [10]. ECC offers nearly equal security as RSA but with shorter key lengths, resulting in computational efficiency. ECC has gained popularity, particularly in resource-constrained environments like mobile devices and IoT devices. These significant milestones represent notable advancements in the history of digital encryption, addressing evolving security needs and technological progress. Encryption remains integral to protecting sensitive information in the digital age [11].

### 1.3. ADVANCED ENCRYPTION STANDARD

Cyber-attacks and data theft are a major concern for companies worldwide. Securing messages containing digital images and videos for real time communication can be a difficult and computationally expensive task. AES has been widely used for secure storage and transmission of digital information but its application in real time image/video encryption has been limited [12]. Developed in the year 2001 by National Institute of Standards and Technology, AES is a cryptographic algorithm that uses a symmetric public key for undertaking encryption and decryption tasks [13]. It is robust to existing brute force attacks and has been widely used by financial institutions and government agencies to carry out secure transactions and sharing of sensitive information.

AES processes 128-bit packets of data, although it can have a key size of 128,192 or 256 bits [14]. AES-256 is the most secure form of AES encryption and consists of 14 rounds of iterations for manipulating data into an unrecognizable form [15]. Encryption of multimedia messages using existing cipher technologies requires high bandwidth and large latencies are observed in peer-to-peer communication [16]. A high throughput encryption technique coupled with modules for pre, and post processing of images is needed for solving this issue and have a vast application in social networking / message sharing apps. AES can operate in 5 different modes, most common among these are Cipher Block Chaining, Electronic Code Book [17]. ECB is a widely used mode as it does not require an Initialization Vector (IV) for its operations. CBC requires IV which may lead to propagation of error through the encryption stages [18]. Encryption techniques like AES are based on Shannon's theory of confusion and diffusion (1945) [19]. Here confusion aims at complicating the relationship between cipher message and symmetric key whereas diffusion aims at dispersing the features of input message throughout the encrypted message. It can be efficiently used for both hardware and software applications One of the major limitations of AES algorithm is its

high computational complexity which leads to high power consumption, making it less suitable for battery operated devices. Another area for improvement in AES is the speed of the algorithm [20].

Faster encryption and decryption processes are highly desirable for real time communication and data storage. For hardware implementation, AES algorithms that takes less area for implementation, has high throughput and low power consumption are preferable and is a topic of constant research [21].

#### **1.4. ADVANCEMENTS IN ENCRYPTION**

Among the various underdevelopment encryption techniques, few of the most promising and revolutionary ones are:

1. Quantum key distribution (QKD) is an advanced and groundbreaking encryption technique based on the intricate principles of quantum mechanics [22]. It harnesses the remarkable properties of quantum particles, such as photons, to establish highly secure communication channels. Quantum encryption offers unparalleled security guarantees, as any interception attempt disrupts the delicate quantum state, triggering immediate detection. It presents a promising solution to the looming security challenges posed by quantum computers to conventional encryption algorithms [23].
2. Homomorphic Encryption: It is a powerful and innovative encryption method that empowers computations to be performed on encrypted information without the need for decryption [24]. It enables the processing of critical information while preserving utmost privacy and security. Homomorphic encryption finds diverse applications in secure cloud computing, privacy-preserving data analysis, and collaborative computations, revolutionizing the way data is securely utilized and analyzed [25].

3. **Lattice-Based Cryptography:** It represents a sophisticated form of post-quantum cryptography that capitalizes based on the formidable complexity of mathematical problems related to lattice structures [26]. It offers robust resistance against attacks from both classical and quantum computers, making it a highly reliable and future-proof cryptographic approach. Lattice-based cryptographic schemes provide stringent security guarantees and have emerged as a prominent and rapidly evolving research area in the era of post-quantum cryptography [27].
4. **Zero-Knowledge Proofs:** Zero-knowledge proofs stand as an ingenious and privacy-enhancing cryptographic protocol, allowing one party, known as the prover, to convincingly demonstrate knowledge of a statement to another party, known as the verifier, without divulging any additional information [28]. Zero-knowledge proofs have profound implications for privacy-preserving protocols and secure authentication systems. By ensuring that sensitive information remains undisclosed during authentication or verification processes, they enable individuals and organizations to maintain the highest levels of privacy and security [29].
5. **Multi-Party Computation:** MPC protocols embody a sophisticated and secure collaborative computing paradigm, empowering multiple parties to collectively compute a function on their respective private inputs while preserving the confidentiality of individual inputs through encryption [30]. MPC facilitates secure collaboration and computation, thereby enabling privacy-preserving data analysis and secure outsourcing of computations [31]. It represents a groundbreaking advancement in secure and privacy-enhanced data processing.
6. **Fully Homomorphic Encryption:** FHE exemplifies a state-of-the-art and groundbreaking encryption technique that enables the execution of arbitrary computations on encrypted information without the requirement for decryption [32]. FHE empowers privacy-preserving data processing in scenarios where maintaining the strictest levels of data confidentiality is of paramount importance.



Although FHE is an active and rapidly evolving research domain, it holds immense promise for the future of secure cloud computing and data privacy, revolutionizing the way sensitive data is processed and utilized [33].

These remarkable and scientific advancements in encryption techniques epitomize ongoing research and development efforts, offering unparalleled security, privacy, and functionality [34]. They effectively address the dynamic challenges posed by data protection and secure communication in the ever-evolving digital age.

### **1.5. GENERAL STRUCTURE OF ENCRYPTION ALGORITHM**

The essential elements of an encryption algorithm comprise various key components. These encompass substitution, transposition, key generation, encryption function, decryption function, and key management.

Substitution involves the replacement of plaintext elements with alternative elements or values, which can be achieved through the use of substitution tables or mathematical functions. Transposition refers to the rearrangement of the sequence of plaintext elements to generate ciphertext, employing techniques such as permutation or reordering based on specific patterns [35].

Key generation plays a pivotal role in encryption algorithms, as it involves the creation of suitable keys with appropriate length and complexity. These keys govern the conversion of plaintext into ciphertext and vice versa. They can be derived from user input, passphrases, or cryptographic protocols [36].

The encryption function is responsible for the actual transformation of plaintext into ciphertext using the provided key. This function combines substitution and transposition operations, employing mathematical operations on the plaintext and key to produce the encrypted output.

The decryption function functions as the inverse of the encryption function. It takes the ciphertext and the corresponding decryption key, reversing the encryption process and converting the ciphertext back into plaintext [37].

Key management is a critical aspect of encryption algorithms, encompassing secure key generation, distribution, storage, and disposal. It ensures the confidentiality and safeguarding of keys from unauthorized access, as well as their appropriate synchronization between communicating parties [38-39].

Regarding the structure of encryption algorithms, many modern ones, such as the widely used Advanced Encryption Standard (AES), adhere to a prevalent structure known as a Feistel structure. This structure involves multiple rounds of processing, with each round employing a fusion of substitution and transposition operations on the input data [40].

In a typical Feistel structure, the plaintext is divided into two equivalent parts. These segments undergo multiple rounds of processing, involving the following steps:

1. The right segment of the data undergoes a substitution operation using a specific function that takes the right segment and a round key as inputs. The substitution function employed may vary depending on the algorithm [41].
2. The outcome of the substitution is combined with the left segment of the data using an operation such as XOR.
3. The left and right segments are interchanged, with the previous right segment becoming the new left segment.
4. These steps are repeated for a predefined number of rounds, typically 10, 12, or 14, depending on the algorithm and key length [42-44].
5. In the final round, the left and right segments are swapped again, but no further processing is applied.

The ultimate output of the last round represents the ciphertext, the altered and encrypted version of the plaintext.

It is crucial to note that while the Feistel structure is a prevalent approach, the specific constituents and structure of an encryption algorithm may vary based on its design and cryptographic properties [45]. Different algorithms may employ diverse methodologies to achieve encryption, but they generally encompass the a forementioned key components.

## 1.6. STRENGTH OF ENCRYPTION

Encryption algorithms are specifically crafted to offer robust security and safeguard data from unauthorized access. The strength of an encryption algorithm refers to its resilience against diverse attacks and the arduousness of decrypting the encryption without possessing the correct decryption key.

The process of attempting to decrypt encrypted data and uncover the original plaintext without the key is commonly referred to as "cracking" or "cryptanalysis." Cracking encryption algorithms often involves employing a variety of methods, such as brute-force attacks, exploiting cryptographic vulnerabilities, and executing side-channel attacks. Nevertheless, contemporary encryption algorithms are intentionally designed to withstand such attacks, making their successful cracking exceedingly challenging and time-consuming [46].

1. Brute-Force Attacks: Brute-force attacks entail systematically attempting every conceivable key combination until the correct one is identified. The strength of an encryption algorithm is intricately tied to the length of the encryption key [47]. Lengthier key sizes exponentially expand the number of possible key combinations, rendering brute-force attacks impractical. For instance, a 128-bit AES key boasts an overwhelmingly vast number of  $2^{128}$  possible combinations.

2. Cryptographic Weaknesses: Cryptographic weaknesses may arise from algorithmic design or implementation flaws. These weaknesses can be exploited to uncover vulnerabilities that can subsequently be leveraged to crack the encryption. However, modern encryption algorithms, such as AES, have undergone meticulous scrutiny and evaluation by cryptographic experts to ensure their robustness and resilience against known weaknesses [48].

3. Side-Channel Attacks: Side-channel attacks capitalize on information leaked during the execution of an encryption algorithm, such as timing data, power consumption, electromagnetic emissions, or even acoustic emanations. By scrutinizing these side-channel signals, an attacker may gather information about the encryption key. Guarding against side-channel attacks necessitates implementing countermeasures, such as constant-time algorithms, to minimize the leakage of sensitive information [49].

It is essential to acknowledge that while encryption algorithms themselves may exhibit strength, vulnerabilities may potentially emerge in their implementation or the surrounding systems. These vulnerabilities may encompass weak key management practices, improper encryption utilization, or other security weaknesses that can undermine the encryption's effectiveness [50].

To ensure the security of encrypted data, it is of paramount importance to adhere to recommended best practices. These may entail employing encryption algorithms with sufficiently long and robust key sizes, implementing sound key management and secure storage practices, regularly updating software and systems with security patches, and employing robust authentication mechanisms. In summary, contemporary encryption algorithms are thoughtfully crafted to offer formidable security and resilience against cracking attempts [51]. The strength of encryption resides in the vast key spaces, absence of known vulnerabilities, and the ability to withstand diverse attack vectors. Nonetheless, it is vital to adopt best practices and remain vigilant to uphold the security of encrypted data.

### **1.7. POWER AND PERFORMANCE OF ENCRYPTION TECHNIQUES**

The power efficiency and throughput of encryption techniques are pivotal aspects when assessing their performance and effectiveness.

1. Power Efficiency: Power efficiency relates to the energy consumption of an encryption technique during its operation. It is crucial to minimize power usage, particularly in

limited resource environments such as mobiles, IoT devices, or battery-powered systems. Power-efficient encryption techniques are engineered to curtail energy consumption while upholding the security of the encryption process [52]. By reducing power consumption, devices can operate for extended periods on limited power sources, thus prolonging their battery life.

2. **Throughput:** Throughput, in the context of encryption techniques, refers to the rate at which data can be processed and encrypted. It measures the efficiency and speed of the encryption process [53]. Higher throughput translates to expedited encryption and decryption operations, facilitating efficient data transmission and processing. Throughput assumes paramount importance in scenarios involving copious amounts of data, real-time applications, or high-performance computing environments. Encryption techniques boasting high throughput ensure that the encryption process does not impede data processing or communication systems.

Attaining a harmonious equilibrium between power efficiency and throughput is critical when selecting an appropriate encryption technique for a given use case. Different encryption algorithms and implementations exhibit varying power consumption and throughput characteristics [54].

In practice, the power and throughput of encryption techniques can be influenced by several factors, including:

- **Algorithm Complexity:** The computational complexity of the encryption algorithm has a direct impact on power consumption and throughput. Elaborate algorithms typically necessitate more computational resources and might incur higher power usage, resulting in diminished throughput. Striking a balance between algorithm complexity and security requirements is essential to achieve desired performance attributes [55].
- **Hardware Acceleration:** Hardware acceleration techniques, such as specialized cryptographic processors or dedicated hardware modules, can significantly enhance the power efficiency and throughput of encryption. These hardware components are

optimized to expedite encryption operations, mitigating power consumption, and bolstering processing speed [56].

- **Implementation Efficiency:** The efficiency of the encryption implementation, encompassing software optimizations, parallelization methodologies, or algorithmic enhancements, can impact power consumption and throughput. Well-optimized implementations reduce computational overheads and enhance overall efficiency.
- **Key Size:** The length of encryption keys can affect both power consumption and throughput. Lengthier key sizes may necessitate additional computational resources, resulting in heightened power consumption and potentially reduced throughput [57]. Optimal selection of key sizes that strike a balance between security requirements and performance considerations is paramount.

Considering power efficiency and throughput requisites while selecting an encryption technique is crucial, factoring in the specific constraints and objectives of the target system or application. Assessing the power consumption and throughput characteristics of encryption techniques ensures an optimal equilibrium between security, performance, and energy efficiency.

## **1.8. IMPLEMENTATION OF ENCRYPTION TECHNIQUES**

The utilization of encryption techniques can be divided into two main approaches: hardware implementation and software implementation.

### **1. Hardware Implementation:**

Hardware implementation involves the utilization of specialized hardware components or dedicated cryptographic processors to execute encryption and decryption operations. These hardware solutions are specifically designed for cryptographic tasks and offer several advantages:

- **Swiftness and Efficiency:** Hardware implementations are generally faster and more efficient compared to software implementations. Dedicated cryptographic hardware can swiftly execute encryption algorithms, reducing processing time and enhancing overall system efficiency [58].
  
- **Parallelism:** Hardware solutions can take advantage of parallel processing capabilities to simultaneously carry out multiple encryption or decryption operations. This parallelism increases throughput and enables the efficient handling of large volumes of data [59].
  
- **Security:** Hardware implementations provide heightened security by isolating cryptographic operations from other system processes. Dedicated cryptographic modules or secure hardware elements ensure the safeguarding of sensitive cryptographic keys and prevent unauthorized access or tampering.
  
- **Resilience to Attacks:** Hardware solutions can be designed with specific security measures to withstand various types of attacks, such as side-channel attacks or physical tampering. These measures strengthen the security of cryptographic keys and protect against potential vulnerabilities [60].

However, hardware implementation also has certain limitations:

- **Flexibility and Upgradability:** Hardware solutions are often tailored to specific encryption algorithms or protocols. Modifying or upgrading the implemented encryption scheme may require physical alterations to the hardware, limiting flexibility and adaptability.
  
- **Cost and Complexity:** Hardware implementation can entail additional costs, as it may require specialized hardware components or dedicated cryptographic processors. The development and integration of custom hardware solutions can also be intricate and time-consuming [61].

## 2. Software Implementation:

Software implementation involves the utilization of software algorithms and libraries to execute encryption and decryption operations on general-purpose computing platforms.

Software implementations offer the following advantages:

- Flexibility and Compatibility: Software encryption algorithms can be easily updated or replaced without the need for hardware modifications. They can be implemented across a wide range of computing platforms, from desktop computers to mobile devices.
- Cost-Effectiveness: Software implementations typically do not require additional hardware components, making them cost-effective compared to hardware solutions.
- Adaptability: Software encryption allows for the support of various encryption algorithms, enabling flexibility in selecting the most suitable algorithm for specific security requirements or regulatory standards.

However, software implementation also has certain limitations:

- Performance: Software implementations are generally slower compared to hardware implementations. The execution of encryption algorithms relies on the computational capabilities of the underlying hardware, which may not be as optimized for cryptographic operations as dedicated hardware solutions.
- Security Risks: Software implementations may be more susceptible to specific types of attacks, such as side-channel attacks or software vulnerabilities. Implementing additional security measures, such as secure coding practices and robust key management, is crucial to mitigate these risks.

In practice, a combination of hardware and software implementation is often employed to strike a balance between performance, security, and flexibility. Dedicated hardware



components can accelerate computationally intensive encryption tasks, while software implementations provide adaptability and versatility across various computing platform.

Encryption plays a pivotal role in the realms of AI and robotics, ensuring data security and privacy. It serves as a vital safeguard, protecting sensitive information generated and processed by AI and robotics systems, shielding it from unauthorized access, interception, and tampering. Encryption techniques are employed to fortify data throughout storage, transmission, and processing stages, effectively mitigating the perils of data breaches. Moreover, encryption enables secure collaboration by safeguarding the exchange of data and models among researchers and organizations. It also upholds privacy by encrypting personal data employed in AI applications, such as facial recognition or user behaviour analysis.

Furthermore, encryption acts as a protective shield for AI models and intellectual property, thwarting unauthorized access and replication, thus upholding the sanctity of proprietary technologies. In the deployment of AI and robotics systems, encryption ensures secure communication among devices, sensors, and actuators, effectively fending off malicious interference. Overall, encryption in AI and robotics augments data security, privacy, collaboration, and engenders trust in these cutting-edge technologies [62].

Military forces and aerial vehicles employ advanced encryption techniques to secure their radio communications.

These methods include Frequency Hopping Spread Spectrum (FHSS) and Advanced Encryption Standard (AES), which provide robust security and prevent interception or decoding of signals [63]. Secure Voice Systems encrypt voice signals, while spread spectrum techniques spread signals across a wide frequency range to deter interception and jamming. Tactical Secure Radios support encryption algorithms like AES, DES, or 3DES, along with secure key management and distribution protocols. Additionally, link encryption ensures data confidentiality during transmission. Advancements in encryption techniques aim to strengthen algorithms, improve key management, and integrate encryption into next-

generation systems, ensuring secure and reliable communication channels for military operations and sensitive information [64]. .

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1. RELATED WORK

There have been significant efforts in the past aimed at utilizing AES algorithm for encrypting alphanumeric messages, but very limited work has been carried out for multimedia messages. Hajihassani et al. [65] proposed a high throughput AES architecture based on bit slicing, which processed 32 128-bit data stream parallelly to provide an encryption throughput of 1.47 Tbps. This method performs expensive Byte transformations using shifting and swapping of registers.

Kim et al. [66] utilized the CTR mode in AES for fast encryption in low-end microcontrollers. This led to a reduced computation of 2 Add Round Key, 2 Shift-Rows, 2 Substitution Bytes, and 1 Mixed Column transformation. Alomari et al. [67] compared different encryption techniques for storage devices for different operating modes. A detailed performance analysis was done to provide guidelines for disk encryption.

Le et al. [68] proposed a fast GPU parallel computing design for AES cryptography. This technique accelerates the speed of AES encryption significantly. A novel low latency FPGA based AES architecture was proposed by Zhang et al. [69] in which an efficient key expansion method along with pipelining was used to obtain a throughput of nearly 21.56 Gbps.

Custom S-box techniques are implemented for low energy consumption applications for AES designed for battery operated devices [70-71]. Pipelined AES

architectures proposed by Chellappa et al. [72] and Oukili et al. [73] provided high throughput encryption of up to 64-79 Gbps.

For encryption of images, Zhang et al. [74] performed MATLAB implementation of AES-128 followed by digital image processing to obtain the encrypted and recovered test image. AES in CBC mode was utilized for image encryption [75] and its security performance was tested and compared with existing systems based on chaos.

A similar system was proposed by Arab et al. [76] for novel image encryption using chaos sequence and an improvised AES-128. This approach reduces computational time and increases the diffusion ability of the proposed scheme. Singh et al. [77] proposed a dynamic AES developed using key dependent S-box for image encryption.

Bui et al. [78] proposed a hardware optimization strategy for AES implementation in ultra-low power IoT applications to provide multilevel security using different key sizes. Power and energy optimization has been performed for both data path and key expansion. This led to significant reduction in energy per bit to a value of 1 pJ/b at 10 MHz at 0.6 V and throughput of 28 Mb/s.

Duran et al. [79] proposed an AES-128/256 S-box acceleration scheme which uses a custom S-box unit connected as a logic unit. All S-box calculations were performed using pipelined and pure combinational approach resulting in lower memory access and lower energy consumption of 9.7 pJ/bit. A large portion of the energy consumed in an AES circuit is during the substitution process, hence S-box architecture plays a crucial role.

Morioka et al. [80] proposed a low power S-box architecture in which signal arrival time at gates are very close if their depth from main input is identical. This led to a minimalistic power consumption of 29  $\mu$ W at 10 MHz using 0.13  $\mu$ m CMOS technology. In recent studies, pipelining of AES architecture has proven to significantly increase the throughput of the encryption system and accompanied power reduction.

Chellappa et al. [81] proposed a fully pipelined 256-bit AES design with pulse clocked latches connecting the pipeline stages that can be made transparent when in use resulting a 7.6% decrease in energy. This design could deliver 64 Gb/s encryption when fabricated on 90 nm technology.

Oukili et al. [82] presented a 5-stage pipeline S-box design to increase the maximum speed and frequency of the AES system. S-box transformations plays a crucial role in the complexity of AES algorithm therefore parallel processing using pipelined stages helped the proposed method to achieve a throughput of 79 Gbps.

Kshirsagar et al. [83] proposed interchanging of byte substitution and shift rows operations in the AES implementation which helped to streamline the processing of 16 data blocks into 4 parallel blocks of data. This led to significant reduction in hardware area consumption by 56% and increasing the throughput by 4.25%.

In their study, Rais et al. (2009) examined the effective hardware design and FPGA implementation of a 128-bit AES using a design based on residue prime numbers. They analyzed various hardware models of AES [84]. Fan et al. (2008) discussed a high-speed, high-throughput design for AES 128-bit, focusing on the implementation of a content addressable memory-based SBox with a pipeline structure that minimizes delay compared to other designs [85].

H. Samiee, R.E. Atani, and H. Amindavar (2011) proposed a novel area-throughput optimized architecture for the AES algorithm, concentrating on a normal basis composite field arithmetic architecture model [86]. Hodjat et al. (2004) designed a fully pipelined structure for a high-speed AES processor with a throughput speed of 21.5 Gbps [87].

In terms of application-based implementation, Daemen et al. (1998) introduced the block cipher Rijndael for smart card applications [88]. Narang et al. (2012) conducted a literature survey on various wireless security designs [89].

Ranjeeth et al. (2012) provided an in-depth analysis of the WiMax structure and security issues in their research. They discussed the different security algorithms used in the WiMax MAC layer [90]. Yu et al. (2005) explored a compact hardware implementation of AES, presenting an efficient hardware structure for AES [91]. Uribe et al. focused on the privacy key management of the WiMax MAC layer [92].

## CHAPTER 3

### PRINCIPLES OF ENCRYPTION

#### 3.1. SHANNON'S THEORY OF CONFUSION AND DIFFUSION

Encryption techniques often incorporate the principles of confusion and diffusion, as outlined in Claude Shannon's theory of cryptography [93]. Detailed description of these concepts are mentioned below:

- 3.1.1. Confusion: Confusion involves introducing complexity and randomness into the relationship between the encryption key and the ciphertext. It aims to make the relationship between the two as obscure as possible, making it difficult for an attacker to derive any meaningful information about the plaintext from the ciphertext without knowledge of the key. Confusion is typically achieved through the use of substitution techniques, where elements of the plaintext are replaced with different elements in the ciphertext based on the key [94].
- 3.1.2. Diffusion: Diffusion refers to spreading the influence of each plaintext element throughout the ciphertext, making the statistical properties of the plaintext less apparent in the ciphertext. Diffusion aims to ensure that changes in the plaintext result in widespread changes in the ciphertext, thereby hiding any patterns or regularities. Diffusion is typically achieved through permutation or transposition techniques, which rearrange the positions of elements within the ciphertext based on the key.

By combining confusion and diffusion, an encryption algorithm can provide a higher level of security and resistance against various cryptographic attacks. Shannon's theory highlights the importance of these principles in achieving robust and effective encryption

schemes. Encryption algorithms like the Advanced Encryption Standard (AES) heavily rely on confusion and diffusion to ensure the confidentiality and integrity of data.

Confusion and diffusion help to thwart statistical analysis, frequency analysis, and other known-plaintext attacks by creating a complex and unpredictable relationship between the plaintext, encryption key, and resulting ciphertext. These concepts form the basis of modern encryption techniques and play a vital role in ensuring the strength and resilience of cryptographic algorithms.

### **3.2. FRAMEWORK OF ENCRYPTION ALGORITHM**

Encryption and ciphers are built upon various scientific theories and mathematical principles. Number theory, probability theory, information theory, Boolean algebra, cryptographic hash functions, modular arithmetic, and complexity theory are some of the key foundations [95]. These concepts provide the framework for designing secure encryption algorithms, ensuring data confidentiality, integrity, and authenticity. Number theory enables the creation of mathematical structures used in encryption, while probability theory and information theory contribute to generating randomness and measuring information.

Number theory, with concepts like prime numbers and modular arithmetic, is vital for encryption algorithms such as RSA. It enables the generation of cryptographic keys and the efficient computation of mathematical operations. Probability theory plays a role in generating random numbers, essential for creating strong encryption keys and ensuring unpredictability. Information theory offers insights into measuring and quantifying information, guiding the design of encryption algorithms that aim to maximize data entropy and minimize predictability. Boolean algebra, rooted in logic gates and binary operations, provides the logical foundation for cryptographic functions and operations, allowing for secure transformations and computations [96].



Cryptographic hash functions, based on number theory and discrete mathematics, generate fixed-size outputs (hashes) that verify data integrity and provide digital signatures. Modular arithmetic, a branch of number theory, facilitates computations in finite fields, enabling efficient and secure cryptographic operations. Complexity theory analyzes the computational complexity of encryption algorithms, assessing their resistance to various attacks and providing insights into the required computational resources. By drawing from these scientific theories and mathematical principles, encryption and ciphers are designed to protect sensitive information, ensure secure communication, and uphold the principles of confidentiality, integrity, and authenticity.

## CHAPTER 4

### METHODOLOGY

#### 4.1. ALPHANUMERIC DATA CIPHER

AES algorithm makes use of iterative process to obscure the relationship between the key and the cipher text. Each iteration step performs fixed number of substitutions and permutations to encrypt the input message. The number of iterations depends on the size of the cipher key used. Key can be of size 128,192 and 256 bits and subsequently the number of iterations are 10,12 or 14 respectively. Similarly, during the decryption process the encrypted message is passed through these iterative steps in reverse order to obtain the original input message. Each of these steps consists of four processes named (1) Substitute bytes (2) Shift rows (3) Mix Column (4) Add round keys the details of which has been discussed in brief in this section [97]. Before the first iteration is performed, a pre round transformation takes place along with key expansion in which the size of cipher key is extended from 4 words (in case of 128 bits key) to 44 words, where each word is of size 4 bytes.

Four words of this expanded key is then supplied to each of the 10 iterative rounds as well as to the pre-round transformation. All operations on the data are performed on a block size of 128 bits in the AES irrespective of the key size, hence the input message is divided into 4x4 matrices where each element in the matrix is of size 1 byte. After each operation the results are stored in a 4x4 intermediate state matrix on which further operations are performed [98]. Figure 4.1 shows the flow of input data through N rounds of transformation where each round is provided with extended key.

The cipher key is first XORed with the input data matrix in the pre-round transformation to produce the state matrix which then acts as the input for subsequent rounds.

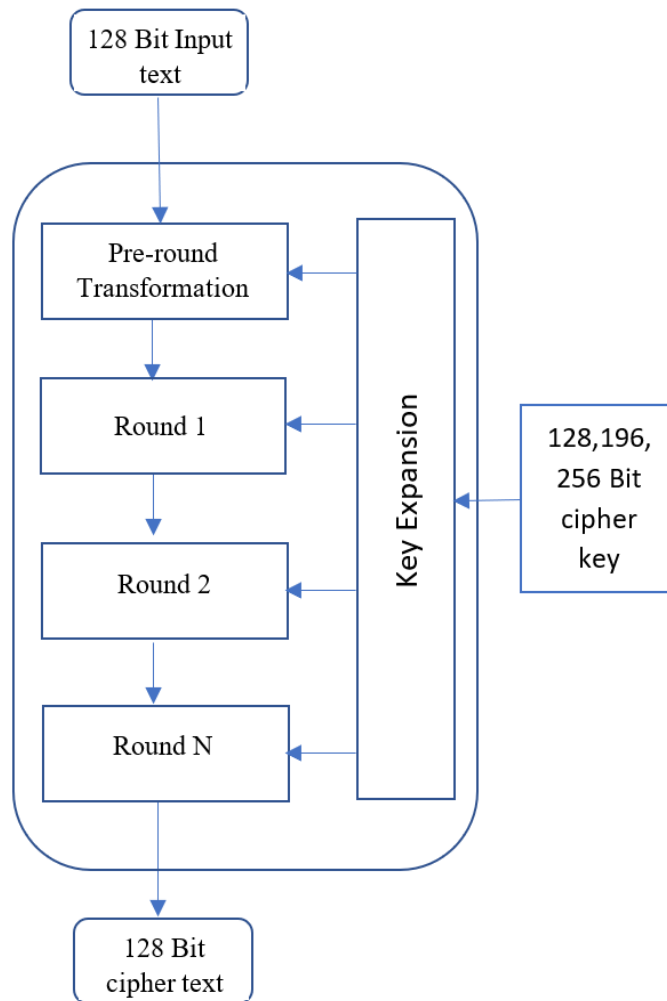


Fig.4.1. Block diagram of N rounds 128-bit AES algorithm.

#### 4.1.1 Key Expansion

The cipher key is supplied by the user as a plain text which is first converted to hexadecimal form. Consider a 128-bit key arranged in the form of a 4x4 matrix with each column representing a 4-byte word as shown in Figure 4.2. This 4-word representation of the original key is expanded to 44 words and supplied to the pre-round transformation stage along with 10 transformation rounds [99]. The first 4 words of the expanded key representation is the original key itself which is XORed with the input message (in hex) and the intermediate state matrix is passed on to round 1 as input.

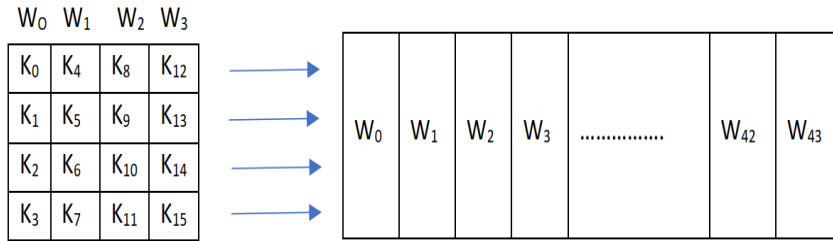


Fig.4.2. Key Expansion process for 128-bit symmetric key.

Figure 4.3 depicts the process of obtaining the next four words ( $W_4-W_7$ ) from the first 4 words ( $W_0-W_3$ ) of the original cipher key using the “g” function.

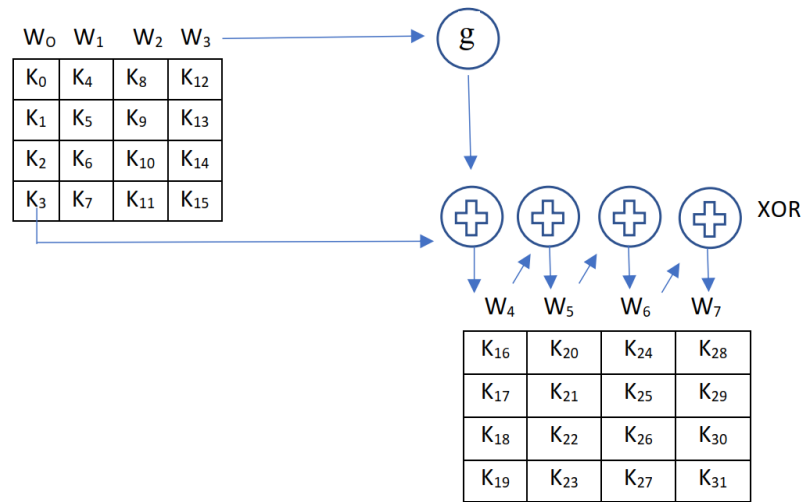


Fig.4.3. Key expansion operation for obtaining next 4 key words.

Mathematically this can be defined as:

$$W_4 = W_0 \oplus g(W_3) \tag{1}$$

$$W_5 = W_4 \oplus W_1 \tag{2}$$

$$W_6 = W_5 \oplus W_2 \tag{3}$$

$$W_7 = W_6 \oplus W_3 \tag{4}$$

Here  $g(W_3)$  is obtained by performing a 3-step process, one-byte circular left shift of the word  $W_3$  to get  $X_1$  and then performing a byte substitution on each byte of  $X_1$  using S-box to get  $Y_1$ .

Finally,

$$g(W_3) = Y_1 \oplus R_{CON}[j] \tag{5}$$

Here  $R_{CON}[j]$  is the round constant described for each iteration round as shown in Table 1.

Table 1. Round constant table for 10 rounds of transformation in AES.

R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
01	02	04	08	10	20	40	80	1B	36
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00

Figure 4.4 depicts the four processes in round 1 to 9 of the 128-bit AES algorithm with a cipher key size of 128 bit. The final round R10 has only 3 internal processes as the Mix column operation is excluded from it.

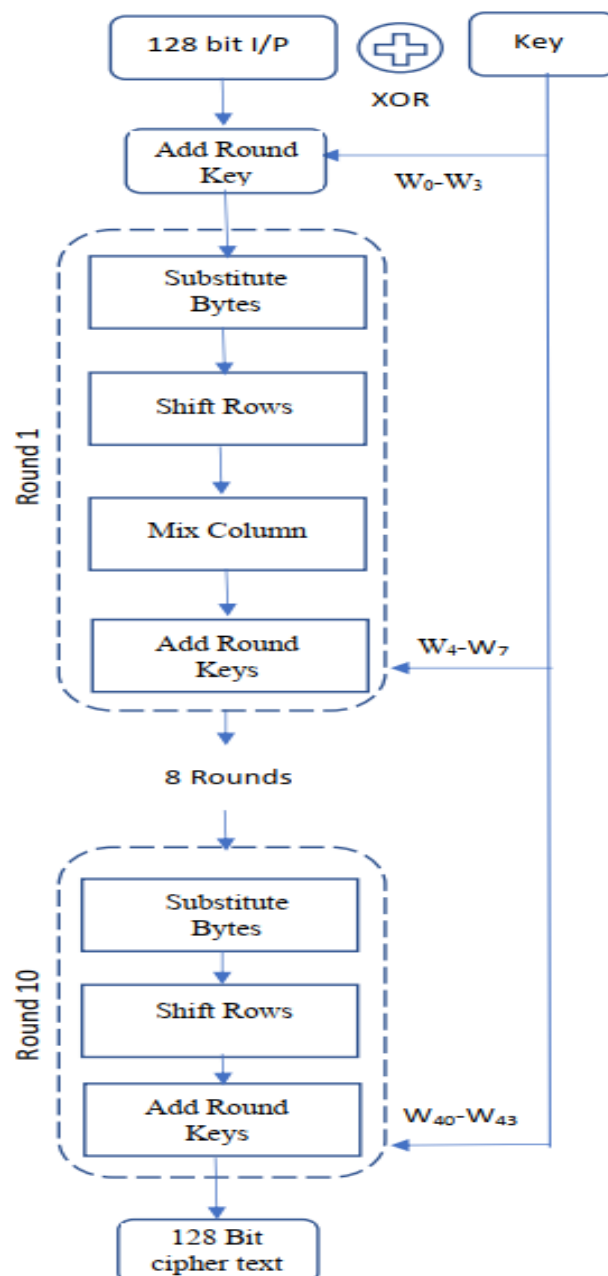


Fig.4.4. XOR of expanded key with each round of AES.

### 4.1.2 SUBSTITUTE BYTES

Substitute byte forms the first step in each round. Here the output of pre-round transformation is used as an input to this step in which the elements of intermediate state matrix are replaced using an S-box table as shown in Figure 4.5. S-box is a major component of any cryptography algorithm, which performs substitution. It is a 16x16 Look Up Table (LUT) with its elements ranging from 00 to FF. Substitution has the largest share in power consumption and is one of the most complex process in the entire algorithm [100].

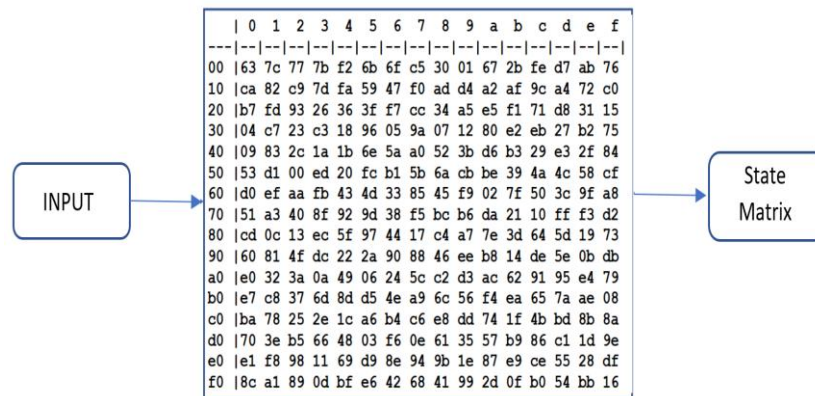


Fig. 4.5. Substitution byte transformation using 16x16 S-box

### 4.1.3 SHIFT ROWS

The result obtained from substitution then undergoes shift rows operation in which circular left shift is performed on the state array as shown in figure 4.6. There is no shift in the first row of the matrix, a 1-byte circular left shift in second row, followed by 2 and 3 bytes circular left shift in row 3 and 4 respectively [101].

$S_{00}$	$S_{01}$	$S_{02}$	$S_{03}$
$S_{10}$	$S_{11}$	$S_{12}$	$S_{13}$
$S_{20}$	$S_{21}$	$S_{22}$	$S_{23}$
$S_{30}$	$S_{31}$	$S_{32}$	$S_{33}$

Fig 4.6. Shift rows operation on 128-bit block size data

#### 4.1.4 MIX COLUMNS

In this step, the state matrix obtained from shift row operation undergoes word by word multiplication with a constant matrix as shown in figure 4.7 [102].

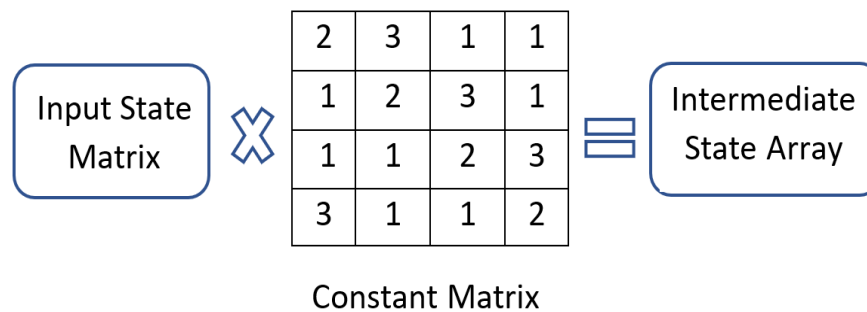


Fig. 4.7. Multiplication of state matrix with constant matrix

#### 4.1.5 ADD ROUND KEYS

In this final step round keys are XORed with the output obtained from mix column operation and the results are passed on to the next round [103]. After all the 4 steps are performed iteratively for all the 10 rounds final cipher text is obtained. This encrypted message is in hexadecimal form. For performing decryption, a similar approach is undertaken where the encrypted message is passed through the 10 rounds of transformation with inverted Mix Columns and inverted shift rows. An inverted S-box is used in the substitute bytes step and the output obtained is the original message in hexadecimal form which can be further converted to plain text.

#### **4.1.6 PIPELINED ARCHITECTURE**

All operation inside the AES is performed on 128-bit block size of data, that means the input message (in hex) needs to be broken to and presented in 128-bit 4x4 matrices before being processed inside the AES [104]. Since the data blocks passes through the 10 iterative rounds in a sequential manner therefore only one block is processed at a given time in a particular round leaving the subsequent stages unused or idle. Parallel pipelining enables multiple data blocks to be processed parallelly in different stages hence making the encryption process of large messages or images faster [105].

### **4.2. MULTIMEDIA DATA CIPHER**

AES encrypts a stream of input data by performing several substitution and shifting operations in an iterative manner. A 256-bit AES algorithm comprises of 14 rounds of transformation. In the proposed method a 256-bit key is entered by the user (as plain text) along with the input image that needs to be encrypted. The cipher key is symmetric in nature i.e., the same key is utilized for encryption and decryption process [106]. The symmetric key undergoes expansion from 8 words (4 bytes each) to 60 words using the key expansion process. Each iterative round manipulates the input through a series of substitutions and permutations based on Shannon's theory of confusion and diffusion [107]. Each of these rounds consist of four processes, Substitute bytes, Shift rows, Mix Column and Add round keys [108]. Only the last round is different from previous rounds as it does not contain mix-column step. Before the input matrix is passed through the first stage, the key is XORed with the input matrix and then the key expansion process takes place. All operations inside the AES are performed on 128-bit packets of data arranged in a 4x4 matrix with each element of size 1 byte.

#### **4.2.1 IMAGE ENCRYPTION USING AES-256**

Digital image undergoes pre-processing before it is sent for encryption. In this paper a digital color image has been taken under consideration. First the Red, Green, and Blue (RGB) layers are extracted from the image [109]. This image is then converted to grey scale format of size 256x256 pixels. This grey scale image is then broken into 4x4 matrices (128-bits each) with each element of size 1 byte. These matrices are fed to the input of



AES-256 algorithm along with the 256-bit cipher key and encrypted  $4 \times 4$  matrices are obtained at output, which are then stitched together to form the  $256 \times 256$  encrypted image [110]. Each set of  $4 \times 4$  matrix undergoes exactly same encryption process.

Figure 4.8 shows the flow chart for RGB image encryption with an image pre-processing component at top of the AES cipher.

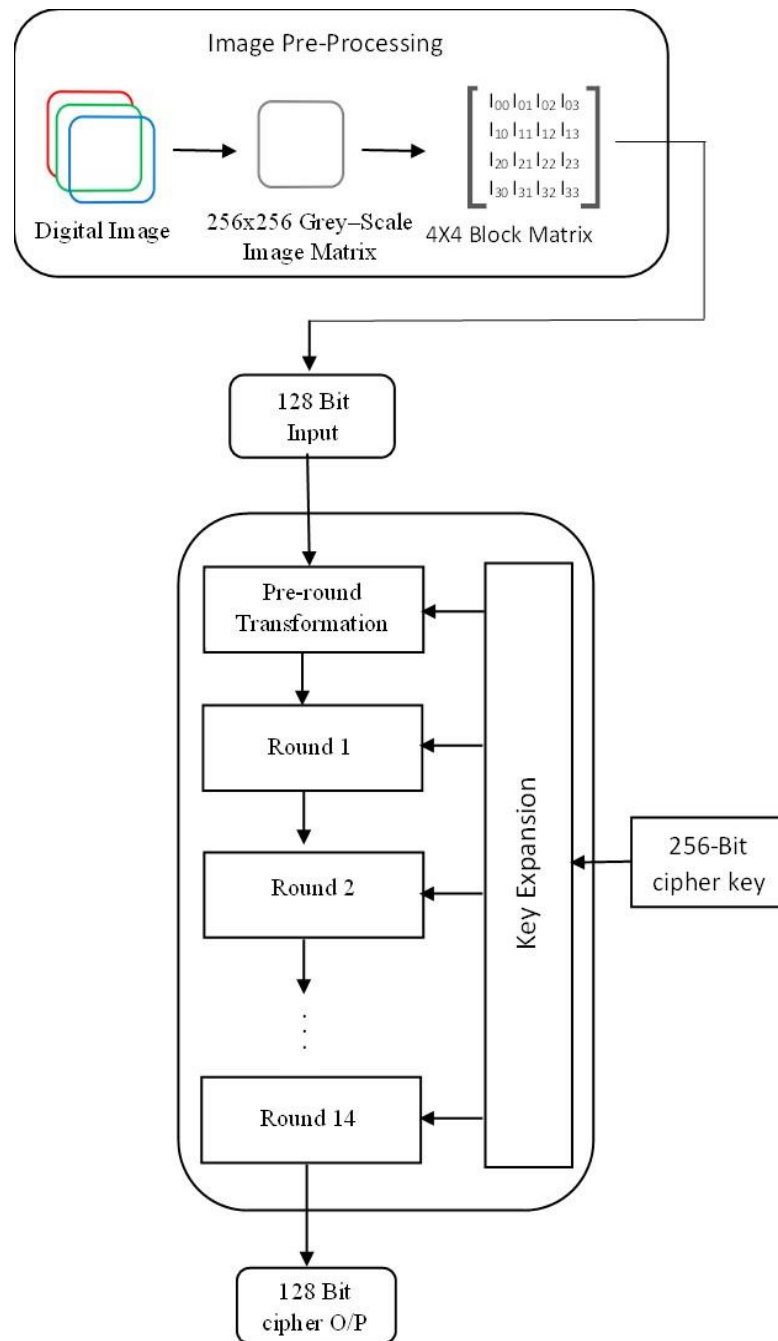


Fig.4.8. Flow chart for encryption process of RGB Image

## CHAPTER 5

### RESULTS

#### 5.1. FPGA SYNTHESIS RESULTS FOR TEXT ENCRYPTION

To perform encryption of “*sample message 1*” with 128 bit key “*secure password1*” the resulting 128-bit encrypted output will be “786e4e6532761c57e253cc34814c233c”. The decryption module will take this 128-bit hexadecimal encrypted data as input and same 128 bit key as input to generate the decipher text message “*sample message 1*”. The energy consumed per bit in transformation during the entire process was 7 pJ/bit and throughput of encrypted results was 68 Gbps. Figure 5.1 and 5.2 shows the produced encryption and decryption outputs.

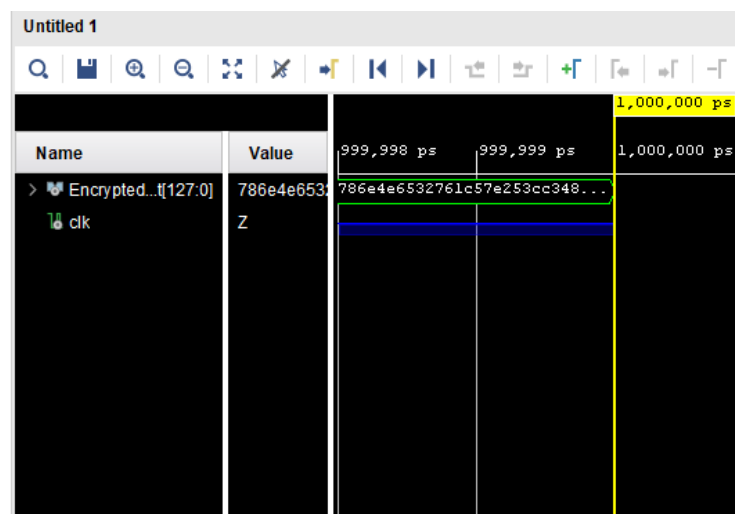


Fig. 5.1. Encrypted 128-bit hexadecimal output.

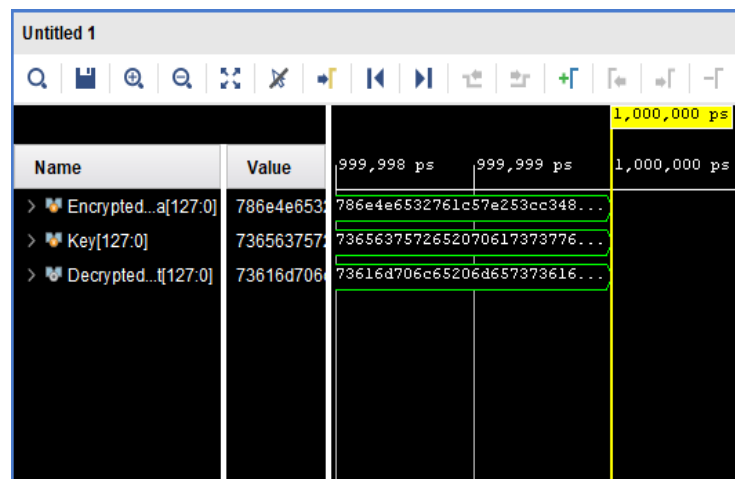


Fig. 5.2. Shows the Encrypted 128-bit hexadecimal results and decrypted message.

Xilinx Artix-7 FPGA device based on 28nm technology was used for hardware evaluation and Verilog HDL for programming. This FPGA provides highest performance/watt and has 215,360 logic cells for design implementation [111]. For the AES-128 implementation, a total of 23,689 logic cells were used which comes to about 11% resource utilization on the FPGA. Simulation and synthesis task has been performed on Xilinx Vivado v2021.1 IDE. Figure 5.3 shows simulation of the given algorithm run for 1000ns and the encryption status in Vivado tcl console, peak memory used, gain, and time elapsed in producing simulation results.

```
# run 1000ns
Encrypted_Message = 786e4e6532761c57e253cc34814c233c
Decrypted_output = sample message 1
Encryption success!
xsim: Time (s): cpu = 00:00:13 ; elapsed = 00:00:10 . Memory (MB): peak = 2137.082 ; gain = 32.738
INFO: [USF-XSim-96] XSim completed. Design snapshot 'TB_behav' loaded.
INFO: [USF-XSim-97] XSim simulation ran for 1000ns
launch_simulation: Time (s): cpu = 00:00:17 ; elapsed = 00:00:50 . Memory (MB): peak = 2137.082 ; gain = 38.902
```

Fig. 5.3. simulation runtime and status of encryption.

Figure 5.4. shows the AES top module in Xilinx Vivado simulator with 128 bits of input data and cipher key and the clock pulse. Dataout[127:0] is the encrypted output message in hex form. Figure 5.5. Shows the Rounds operation module with datain[127:0], keyin[127:0], clk, rc[3:0] input vectors and keyout[127:0], rndout[127:0] output vectors.

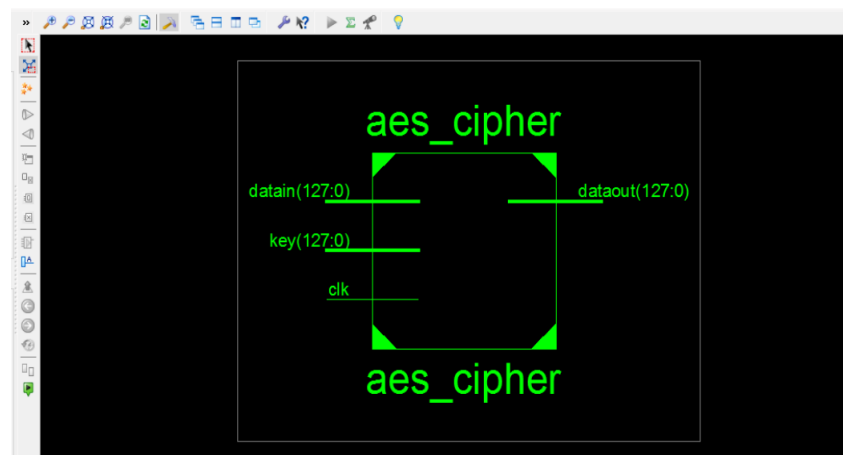


Fig. 5.4. AES top module with input data, key and clock pulse and encrypted output.

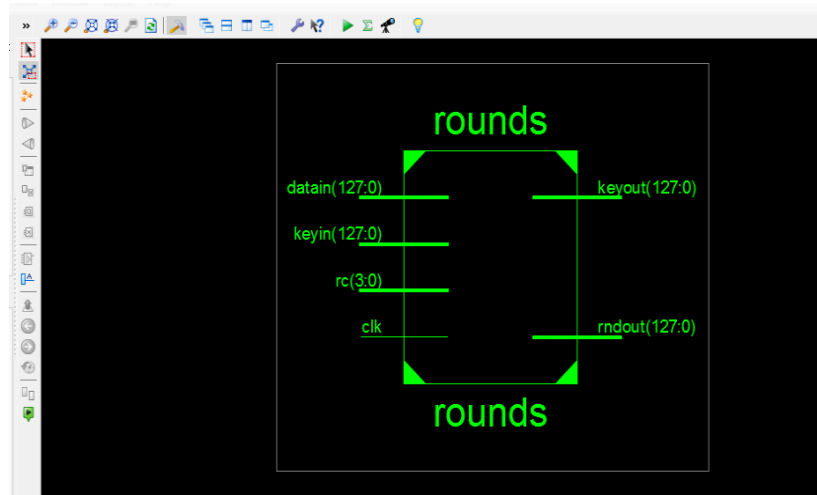


Fig. 5.5. Round transformation module with its input and output vectors

Table 5.1.1. shown below provides a variation in the energy/bit (pJ/b) of the design with varying temperature in kelvin at various input supply voltages (V).

Table 5.1.1. Energy/bit variation with temperature and input voltage

S. No	Temperature (K)	Input Voltage (V)	Energy/bit (pJ/b)
1	290	0.9	7
2	300	1	7.09
3	310	1.05	7.23
4	320	1.1	8.10

Table 5.1.2. shown below provides a variation in the dynamic power (W) and the leakage power (W) at various input supply voltages (V).

Table 5.1.2. Input voltage vs dynamic and leakage power

S. No	Input Voltage (V)	Dynamic Power (W)	Leakage Power (W)
1	0.9	0.69	0.09
2	1	0.83	0.11
3	1.05	0.9	0.12
4	1.1	1.01	0.14

Table 5.1.3. shown below provides a comparison of the throughput and energy consumed in per bit transformation of the presented parallel pipelined architecture with existing work.

Table 5.1.3. Result comparison of presented method with existing research.

S.No	Author	Algorithm	Throughput	Energy	Frequency
1	D.H Bui (2017) [6]	Data Path optimization	28 Mb/s	1 pJ/bit	10 MHz
2	C. Duran (2022) [7]	S-box acceleration	-	9.7 pJ/bit	100 MHz
3	S. Chellapa (2015) [9]	Fully Pipelined	64 Gbps	-	500 MHz

## 5.2. SIMULATION RESULTS FOR IMAGE ENCRYPTION

The implementation and simulation of AES-256 for digital image encryption has been carried out on MATLAB 2021a. The 256-bit key considered for encryption is “*secure encryption password 10111*” (32 characters in plain text). Cipher key representation in hex format “73656375726520656e6372797074696f6e2070617373776f7264203130313131”. Figure 5.6 shows the RGB image along with its histogram plot and its gray scale conversion and corresponding histogram plot. The 4×4 image matrices extracted from the gray scale image that is fed into the encryptor is shown in figure 9. The packets of 4×4 encrypted images obtained at the encryptor output are stitched together to form the 256×256 pixels encrypted form of the original gray scale image. Figure 10 shows the 4×4 encrypted image matrix (in Hex).

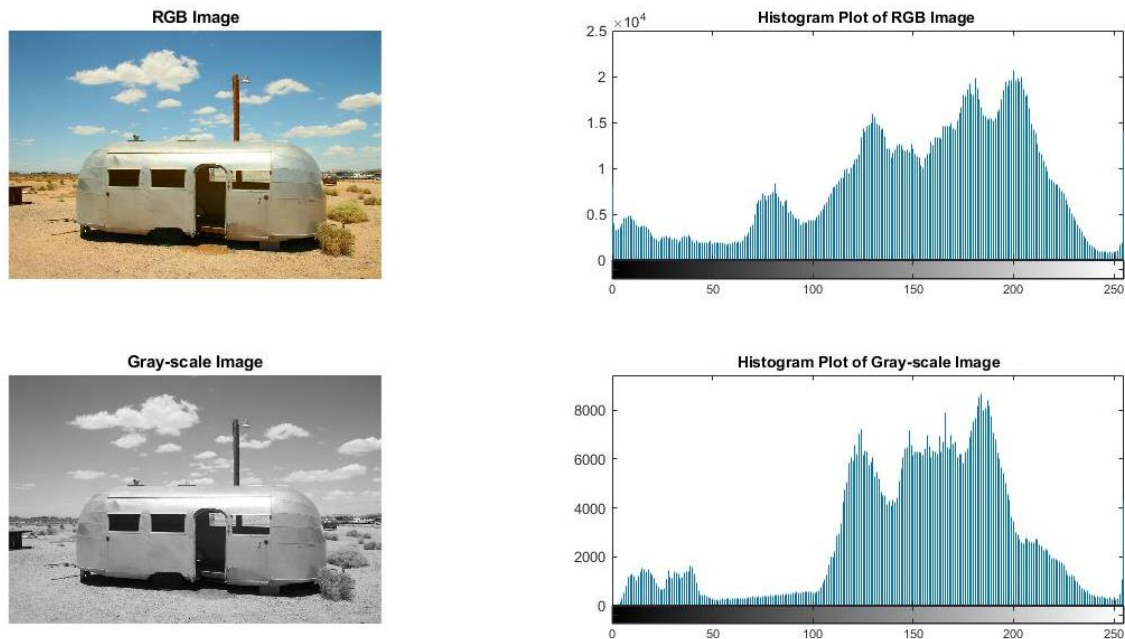


Fig. 5.6. RGB Image and Gray scale image with their Histogram plots.

Figure 11 shows the Encrypted image and its histogram plot along with the recovered image (decrypted) and its corresponding histogram plot. The results exhibited a high PSNR of 61 dB for the decrypted image and the correlation between the input digital image and the decrypted (recovered) image was found to be 0.994. The MSE between the input and the decrypted image was calculated to be 0.0030, indicating very low levels of distortion in the recovered image. This proves that AES-256 is highly suitable for image encryption and is robust to any distortions induced during the encryption processes. Decryption of the encrypted image matrix follows similar approach as the encryption, an inverted S-box is utilized but the flow remains mostly same [112]. Post processing of obtained decrypted matrix is done for stitching of image and conversion to original form.

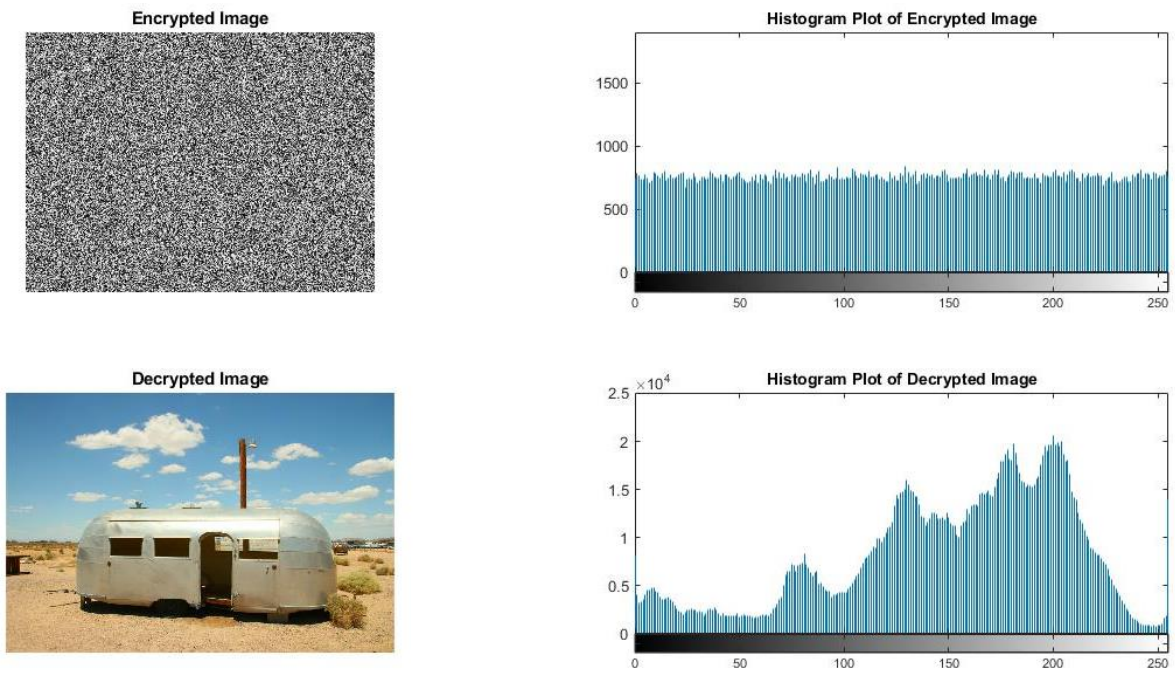


Fig. 5.9. Encrypted & Decrypted images with their Histogram plots.

```

Command Window
Input Image 4x4 Matrix (in Hex):
    73  72  61  66
    65  34  c1  6f
    23  7b  61  20
    75  70  77  31
fx >> |
    
```

Fig. 5.7. Input image matrix to the AES-256.

```

Command Window
Encrypted Image 4x4 Matrix (in Hex):
    20  d2  44  90
    65  34  c1  62
    c3  1b  36  30
    25  25  07  a1
fx >> |
    
```

Fig. 5.8. 4x4 encrypted image matrix

## CHAPTER 6

### CONCLUSIONS AND FUTURE SCOPE

This dissertation proposes an efficient high throughput pipelined architecture of 128-bit AES cipher algorithm. A detailed study of the transformation processes in encryption/decryption of data has been presented to analyse the results obtained from each stage of the AES. The results exhibit high-rate encryption of 68 Gbps and a low energy consumption of 7 pJ/bit. Both encryption and decryption processes have been demonstrated for an alphanumeric text message using a 128-bit symmetric cipher key. This architecture makes it useful for applications in battery operated devices which have speed and power consumption constraints.

An efficient and high throughput 256-bit AES cipher algorithm for encrypting digital image has also been proposed. A user input symmetric cipher key-based encryption process was simulated along with pre-processing of image data. A detailed study of the input image and the encrypted/decrypted images was carried out along with their histogram plots. The results exhibited high PSNR of 61 dB for the decrypted image and the correlation between the input image and the decrypted image was found to be 0.994. The MSE between the two images was calculated to be 0.0030.

This architecture proved to be more efficient and easier to implement as compared to other techniques adopted for image encryption and has vast applications in messaging apps and data sharing platforms. The results proved the robustness of the algorithm as very little deviations were observed in the recovered image. AES is a computationally expensive and power-hungry algorithm hence not suitable for encryption of high-resolution images when there are low power constraints.



Future improvements on this work include pipelining of AES algorithm for higher throughput and use of a low power multiplier in the mix-column step to reduce the power consumption. AES-256 can be made furthermore secure using a dynamic S-box in the substitution byte step.

These facts highlight the significance and strength of the AES encryption algorithm, making it a widely trusted and respected cryptographic standard. Overall, AES offers a strong combination of security, efficiency, compatibility, and proven reliability, making it an excellent choice for multimedia encryption applications. In conclusion, AES has established itself as a highly dependable and secure encryption algorithm, extensively adopted across diverse sectors and applications. Its resilience, immunity to known attacks, and adaptability in key sizes have positioned it as the preferred choice for ensuring the confidentiality and integrity of data.

Looking ahead, the future of AES entails ongoing research and development aimed at optimizing its performance and addressing emerging challenges. Efforts are underway to fine-tune AES implementations for various platforms and devices, leveraging hardware acceleration techniques to enhance speed and efficiency. Additionally, a comprehensive analysis of potential vulnerabilities is crucial to ensuring AES remains impervious to new attack methodologies and quantum computing risks. As technology evolves, the demand for robust and efficient encryption solutions will persist. AES is expected to continue playing a vital role in meeting these demands, serving as a fundamental pillar for data protection in an increasingly interconnected and digital landscape. Continuous advancements and innovative approaches to AES will reinforce its status as a leading encryption standard, empowering secure communication, safeguarding sensitive information, and bolstering overall digital security.

In terms of power and performance, one area of focus is hardware acceleration, where specialized hardware components and instructions are utilized to offload AES computations, improving encryption and decryption speeds. This includes the integration of AES-NI (AES New Instructions) in modern CPUs, which provide dedicated instructions for AES operations, resulting in significant performance gains.

Parallel processing is another avenue for enhancing AES performance. By leveraging multi-core architectures and distributed computing techniques, encryption and decryption tasks can be divided into smaller parallel tasks, allowing for faster processing and improved throughput.

Furthermore, advancements in hardware design, such as the use of FPGA (Field-Programmable Gate Array) or ASIC (Application-Specific Integrated Circuit) technologies, offer opportunities for optimized AES implementations. These specialized hardware solutions can be customized to efficiently execute AES algorithms, resulting in faster and more power-efficient encryption and decryption operation.

## REFERENCES

- [1] W. H. Baker and L. Wallace, "Is Information Security Under Control?: Investigating Quality in Information Security Management," in *IEEE Security & Privacy*, vol. 5, no. 1, pp. 36-44, Jan.-Feb. 2007, doi: 10.1109/MSP.2007.11.
- [2] R. Davis, "The data encryption standard in perspective," in *IEEE Communications Society Magazine*, vol. 16, no. 6, pp. 5-9, November 1978, doi: 10.1109/MCOM.1978.1089771.
- [3] M. Klang, "Who do you trust? Beyond encryption, secure e-business," *Decision Support Systems*, vol. 31, no. 3, pp. 293-301, 2001.
- [4] P. Singh and K. Kaur, "Database security using encryption," 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), Greater Noida, India, pp. 353-358, 2015, doi: 10.1109/ABLAZE.2015.7155019.
- [5] B. Furht, D. Socek, and A. M. Eskicioglu, "Fundamentals of multimedia encryption techniques," in *Multimedia Security Handbook*, pp. 95-132, CRC Press, 2004.
- [6] D. Coppersmith, "The Data Encryption Standard (DES) and its strength against attacks," in *IBM Journal of Research and Development*, vol. 38, no. 3, pp. 243-250, May 1994, doi: 10.1147/rd.383.0243.
- [7] D. Coppersmith, D. B. Johnson and S. M. Matyas, "A proposed mode for triple-DES encryption," in *IBM Journal of Research and Development*, vol. 40, no. 2, pp. 253-262, March 1996, doi: 10.1147/rd.402.0253.
- [8] Seung-Jo Han, Heang-Soo Oh and Jongan Park, "The improved data encryption standard (DES) algorithm," *Proceedings of ISSSTA'95 International Symposium on Spread Spectrum Techniques and Applications*, Mainz, Germany, pp. 1310-1314 vol.3, 1996, doi: 10.1109/ISSSTA.1996.563518.
- [9] G. Singh, "A study of encryption algorithms (RSA, DES, 3DES and AES) for information security," *International Journal of Computer Applications*, vol. 67, no. 19, 2013.
- [10] T. Nie and T. Zhang, "A study of DES and Blowfish encryption algorithm," *TENCON 2009 - 2009 IEEE Region 10 Conference*, Singapore, pp. 1-4, 2009, doi: 10.1109/TENCON.2009.5396115.
- [11] Z. Yun-peng, L. Wei, C. Shui-ping, Z. Zheng-jun, N. Xuan and D. Wei-di, "Digital image encryption algorithm based on chaos and improved DES," *2009 IEEE International Conference on Systems, Man and Cybernetics*, San Antonio, TX, USA, pp. 474-479, 2009, doi: 10.1109/ICSMC.2009.5346839.

- [12] H. S. Deshpande, K. J. Karande and A. O. Mulani, "Efficient implementation of AES algorithm on FPGA," 2014 International Conference on Communication and Signal Processing, pp. 1895-1899, 2014 doi: 10.1109/ICCSP.2014.6950174.
- [13] Q. Li, C. Zhong, K. Zhao, X. Mei and X. Chu, "Implementation and Analysis of AES Encryption on GPU," 2012 IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems, Liverpool, UK, pp. 843-848, 2012, doi: 10.1109/HPCC.2012.119.
- [14] Chih-Chung Lu and Shau-Yin Tseng, "Integrated design of AES (Advanced Encryption Standard) encrypter and decrypter," Proceedings IEEE International Conference on Application- Specific Systems, Architectures, and Processors, San Jose, CA, USA, 2002, pp. 277-285, doi: 10.1109/ASAP.2002.1030726.
- [15] A. M. Abdullah, "Advanced encryption standard (AES) algorithm to encrypt and decrypt data," Cryptography and Network Security, vol. 16, pp. 1-11, 2017.
- [16] P. Hamalainen, T. Alho, M. Hannikainen and T. D. Hamalainen, "Design and Implementation of Low-Area and Low-Power AES Encryption Hardware Core," 9th EUROMICRO Conference on Digital System Design (DSD'06), Cavtat, Croatia, 2006, pp. 577-583, doi: 10.1109/DSD.2006.40.
- [17] M. J. Dworkin, E. B. Barker, J. R. Nechvatal, J. Foti, L. E. Bassham, E. Roback, and J. F. Dray Jr., "Advanced encryption standard (AES)," 2001.
- [18] J. Nechvatal, E. Barker, L. Bassham, W. Burr, M. Dworkin, J. Foti, and E. Roback, "Report on the development of the Advanced Encryption Standard (AES)," Journal of research of the National Institute of Standards and Technology, vol. 106, no. 3, pp. 511, 2001.
- [19] N. Aleisa, "A Comparison of the 3DES and AES Encryption Standards," International Journal of Security and Its Applications, vol. 9, no. 7, pp. 241-246, 2015.
- [20] A. M. Deshpande, M. S. Deshpande and D. N. Kayatanavar, "FPGA implementation of AES encryption and decryption," 2009 International Conference on Control, Automation, Communication and Energy Conservation, Perundurai, India, 2009, pp. 1-6.
- [21] S. Heron, "Advanced encryption standard (AES)," Network Security, vol. 2009, no. 12, pp. 8-12, 2009.
- [22] F. G. Deng and G. L. Long, "Controlled order rearrangement encryption for quantum key distribution," Physical Review A, vol. 68, no. 4, p. 042315, 2003.
- [23] Y. S. Zhang, C. F. Li, and G. C. Guo, "Quantum key distribution via quantum encryption," Physical Review A, vol. 64, no. 2, p. 024302, 2001.

- [24] X. Yi, R. Paulet, and E. Bertino, "Homomorphic encryption," in *Homomorphic Encryption*, Springer International Publishing, pp. 27-46, 2014.
- [25] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, pp. 1-35, 2018.
- [26] X. Zhang, Y. Tang, H. Wang, C. Xu, Y. Miao, and H. Cheng, "Lattice-based proxy-oriented identity-based encryption with keyword search for cloud storage," *Information Sciences*, vol. 494, pp. 193-207, 2019.
- [27] N. Göttert, T. Feller, M. Schneider, J. Buchmann, and S. Huss, "On the design of hardware building blocks for modern lattice-based encryption schemes," in *Cryptographic Hardware and Embedded Systems—CHES 2012: 14th International Workshop*, Leuven, Belgium, September 9-12, 2012. *Proceedings*, vol. 14, pp. 512-529, Springer Berlin Heidelberg.
- [28] F. Benhamouda, J. Camenisch, S. Krenn, V. Lyubashevsky, and G. Neven, "Better zero-knowledge proofs for lattice encryption and their application to group signatures," in *Advances in Cryptology—ASIACRYPT 2014: 20th International Conference on the Theory and Application of Cryptology and Information Security*, Kaoshiung, Taiwan, ROC, December 7-11, 2014. *Proceedings, Part I*, vol. 20, pp. 551-572, Springer Berlin Heidelberg.
- [29] O. Goldreich, "A uniform-complexity treatment of encryption and zero-knowledge," *Journal of Cryptology*, vol. 6, no. 1, pp. 21-53, 1993.
- [30] B. Knott, S. Venkataraman, A. Hannun, S. Sengupta, M. Ibrahim, and L. van der Maaten, "Crypten: Secure multi-party computation meets machine learning," in *Advances in Neural Information Processing Systems*, vol. 34, pp. 4961-4973, 2021.
- [31] J. Katz, R. Ostrovsky, and A. Smith, "Round efficiency of multi-party computation with a dishonest majority," in *Advances in Cryptology—EUROCRYPT 2003: International Conference on the Theory and Applications of Cryptographic Techniques*, Warsaw, Poland, May 4–8, 2003 *Proceedings*, vol. 22, pp. 578-595, Springer Berlin Heidelberg.
- [32] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pp. 169-178, 2009.
- [33] D. Stehlé and R. Steinfeld, "Faster fully homomorphic encryption," in *Advances in Cryptology-ASIACRYPT 2010: 16th International Conference on the Theory and Application of Cryptology and Information Security*, pp. 377-394, 2010.
- [34] D. E. Denning and P. J. Denning, "Data security," *ACM Computing Surveys (CSUR)*, vol. 11, no. 3, pp. 227-249, 1979.
- [35] A. Ramesh and A. Suruliandi, "Performance analysis of encryption algorithms for Information Security," *2013 International Conference on Circuits, Power and Computing*

- Technologies (ICCPCT), Nagercoil, India, pp. 840-844, 2013, doi: 10.1109/ICCPCT.2013.6528957.
- [36] S. Li, C. Li, G. Chen, D. Zhang, and N. G. Bourbakis, "A general cryptanalysis of permutation-only multimedia encryption algorithms," IACR's Cryptology ePrint Archive: Report, 374, 2004.
- [37] A. U. S. Muhammad and F. Özkaynak, "SIEA: secure image encryption algorithm based on chaotic systems optimization algorithms and PUFs," *Symmetry*, vol. 13, no. 5, pp. 824, 2021.
- [38] Y. Zhang, Y. Wang, and X. Shen, "A chaos-based image encryption algorithm using alternate structure," *Science in China Series F: Information Sciences*, vol. 50, no. 3, pp. 334-341, 2007.
- [39] H. S. Mohan and A. R. Reddy, "Performance analysis of AES and MARS encryption algorithms," *International Journal of Computer Science Issues (IJCSI)*, vol. 8, no. 4, pp. 363, 2011.
- [40] F. Özkaynak and A. B. Özer, "Cryptanalysis of a new image encryption algorithm based on chaos," *Optik*, vol. 127, no. 13, pp. 5190-5192, 2016.
- [41] A. A. Yazdeen, S. R. Zeebaree, M. M. Sadeeq, S. F. Kak, O. M. Ahmed, and R. R. Zebari, "FPGA implementations for data encryption and decryption via concurrent and parallel computation: A review," *Qubahan Academic Journal*, vol. 1, no. 2, pp. 8-16, 2021.
- [42] J. Zhang and W. L. Wu, "Authenticated encryption based on SM4 round function," *ACTA ELECTONICA SINICA*, vol. 46, no. 6, pp. 1294, 2018.
- [43] Q. Shen and W. Liu, "A novel digital image encryption algorithm based on orbit variation of phase diagram," *International Journal of Bifurcation and Chaos*, vol. 27, no. 13, p. 1750204, 2017.
- [44] F. X. Standaert, G. Piret, N. Gershenfeld, and J. J. Quisquater, "SEA: A scalable encryption algorithm for small embedded applications," in *Smart Card Research and Advanced Applications: 7th IFIP WG 8.8/11.2 International Conference, CARDIS 2006, Tarragona, Spain, April 19-21, 2006. Proceedings*, vol. 7, pp. 222-236, Springer Berlin Heidelberg.
- [45] T. Okamoto and K. Takashima, "Fully secure functional encryption with general relations from the decisional linear assumption," in *Advances in Cryptology—CRYPTO 2010: 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, vol. 30, pp. 191-208, Springer Berlin Heidelberg.
- [46] C. Li, "Cracking a hierarchical chaotic image encryption algorithm based on permutation," *Signal Processing*, vol. 118, pp. 203-210, 2016.

- [47] D. Stiawan, M. Idris, R. F. Malik, S. Nurmaini, N. Alsharif, and R. Budiarto, "Investigating brute force attack patterns in IoT network," *Journal of Electrical and Computer Engineering*, vol. 2019.
- [48] M. Blumenthal, "Encryption: Strengths and weaknesses of public-key cryptography," *CSRS 2007*.
- [49] F. X. Standaert, "Introduction to side-channel attacks," in *Secure Integrated Circuits and Systems*, pp. 27-42, 2010.
- [50] F. Koeune and F. X. Standaert, "A tutorial on physical security and side-channel attacks," in *Foundations of Security Analysis and Design III: FOSAD 2004/2005 Tutorial Lectures*, pp. 78-108, 2005.
- [51] S. Gold, "Cracking wireless networks," *Network Security*, vol. 2011, no. 11, pp. 14-18, 2011.
- [52] R. Chandramouli, S. Bapatla, K. P. Subbalakshmi, and R. N. Uma, "Battery power-aware encryption," *ACM Transactions on Information and System Security (TISSEC)*, vol. 9, no. 2, pp. 162-180, 2006.
- [53] M. Haleem, C. Mathur, R. Chandramouli and K. Subbalakshmi, "Opportunistic Encryption: A Trade-Off between Security and Throughput in Wireless Networks," in *IEEE Transactions on Dependable and Secure Computing*, vol. 4, no. 4, pp. 313-324, Oct.-Dec. 2007, doi: 10.1109/TDSC.2007.70214.
- [54] A. Romeo, G. Romolotti, M. Mattavelli and D. Mlynek, "Cryptosystem architectures for very high throughput multimedia encryption: the RPK solution," *ICECS'99. Proceedings of ICECS '99. 6th IEEE International Conference on Electronics, Circuits and Systems (Cat. No.99EX357)*, Paphos, Cyprus, 1999, pp. 261-264 vol.1, doi: 10.1109/ICECS.1999.812273.
- [55] X. Guo, J. Hua, Y. Zhang and D. Wang, "A Complexity-Reduced Block Encryption Algorithm Suitable for Internet of Things," in *IEEE Access*, vol. 7, pp. 54760-54769, 2019, doi: 10.1109/ACCESS.2019.2912929.
- [56] M. Chiosa, F. Maschi, I. Müller, G. Alonso, and N. May, "Hardware acceleration of compression and encryption in SAP HANA," in *48th International Conference on Very Large Databases (VLDB 2022)*.
- [57] A. L. Jeeva, D. V. Palanisamy, and K. Kanagaram, "Comparative analysis of performance efficiency and security measures of some encryption algorithms," *International Journal of Engineering Research and Applications (IJERA)*, vol. 2, no. 3, pp. 3033-3037, 2012.
- [58] G. Bertoni, L. Breveglieri, I. Koren, P. Maistri and V. Piuri, "Error analysis and detection procedures for a hardware implementation of the advanced encryption standard," in *IEEE Transactions on Computers*, vol. 52, no. 4, pp. 492-505, April 2003, doi: 10.1109/TC.2003.1190590.

- [59] Chih-Hsu Yen and Bing-Fei Wu, "Simple error detection methods for hardware implementation of Advanced Encryption Standard," in *IEEE Transactions on Computers*, vol. 55, no. 6, pp. 720-731, June 2006, doi: 10.1109/TC.2006.90.
- [60] Bo Yang, Kaijie Wu and Ramesh Karri, "Scan based side channel attack on dedicated hardware implementations of Data Encryption Standard," 2004 International Conference on Test, Charlotte, NC, USA, 2004, pp. 339-344, doi: 10.1109/TEST.2004.1386969.
- [61] H. Lin et al., "An Extremely Simple Multiwing Chaotic System: Dynamics Analysis, Encryption Application, and Hardware Implementation," in *IEEE Transactions on Industrial Electronics*, vol. 68, no. 12, pp. 12708-12719, Dec. 2021, doi: 10.1109/TIE.2020.3047012.
- [62] E. Zavattoni, L. J. D. Perez, S. Mitsunari, A. H. Sanchez-Ramirez, T. Teruya and F. Rodríguez-Henríquez, "Software Implementation of an Attribute-Based Encryption Scheme," in *IEEE Transactions on Computers*, vol. 64, no. 5, pp. 1429-1441, 1 May 2015, doi: 10.1109/TC.2014.2329681.
- [63] R. C. Merkle, "Fast software encryption functions," in *Advances in Cryptology-CRYPTO'90: Proceedings 10*, pp. 477-501, Springer Berlin Heidelberg, 1991.
- [64] D. A. Osvik, J. W. Bos, D. Stefan, and D. Canright, "Fast software AES encryption," in *Fast Software Encryption: 17th International Workshop, FSE 2010, Seoul, Korea, February 7-10, 2010, Revised Selected Papers 17*, pp. 75-93, Springer Berlin Heidelberg, 2010.
- [65] O. Hajihassani, S. K. Monfared, S. H. Khasteh and S. Gorgin, "Fast AES Implementation: A High-Throughput Bitsliced Approach," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 10, pp. 2211-2222, 1 Oct. 2019, doi: 10.1109/TPDS.2019.2911278.
- [66] K. Kim, S. Choi, H. Kwon, Z. Liu, and H. Seo, "FACE-LIGHT: Fast AES-CTR Mode Encryption for Low-End Microcontrollers." *Lecture Notes in Computer Science*, 2020, pp. 102-114, doi: 10.1007/978-3-030-40921-0\_6.
- [67] M. A. Alomari, K. Samsudin and A. R. Ramli, "A Study on Encryption Algorithms and Modes for Disk Encryption," 2009 International Conference on Signal Processing Systems, Singapore, 2009, pp. 793-797, doi: 10.1109/ICSPS.2009.118.
- [68] D. Le, J. Chang, X. Gou, A. Zhang and C. Lu, "Parallel AES algorithm for fast Data Encryption on GPU," 2010 2nd International Conference on Computer Engineering and Technology, Chengdu, China, 2010, pp. V6-1-V6-6, doi: 10.1109/ICCET.2010.5486259.
- [69] X. Zhang and K. K. Parhi, "High-speed VLSI architectures for the AES algorithm," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 9, Sept. 2004, pp. 957-967, doi: 10.1109/TVLSI.2004.832943.
- [70] C. Duran and E. Roa, "A 10pJ/bit 256b AES-SoC Exploiting Memory Access Acceleration," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 3, pp. 1612-1616, March 2022, doi: 10.1109/TCSII.2021.3126984.



- [71] M. Sumio, and A. Satoh. "An optimized S-Box circuit architecture for low power AES design." In International Workshop on Cryptographic Hardware and Embedded Systems, pp. 172-186. Springer, Berlin, Heidelberg, 2002.
- [72] S. Chellappa, C. Ramamurthy, V. Vashishtha and L. T. Clark, "Advanced encryption system with dynamic pipeline reconfiguration for minimum energy operation," Sixteenth International Symposium on Quality Electronic Design, pp. 201-206, 2015 doi: 10.1109/ISQED.2015.7085425.
- [73] S. Oukili and S. Bri, "High speed efficient advanced encryption standard implementation," 2017 International Symposium on Networks, Computers and Communications (ISNCC), 2017, pp. 1-4, doi: 10.1109/ISNCC.2017.8071975.
- [74] Q. Zhang and Q. Ding, "Digital Image Encryption Based on Advanced Encryption Standard (AES)," 2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC), Qinhuangdao, China, 2015, pp. 1218-1221, doi: 10.1109/IMCCC.2015.261.
- [75] Y. Zhang, "Test and Verification of AES Used for Image Encryption." 3D Research, vol. 9, no. 1, 2018, doi: 10.1007/s13319-017-0154-7.
- [76] A. Arab, M. J. Rostami, and B. Ghavami, "An image encryption method based on chaos system and AES algorithm." The Journal of Supercomputing, vol. 75, no. 10, 2019, pp. 6663-6682, doi: 10.1007/s11227-019-02878-7.
- [77] A. Singh, P. Agarwal and M. Chand, "Image Encryption and Analysis using Dynamic AES," 2019 5th International Conference on Optimization and Applications (ICOA), Kenitra, Morocco, 2019, pp. 1-6, doi: 10.1109/ICOA.2019.8727711.
- [78] D. -H. Bui, D. Puschini, S. Bacles-Min, E. Beigné and X. -T. Tran, "AES Datapath Optimization Strategies for Low-Power Low-Energy Multisecurity-Level Internet-of-Things Applications," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 12, pp. 3281-3290, Dec. 2017, doi: 10.1109/TVLSI.2017.2716386.
- [79] C. Duran and E. Roa, "A 10pJ/bit 256b AES-SoC Exploiting Memory Access Acceleration," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 69, no. 3, pp. 1612-1616, March 2022, doi: 10.1109/TCSII.2021.3126984.
- [80] S. Morioka and A. Satoh, "An optimized S-Box circuit architecture for low power AES design," in Cryptographic Hardware and Embedded Systems-CHES 2002: 4th International Workshop Redwood Shores, CA, USA, August 13–15, 2002 Revised Papers, pp. 172-186, Springer Berlin Heidelberg, February 2003.
- [81] S. Chellappa, C. Ramamurthy, V. Vashishtha and L. T. Clark, "Advanced encryption system with dynamic pipeline reconfiguration for minimum energy operation," Sixteenth International Symposium on Quality Electronic Design, 2015, pp. 201-206, doi: 10.1109/ISQED.2015.7085425.

- [82] S. Oukili and S. Bri, "High speed efficient advanced encryption standard implementation," 2017 International Symposium on Networks, Computers and Communications (ISNCC), 2017, pp. 1-4, doi: 10.1109/ISNCC.2017.8071975.
- [83] R. V. Kshirsagar and M. V. Vyawahare, "FPGA Implementation of High Speed VLSI Architectures for AES Algorithm," 2012 Fifth International Conference on Emerging Trends in Engineering and Technology, pp. 239-242, 2012 doi: 10.1109/ICETET.2012.53.
- [84] M. H. Rais and S. M. Qasim, "Efficient hardware realization of advanced encryption standard algorithm using Virtex-5 FPGA," International Journal of Computer Science and Network Security, vol. 9, no. 9, pp. 59-63, 2009.
- [85] C.-P. Fan and J.-K. Hwang, "Implementations of high throughput sequential and fully pipelined AES processors on FPGA," in 2007 International Symposium on Intelligent Signal Processing and Communication Systems, Xiamen, China, 2007, pp. 353-356, doi: 10.1109/ISPACS.2007.4445896.
- [86] H. Samiee, R. E. Atani and H. Amindavar, "A novel area-throughput optimized architecture for the AES algorithm," 2011 International Conference on Electronic Devices, Systems and Applications (ICEDSA), Kuala Lumpur, Malaysia, 2011, pp. 29-32, doi: 10.1109/ICEDSA.2011.5959055.
- [87] A. Hodjat and I. Verbaauwhede, "A 21.54 Gbits/s fully pipelined AES processor on FPGA," 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, Napa, CA, USA, 2004, pp. 308-309, doi: 10.1109/FCCM.2004.1.
- [88] J. Daemen and V. Rijmen, "The block cipher Rijndael," in Smart Card Research and Applications: Third International Conference, CARDIS'98, Louvain-la-Neuve, Belgium, September 14-16, 1998. Proceedings 3, pp. 277-284, Springer Berlin Heidelberg.
- [89] S. Narang, T. Nalwa, T. Choudhury and N. Kashyap, "An efficient method for security measurement in internet of things," 2018 International Conference on Communication, Computing and Internet of Things (IC3IoT), Chennai, India, 2018, pp. 319-323, doi: 10.1109/IC3IoT.2018.8668159.
- [90] M. Chauhan, R. Choubey, and R. Soni, "Survey on Handoff with QoS in WiMAX," International Journal of Computer Applications, vol. 50, no. 16.
- [91] N. Yu and H. M. Heys, "Investigation of compact hardware implementation of the advanced encryption standard," Canadian Conference on Electrical and Computer Engineering, 2005, Saskatoon, SK, Canada, 2005, pp. 1069-1072, doi: 10.1109/CCECE.2005.1557161.
- [92] N. Uribe-Pérez, L. Hernández, D. De la Vega, and I. Angulo, "State of the art and trends review of smart metering in electricity grids," Applied Sciences, vol. 6, no. 3, p. 68, 2016.

- [93] C. E. Shannon, "Communication theory of secrecy systems," in *The Bell System Technical Journal*, vol. 28, no. 4, pp. 656-715, Oct. 1949, doi: 10.1002/j.1538-7305.1949.tb00928.x.
- [94] A. Qayyum et al., "Chaos-Based Confusion and Diffusion of Image Pixels Using Dynamic Substitution," in *IEEE Access*, vol. 8, pp. 140876-140895, 2020, doi: 10.1109/ACCESS.2020.3012912.
- [95] C. A. Sun, Z. Wang, and G. Wang, "A property-based testing framework for encryption programs," *Frontiers of Computer Science*, vol. 8, pp. 478-489, 2014.
- [96] V. M. Lidkea, R. Muresan and A. Al-Dweik, "Convolutional Neural Network Framework for Encrypted Image Classification in Cloud-Based ITS," in *IEEE Open Journal of Intelligent Transportation Systems*, vol. 1, pp. 35-50, 2020, doi: 10.1109/OJITS.2020.2996063.
- [97] K. -L. Tsai, Y. -L. Huang, F. -Y. Leu, I. You, Y. -L. Huang and C. -H. Tsai, "AES-128 Based Secure Low Power Communication for LoRaWAN IoT Environments," in *IEEE Access*, vol. 6, pp. 45325-45334, 2018, doi: 10.1109/ACCESS.2018.2852563.
- [98] C. Lu and S. Tseng, "Integrated design of AES (Advanced Encryption Standard) encrypter and decrypter," *Proceedings IEEE International Conference on Application-Specific Systems, Architectures, and Processors*, pp. 277-285, 2002 doi: 10.1109/ASAP.2002.1030726.
- [99] B. Subramanyan, V. M. Chhabria and T. G. S. Babu, "Image Encryption Based on AES Key Expansion," 2011 Second International Conference on Emerging Applications of Information Technology, pp. 217-220, 2011 doi: 10.1109/EAIT.2011.60.
- [100] S. Kumar, V. K. Sharma and K. K. Mahapatra, "Low latency VLSI architecture of S-box for AES encryption," 2013 International Conference on Circuits, Power and Computing Technologies (ICCPCT), pp. 694-698, 2013, doi: 10.1109/ICCPCT.2013.6528906.
- [101] B. Bhat, A. W. Ali and A. Gupta, "DES and AES performance evaluation," *International Conference on Computing, Communication & Automation*, pp. 887-890, 2015 doi: 10.1109/CCAA.2015.7148500.
- [102] K. Wu, Ramesh Karri, G. Kuznetsov and M. Goessel, "Low cost concurrent error detection for the advanced encryption standard," 2004 International Conference on Test, pp. 1242-1248, 2004, doi: 10.1109/TEST.2004.1387397.
- [103] F. J. D'souza and D. Panchal, "Advanced encryption standard (AES) security enhancement using hybrid approach," 2017 International Conference on Computing, Communication and Automation (ICCCA), pp. 647-652, 2017, doi: 10.1109/CCAA.2017.8229881.

- [104] D. Punia and B. Singh, "Speed Optimization of the AES Algorithm Using Pipeline Hardware Architecture," 2019 International Conference on Communication and Electronics Systems (ICCES), pp. 2070-2074, 2019, doi: 10.1109/ICCES45898.2019.9002086.
- [105] Q. Liu, Z. Xu and Y. Yuan, "A 66.1 Gbps single-pipeline AES on FPGA," 2013 International Conference on Field-Programmable Technology (FPT), pp. 378-381, 2013, doi: 10.1109/FPT.2013.6718392.
- [106] C. H. Kim, "Improved Differential Fault Analysis on AES Key Schedule," in IEEE Transactions on Information Forensics and Security, vol. 7, no. 1, Feb. 2012, pp. 41-50, doi: 10.1109/TIFS.2011.2161289.
- [107] C. E. Shannon, "Communication theory of secrecy systems," in The Bell System Technical Journal, vol. 28, no. 4, Oct. 1949, pp. 656-715, doi: 10.1002/j.1538-7305.1949.tb00928.x.
- [108] S. Chow, P. Eisen, H. Johnson, and P. C. Van Oorschot, "White-box cryptography and an AES implementation," in Selected Areas in Cryptography: 9th Annual International Workshop, SAC 2002 St. John's, Newfoundland, Canada, August 15–16, 2002 Revised Papers, vol. 9, pp. 250-270, Springer Berlin Heidelberg, 2003.
- [109] M. A. Murillo-Escobar, C. Cruz-Hernández, F. Abundiz-Pérez, R. M. López-Gutiérrez, and O. A. Del Campo, "A RGB image encryption algorithm based on total plain image characteristics and chaos," Signal Processing, vol. 109, pp. 119-131, 2015.
- [110] M. Kumar, D. C. Mishra, and R. K. Sharma, "A first approach on an RGB image encryption," Optics and Lasers in Engineering, vol. 52, pp. 27-34, 2014.
- [111] D. S. Lee, M. Wirthlin, G. Swift and A. C. Le, "Single-Event Characterization of the 28 nm Xilinx Kintex-7 Field-Programmable Gate Array under Heavy Ion Irradiation," 2014 IEEE Radiation Effects Data Workshop (REDW), Paris, France, pp. 1-5, 2014, doi: 10.1109/REDW.2014.7004595.
- [112] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "A Low-Power High-Performance Concurrent Fault Detection Approach for the Composite Field S-Box and Inverse S-Box," in IEEE Transactions on Computers, vol. 60, no. 9, pp. 1327-1340, Sept. 2011, doi: 10.1109/TC.2011.85.

## **APPENDIX A (LIST OF PUBLICATIONS)**

### **DETAILS OF PUBLICATIONS**

Below is the list of published research article in SCOPUS indexed conference proceedings along with the proofs of publications/acceptance.

1. J. Kala, J. Panda and L. Tanwar, "FPGA Implementation of a High Throughput Low Power Advanced Encryption Standard (AES-128) Cipher," 2023 10th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 2023, pp. 367-372, doi: 10.1109/SPIN57001.2023.10116674. **(IEEE XPLORE)**
2. J. Kala, J. Panda and L. Tanwar, "Digital Image Encryption Using 256-Bit Advanced Encryption Standard Algorithm" 2023 International Conference on Advancement in Computation & Computer Technologies (ICACCT) Chandigarh University Mohali, Punjab India, 2023 **(Accepted & Presented, yet to be Published – IEEE-XPLORE)**

# FPGA IMPLEMENTATION OF A HIGH THROUGHPUT LOW POWER ADVANCED ENCRYPTION STANDARD (AES-128) CIPHER

Jaideep kala  
Department of Electronics and  
Communication  
Delhi Technological University  
New Delhi, India  
jaideepkala\_2k21spd04@dtu.ac.in

Jeebananda Panda  
Department of Electronics and  
Communication  
Delhi Technological University  
New Delhi, India  
jpanda@dce.ac.in

Lavi Tanwar  
Department of Electronics and  
Communication  
Delhi Technological University  
New Delhi, India  
lavi.tanwar@dtu.ac.in

**Abstract**— Ensuring secure transmission and storage of digital information is critical for any organization to function properly. To address this issue, encryption algorithms are commonly used. Advanced Encryption Standard (AES) has been globally adopted as the mainstay cipher algorithm for securing transmission networks and storage devices due to its easy implementation and compatibility with both hardware and software applications. Another reason for its widespread popularity is its uncompromisable nature against existing brute force attacks, making it practically unbreakable on existing computing power. AES implementation for battery operated devices requires an algorithm with low power consumption and high-speed encryption/decryption of digital data. This paper proposes an FPGA implementation of a high throughput parallel pipelined 128-bit AES algorithm with a low power key expansion mechanism for iterative stages. A 128-bit symmetric key has been used for undertaking 10 rounds of transformations. All the encryption and decryption transformations are simulated using iterative design methodology in order to minimize hardware consumption. Xilinx Artix-7 FPGA device is used for hardware evaluation and Verilog HDL for programming. Simulation and synthesis task has been performed on Xilinx Vivado v2021.1 IDE. The results exhibit high-rate encryption of 68 Gb/s and low energy consumption of 7 pJ/bit.

**Keywords**— AES, Encryption, FPGA, Low Power, Secure, Simulation, Throughput.

## I. INTRODUCTION

Information security is a critical part of any communication system. At the heart of this security system is a cryptographic algorithm that manipulates the input message or plain text into cipher text with the help of a key [1]. An Encryption algorithm ensures that data is safely transmitted from sender to the receiver without any unauthorized manipulations or attacks. Advanced encryption standard (AES) is widely used encryption algorithm that has been adopted globally by governments and organizations for secure communication and data storage. Developed by National Institute of Standards and Technology in the year 2001, AES is a symmetric key cipher with a fixed block size of 128 bits. Meaning it uses the same key for encryption and decryption process and all operations are performed on 128 bits of data [2]. AES is the successor of Data encryption standard (DES) algorithm which was prone to hacking hence needed to be replaced by a much stronger algorithm. AES utilizes cipher key of sizes 128,192 and 256 bits for encryption hence making it one of the most secure and robust algorithms. It has proven to be safe from brute force attacks with not a single registered case of its failure against known attacks. Encryption techniques like AES are based on Shannon's theory of confusion and diffusion (1945) [3]. Here confusion aims at complicating the relationship between cipher message and symmetric key whereas diffusion aims at dispersing the

features of input message throughout the encrypted message. It can be efficiently used for both hardware and software applications One of the major limitations of AES algorithm is its high computational complexity which leads to high power consumption, making it less suitable for battery operated devices. Another area for improvement in AES is the speed of the algorithm [4]. Faster encryption and decryption processes are highly desirable for real time communication and data storage. For hardware implementation, AES algorithms that takes less area for implementation, has high throughput and low power consumption are preferable and is a topic of constant research [5].

## II. LITERATURE SURVEY

There have been significant efforts in the past aimed at reducing the power consumption and increasing throughput of the AES algorithm. Bui et al. [6] proposed a hardware optimization strategy for AES implementation in ultra-low power IoT applications to provide multilevel security using different key sizes. Power and energy optimization has been performed for both data path and key expansion. This led to significant reduction in energy per bit to a value of 1 pJ/b at 10 MHz at 0.6 V and throughput of 28 Mb/s. Duran et al. [7] proposed an AES-128/256 S-box acceleration scheme which uses a custom S-box unit connected as a logic unit. All S-box calculations were performed using pipelined and pure combinational approach resulting in lower memory access and lower energy consumption of 9.7 pJ/bit. A large portion of the energy consumed in an AES circuit is during the substitution process, hence S-box architecture plays a crucial role. Morioka et al. [8] proposed a low power S-box architecture in which signal arrival time at gates are very close if their depth from main input is identical. This led to a minimalistic power consumption of 29  $\mu$ W at 10 MHz using 0.13  $\mu$ m CMOS technology. In recent studies, pipelining of AES architecture has proven to significantly increase the throughput of the encryption system and accompanied power reduction. Chellappa et al. [9] proposed a fully pipelined 256-bit AES design with pulse clocked latches connecting the pipeline stages that can be made transparent when in use resulting a 7.6% decrease in energy. This design could deliver 64 Gb/s encryption when fabricated on 90 nm technology. Oukili et al. [10] presented a 5-stage pipeline S-box design to increase the maximum speed and frequency of the AES system. S-box transformations plays a crucial role in the complexity of AES algorithm therefore parallel processing using pipelined stages helped the proposed method to achieve a throughput of 79 Gbps. Kshirsagar et al. [11] proposed interchanging of byte substitution and shift rows operations in the AES implementation which helped to streamline the processing of 16 data blocks into 4 parallel blocks of data. This led to significant reduction in hardware area consumption by 56%

## DIGITAL IMAGE ENCRYPTION USING 256-BIT ADVANCED ENCRYPTION STANDARD ALGORITHM

Jaideep kala  
Department of Electronics and  
Communication  
Delhi Technological University  
New Delhi, India  
jaideepkala\_2k21spd04@dtu.ac.in

Jeebananda Panda  
Department of Electronics and  
Communication  
Delhi Technological University  
New Delhi, India  
jpanda@dce.ac.in

Lavi Tanwar  
Department of Electronics and  
Communication  
Delhi Technological University  
New Delhi, India  
lavi.tanwar@dtu.ac.in

**Abstract**— Ensuring secure communication of multimedia messages is crucial for social networking and data sharing platforms. Prevention of data manipulation and theft has led to the development of various encryption techniques, but scope remains for a fast and efficient multi-media encryptor. Advanced Encryption Standard (AES) is mathematically one of the most complex cipher algorithms to crack and has been widely deployed in the banking sector. This paper aims to present a high throughput implementation of 256-bit AES cipher for encrypting digital images and explore its practicality in peer-to-peer communication. Pre-processing of images has been performed to make them suitable for encryption. A detailed study of the encryption results and histogram analysis has been carried out. The proposed algorithm achieved a Peak Signal to Noise Ratio (PSNR) of 61 dB for the decrypted image. Correlation between the input and the decrypted image was found to be 0.994 while the Mean Square Error (MSE) was calculated to be 0.0030.

**Keywords**— Advanced Encryption Standard, Decryption, Images, Key, Secure, Simulation.

### I. INTRODUCTION

Cyber-attacks and data theft are a major concern for companies worldwide. Securing messages containing digital images and videos for real time communication can be a difficult and computationally expensive task. AES has been widely used for secure storage and transmission of digital information but its application in real time image/video encryption has been limited [1]. Developed in the year 2001 by National Institute of Standards and Technology, AES is a cryptographic algorithm that uses a symmetric public key for undertaking encryption and decryption tasks [2]. It is robust to existing brute force attacks and has been widely used by financial institutions and government agencies to carry out secure transactions and sharing of sensitive information.



AES processes 128-bit packets of data, although it can have a key size of 128,192 or 256 bits [3]. AES-256 is the most secure form of AES encryption and consists of 14 rounds of iterations for manipulating data into an unrecognizable form [4]. Encryption of multimedia messages using existing cipher technologies requires high bandwidth and large latencies are observed in peer-to-peer communication [5]. A high throughput encryption technique coupled with modules for pre, and post processing of images is needed for solving this issue and have a vast application in social networking / message sharing apps. AES can operate in 5 different modes, most common among these are Cipher Block Chaining, Electronic Code Book [6]. ECB is a widely used mode as it does not require an Initialization Vector (IV) for its

operations. CBC requires IV which may lead to propagation of error through the encryption stages [7]. The remaining part of this paper is presented as follows. Section II shows the related work carried out in this field and contribution of this paper. Section III discusses the proposed methodology and steps involved in image encryption. Section IV presents the results of encrypted and decrypted images along with their histogram analysis. Section V concludes the work carried out in this paper along with some limitations and future scope.

### II. LITERATURE SURVEY

There have been significant efforts in the past aimed at utilizing AES algorithm for encrypting alphanumeric messages, but very limited work has been carried out for multimedia messages. Hajihassani et al. [8] proposed a high throughput AES architecture based on bit slicing, which processed 32 128-bit data stream parallelly to provide an encryption throughput of 1.47 Tbps. This method performs expensive Byte transformations using shifting and swapping of registers. Kim et al. [9] utilized the CTR mode in AES for fast encryption in low-end microcontrollers. This led to a reduced computation of 2 Add Round Key, 2 Shift-Rows, 2 Substitution Bytes, and 1 Mixed Column transformation. Alomari et al. [10] compared different encryption techniques for storage devices for different operating modes. A detailed performance analysis was done to provide guidelines for disk encryption. Le et al. [11] proposed a fast GPU parallel computing design for AES cryptography. This technique accelerates the speed of AES encryption significantly. A novel low latency FPGA based AES architecture was proposed by Zhang et al. [12] in which an efficient key expansion method along with pipelining was used to obtain a throughput of nearly 21.56 Gbps. Custom S-box techniques are implemented for low energy consumption applications for AES designed for battery operated devices [13,14]. Pipelined AES architectures proposed by Chellappa et al. [15] and Oukili et al. [16] provided high throughput encryption of up to 64-79 Gbps.

For encryption of images, Zhang et al. [17] performed MATLAB implementation of AES-128 followed by digital image processing to obtain the encrypted and recovered test image. AES in CBC mode was utilized for image encryption [18] and its security performance was tested and compared with existing systems based on chaos. A similar system was proposed by Arab et al. [19] for novel image encryption using chaos sequence and an improvised AES-128. This approach reduces computational time and increases the diffusion ability of the proposed scheme. Singh et al. [20] proposed a dynamic AES developed using key dependent S-box for image encryption. Contributions of this paper:

**AMITY UNIVERSITY**  
UTTAR PRADESH

Department of Electronics & Communication Engineering  
Amity School of Engineering & Technology


10<sup>th</sup> International Conference on Signal Processing and Integrated Networks


**SPIN<sup>2023</sup>**  
23-24 March, 2023


**Certificate of Participation**

This is to certify that **Mr./Ms./Dr./Prof. Jaideep Kala** of **Delhi Technological University, India** has presented his/her paper (online) entitled **FPGA IMPLEMENTATION OF A HIGH THROUGHPUT LOW POWER ADVANCED ENCRYPTION STANDARD (AES-128) CIPHER** at the **10<sup>th</sup> International Conference on Signal Processing and Integrated Networks (SPIN 2023)** held on 23-24 March, 2023 in Hybrid Mode at Amity University, Noida, India.

This certificate is awarded for his/her valuable contribution in the success of SPIN 2023.

  
**Dr. Pradeep Kumar**  
Professor & Deputy Head  
Dept of ECE, ASET, AUUP  
Organizing Chair (SPIN-2023)

  
**Dr. J. K. Rai**  
Professor & Head,  
Dept of ECE, ASET, AUUP  
Conference Chair (SPIN-2023)  
24/03/2023

  
**Dr. Manoj Kumar Pandey**  
Professor & Joint Head,  
ASET, AUUP  
General Chair (SPIN-2023)





## Certificate of Appreciation

Jaideep kala

This is to certify that Prof./ Dr./ Mr./ Ms. \_\_\_\_\_  
of \_\_\_\_\_ **Delhi Technological University, New Delhi, India** \_\_\_\_\_ has  
been awarded for **Best Paper** titled \_\_\_\_\_  
**DIGITAL IMAGE ENCRYPTION USING 256-BIT ADVANCED**  
**ENCRYPTION STANDARD ALGORITHM**  
in **1st International Conference on Advancement in Computation & Computer Technologies (InCACCT-2023)** organized by the Department of Computer Science & Engineering, with the technical sponsor **IEEE Delhi Section (IEEE Conference Record No.: 57535X)** held on **05th – 06th May 2023** at Chandigarh University, Gharuan, Mohali, Punjab, India.

**Dr. Meenu Gupta**  
Convener & Conf. Organizing Chair  
Chandigarh University, Punjab, India

**Prof. (Dr.) Rakesh Kumar**  
Convener & Conf. Organizing Chair  
AD-CSE, Chandigarh University, Punjab, India



Sr. No. 142

## Certificate of Participation

JAIDEEP KALA

This is to certify that Prof./ Dr./ Mr./ Ms. \_\_\_\_\_  
of \_\_\_\_\_ **Delhi Technological University** \_\_\_\_\_  
*participated/ presented* a paper titled \_\_\_\_\_  
**Digital image encryption using 256-bit advanced encryption standard algorithm**

in **1st International Conference on Advancement in Computation & Computer Technologies (InCACCT- 2023)** organized by the Department of Computer Science & Engineering, with the technical sponsor **IEEE Delhi Section (IEEE Conference Record No.: 57535X)** held on **05th – 06th May 2023** at Chandigarh University, Gharuan, Mohali, Punjab, India.

**Dr. Meenu Gupta**  
Convener & Conf. Organizing Chair  
Chandigarh University, Punjab, India

**Prof. (Dr.) Rakesh Kumar**  
Convener & Conf. Organizing Chair  
AD-CSE, Chandigarh University, Punjab, India

## APPENDIX B (PLAGIARISM REPORT)



Similarity Report ID: oid:27535:36241550

PAPER NAME

**Major Project Thesis - Jaideep Kala - 2K  
21SPD04**

AUTHOR

**Jaideep Kala**

WORD COUNT

**8350 Words**

CHARACTER COUNT

**49665 Characters**

PAGE COUNT

**41 Pages**

FILE SIZE

**1.8MB**

SUBMISSION DATE

**May 26, 2023 2:51 PM GMT+5:30**

REPORT DATE

**May 26, 2023 2:52 PM GMT+5:30**

### ● 6% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 3% Internet database
- Crossref database
- 5% Submitted Works database
- 1% Publications database
- Crossref Posted Content database

### ● Excluded from Similarity Report

- Bibliographic material
- Cited material

## ● 6% Overall Similarity

Top sources found in the following databases:

- 3% Internet database
- Crossref database
- 5% Submitted Works database
- 1% Publications database
- Crossref Posted Content database

### TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	<b>German University of Technology in Oman on 2022-12-20</b>	<1%
	Submitted works	
2	<b>Mar Baselios College of Engineering and Technology on 2015-08-14</b>	<1%
	Submitted works	
3	<b>National College of Ireland on 2022-05-21</b>	<1%
	Submitted works	
4	<b>Visvesvaraya Technological University on 2014-11-27</b>	<1%
	Submitted works	
5	<b>ebin.pub</b>	<1%
	Internet	
6	<b>University College Birmingham on 2023-05-19</b>	<1%
	Submitted works	
7	<b>Vels University on 2019-02-18</b>	<1%
	Submitted works	
8	<b>mrcet.com</b>	<1%
	Internet	

9	<b>CollegeAmerica Services, Inc. on 2015-08-06</b> Submitted works	<1%
10	<b>KIIT International School on 2023-03-12</b> Submitted works	<1%
11	<b>University of Mosul on 2022-11-24</b> Submitted works	<1%
12	<b>American InterContinental University on 2023-04-22</b> Submitted works	<1%
13	<b>Leeds Beckett University on 2022-05-16</b> Submitted works	<1%
14	<b>doi.org</b> Internet	<1%
15	<b>ijsrd.com</b> Internet	<1%
16	<b>City University of Hong Kong on 2007-03-06</b> Submitted works	<1%
17	<b>Manchester Metropolitan University on 2023-05-19</b> Submitted works	<1%
18	<b>researchgate.net</b> Internet	<1%
19	<b>University of Maryland, University College on 2021-09-07</b> Submitted works	<1%
20	<b>Kennedy-Western University on 2003-02-14</b> Submitted works	<1%



Similarity Report ID: oid:27535:36241550

21	<b>SUNY, Binghamton on 2018-04-11</b> Submitted works	<1%
22	<b>core.ac.uk</b> Internet	<1%

