IMPROVED ADVANCED ENCRYPTION STANDARDS(AES) ALGORITHM

Project Report submitted in partial fulfillment of the Requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

D. LOKESH	**(1011902017)**
K. MADHUSUDHAN	**(1011902018)**
K. SASIREKHA	**(1011902030)**
G. DEVENDRA	**(1011902008)**
L. PRAVEEN KUMAR	**(1012002906)**

Under the Esteemed Guidance of

Sri. T. Mukthar Ahamed M.Tech,(Ph.D)

Academic Consultant

Department of Computer Science and Engineering

Y.S.R ENGINEERING COLLEGE OF YOGIVEMANA UNIVERSITY

PRODDATUR-516360, Y.S.R (Dt.), A.P.

(2022-2023)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Y.S.R. ENGINEERING COLLEGE OF YOGI VEMANA UNIVERSITY

PRODDATUR -516360, Y.S.R. (Dt.), A.P.

CERTIFICATE

This is to certify that the project report entitled " **IMPROVED ADVANCED ENCRYPTION STANDARD (AES) ALGORITHM** " is submitted by D LOKESH, K MADHUSUDHAN, K SASIREKHA, G DEVENDRA, L PRAVEEN KUMAR in partial fulfillment of the requirement for the award of the Degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING, Y.S.R. ENGINEERING COLLEGE OF YOGI VEMANA UNIVERSITY,PRODDATUR , is a record of bonafide work carried out by them under my guidance and supervision

PROJECT GUIDE **PROJECT CO-ORDINATOR** **HEAD OF THE DEPARTMENT**

Examiners:

1. **2.**

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without mention of the people, who made it possible, whose constant guidance and encouragement crowned our efforts with success. We take this opportunity to express my deepest gratitude and appreciation to all those who have helped us directly or indirectly towards the successful completion of this project.

It is great pleasure in expressing deep sense of gratitude and veneration to our guide **SRI. T. MUKTHAR AHAMED ,** Academic consultant, in the Department of Computer Science and Engineering for his valuable guidance and thought provoking discussion throughout the course of project work.

We extend our profound gratefulness to our **Dr. R. PRADEEP KUMAR REDDY**, Associate professor and Head of the Computer Science and Engineering department for his encouragement and support throughout the project and also our project coordinator **Dr. R. PRADEEP KUMAR REDDY**, Associate Professor, in the Department of Computer Science and Engineering who helped us a lot in achieving success of the project.

We take this opportunity to offer gratefulness to our Hon'ble Vice-Chancellor **Prof. C. SUDHAKAR GARU**, Dean of Engineering **Prof. K. VENKATA RAMANAIAH**, our Principal **Prof. C. NAGARAJU** and for providing all sorts of environment during the project work.

We express our gratitude to **Dr. S. KIRAN** , Associate Professor ,in the Department of Computer Science and Engineering, for support and putting up the extra time to reach our targets, which was greatly helpful in the progression and smoothness of the entire project work.

We express our thanks to all our college teaching and non-teaching staff members who encouraged and helped us in some way or other throughout the project work.

D LOKESH	**(1011902017)**
K MADHUSUDHAN	**(1011902018)**
K SASIREKHA	**(1011902030)**
G DEVENDRA	**(1011902008)**
L PRAVEEN KUMAR	**(1012002906)**

LIST OF CONTENT

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The Advanced Encryption Standard (AES) is a widely-used symmetric encryption algorithm that provides secure and efficient data encryption. However, there is always a need to improve the security and performance of cryptographic algorithms to meet the increasing demands of secure communication and data protection. In this project a modified AES encryption algorithm is proposed using an additional AddRoundKey operation. The additional AddRoundKey operation in between ShiftRows and the MixColumns operation, which can provide an additional level of diffusion and improve the security of the encryption. The modified AES encryption algorithm is tested against standard AES encryption algorithm using a variety of test vectors. Test results shows that the modified AES algorithm has better Avalanche effect compared to the standard AES algorithm. The modified AES algorithm provides improved diffusion and confusion properties than standard AES encryption algorithm, which can make it more secure against certain types of attacks.

Keywords: *Standard AES , Modified AES, AddRoundKey, Avalanche effect, Diffusion, confusion.*

CHAPTER - 1

INTRODUCTION

CHAPTER-1

INTRODUCTION

1.1 Introduction

Cryptography is the practice of securing information by transforming it into a form that is unintelligible to unauthorized individuals. It is a field of study that focuses on techniques for ensuring confidentiality, integrity, and authenticity of data. Cryptography plays a crucial role in various aspects of modern life, such as securing communications, protecting sensitive information, and enabling secure transactions over the internet. The primary goal of cryptography is to provide secure communication in the presence of adversaries. It achieves this goal through the use of mathematical algorithms and methods. These algorithms manipulate the original data, known as plaintext, to produce encrypted data, known as ciphertext. The process of converting plaintext into ciphertext is called encryption, and the reverse process is called decryption.

Cryptography relies on two fundamental types of algorithms: symmetric key algorithms and asymmetric key algorithms (also known as public-key algorithms). Symmetric key algorithms use the same key for both encryption and decryption, which means that the sender and the receiver must share a secret key. Asymmetric key algorithms, on the other hand, use a pair of mathematically related keys: a public key for encryption and a private key for decryption. The public key can be freely distributed, while the private key must be kept secret. Cryptography is used in various applications, including secure communication protocols (such as SSL/TLS), virtual private networks (VPNs), secure email, online banking, secure file storage, and digital rights management (DRM). It is a complex and evolving field that continues to play a vital role in securing sensitive information in our increasingly digital world.

Confidentiality: One of the primary goals of cryptography is to ensure confidentiality. This means that only authorized parties should be able to access and understand the information being transmitted or stored. Cryptographic techniques achieve confidentiality by transforming plaintext data into ciphertext using encryption algorithms. The encrypted data can only be decrypted back into its original form by individuals possessing the correct decryption key. This ensures that even if intercepted or accessed by unauthorized entities, the encrypted data remains unreadable and confidential.

Integrity: In addition to ensuring confidentiality, cryptography also addresses data integrity, which refers to the assurance that information remains unaltered and tamper-proof during transmission or storage. Data integrity is crucial in preventing unauthorized modifications or alterations to the data, which could lead to erroneous or malicious outcomes. Cryptographic techniques, such as hash functions and message authentication codes (MACs), play a key role in maintaining data integrity. Hash functions generate unique digital fingerprints or hash values for the data, which act as a "checksum" representing the data's integrity. Even a slight modification to the data will result in a different hash value, making it easy to detect tampering. Similarly,

MACs provide integrity protection by generating a cryptographic tag or signature that is securely attached to the data. Upon receiving the data, the recipient can verify the integrity of the data by recomputing the hash or verifying the MAC. By employing these cryptographic mechanisms, data integrity is preserved, ensuring the accuracy, reliability, and trustworthiness of the information being transmitted or stored.

Authentication: Authentication is a critical aspect of cryptography, ensuring the verification of the identity of individuals or entities involved in communication. It provides assurance that the sender or receiver of information is indeed the claimed party. Cryptographic mechanisms, such as digital signatures and public key infrastructure (PKI), play a vital role in authentication. Digital signatures use asymmetric key pairs, where the sender signs the data with their private key, and the receiver verifies the signature using the sender's public key. This process guarantees the integrity and authenticity of the data, as any modification to the data would invalidate the signature. PKI, on the other hand, relies on a trusted third party, a certificate authority (CA), to verify the identity of entities through the issuance of digital certificates. These certificates bind the public key to the identity of the entity, enabling secure authentication in various applications. By employing cryptographic authentication mechanisms, organizations can establish trust, prevent unauthorized access, and ensure secure communication between parties in a digital environment.

Non-repudiation: Non-repudiation is an important concept in cryptography, ensuring that the sender of a message cannot deny their involvement or claim that they did not send the message. It provides evidence and accountability for digital transactions, preventing parties from later refuting their actions. Cryptographic mechanisms such as digital signatures are commonly used to achieve non-repudiation. A digital signature is generated using the sender's private key and can be verified by anyone using the corresponding public key. By attaching a digital signature to a message or a transaction, the recipient can verify the sender's identity and integrity of the message, while the sender cannot deny their involvement. Non-repudiation is particularly important in legal and financial contexts, where it is necessary to establish proof of actions or agreements. By employing cryptographic techniques for non-repudiation, organizations can ensure the authenticity and trustworthiness of digital transactions and communications, reducing disputes and providing strong evidence in case of disputes or legal proceedings.

Key management: Key management is a critical aspect of cryptography that encompasses the generation, distribution, storage, and revocation of encryption keys. The security of cryptographic systems heavily relies on the secrecy and integrity of the keys involved. Proper key management practices are essential to ensure the confidentiality and integrity of encrypted data. Key generation involves the use of secure random number generators to create strong keys that are resistant to brute-force attacks. Distribution of keys must be done securely, ensuring that only authorized parties receive the keys and preventing interception or unauthorized access. Secure key storage mechanisms, such as hardware security modules (HSMs) or key management systems, are used to protect keys from unauthorized disclosure or tampering. Key revocation is crucial when keys are compromised or when individuals or entities no

longer have authorization to access the encrypted data. Regularly rotating keys and following secure key destruction procedures are important to maintain the security of cryptographic systems. Effective key management practices and protocols should be established, documented, and enforced within organizations to minimize the risk of key-related vulnerabilities and ensure the overall security of encrypted information.

1.2 History of Cryptography

The history of cryptography dates back thousands of years, with evidence of its use found in ancient civilizations. The history of cryptography showcases the continuous interplay between encryption and decryption techniques, with each advancement leading to new challenges and innovations. Cryptography remains a dynamic field, adapting to technological advancements and the evolving landscape of security threats.

key milestones and developments in the history of cryptography:

Ancient Cryptography: Cryptographic techniques were employed by ancient civilizations such as the Egyptians, Greeks, and Romans. One of the earliest known examples is the Caesar cipher, attributed to Julius Caesar, which involved shifting each letter of the alphabet by a certain number of positions.

Figure 1.1. Ancient Cryptography Caesar Cipher

Renaissance and Polyalphabetic Ciphers: In the Renaissance era, more advanced cryptographic techniques emerged. The Italian scholar Leon Battista Alberti introduced the polyalphabetic cipher, which used multiple alphabets to make encryption more secure. This innovation laid the foundation for future developments in cryptography.

The Vigenère Cipher: In the 16th century, the Vigenère cipher was invented by the French diplomat Blaise de Vigenère. It expanded upon Alberti's polyalphabetic cipher by using a keyword to determine the alphabets' sequence for encryption. The Vigenère cipher remained unbroken for several centuries.

Figure 1.2. Polyalphabetic Substitution Cipher

Cryptanalysis Advances: The 19th and early 20th centuries saw significant progress in cryptanalysis, the art of breaking codes. Notable cryptanalysts, such as Auguste and Louis Lumière, Charles Babbage, and Étienne Bazeries, developed techniques to crack various ciphers.

World War II and Enigma: During World War II, cryptography played a crucial role. The German Enigma machine, a complex electromechanical device, was used for encryption. However, British cryptanalysts at Bletchley Park, including Alan Turing, made significant breakthroughs in decrypting Enigma messages, contributing to the Allied victory.

Figure 1.3. Nazi Enigma machine

Modern Cryptography: The advent of computers in the mid-20th century led to significant advancements in cryptography. Symmetric key algorithms, such as the Data Encryption Standard (DES) and Advanced Encryption Standard (AES), became widely used for secure communications and data protection.

Public-Key Cryptography: In the 1970s, Whitfield Diffie and Martin Hellman introduced the concept of public-key cryptography, which revolutionized the field. Public-key algorithms, like the RSA algorithm developed by Ron Rivest, Adi Shamir,

and Leonard Adleman, enabled secure key exchange and digital signatures without requiring a shared secret key.

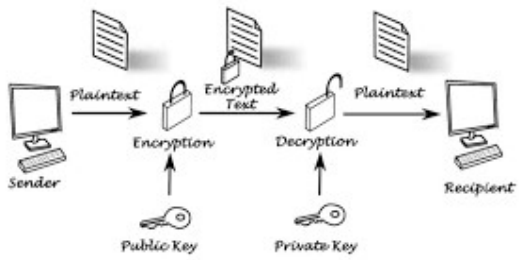


Figure 1.4. Public-key Cryptography

Internet Era: With the rise of the internet, cryptography became indispensable for securing online communications and transactions. Protocols like Secure Sockets Layer (SSL) and its successor Transport Layer Security (TLS) use cryptographic techniques to ensure secure connections.

Quantum Cryptography: In recent years, the emergence of quantum computers has posed challenges to classical cryptographic systems. Quantum cryptography, based on the principles of quantum mechanics, aims to provide secure communication resistant to quantum attacks.

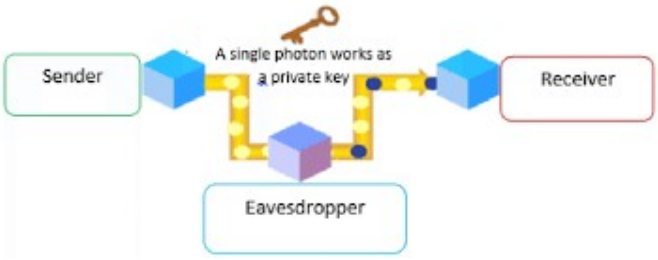


Figure 1.5. Quantum Cryptography

1.3 Overview of the AES algorithm

The Advanced Encryption Standard (AES) is a widely used symmetric key encryption algorithm. It was selected by the U.S. National Institute of Standards and Technology (NIST) in 2001 to replace the aging Data Encryption Standard (DES). AES has become the de facto standard for securing sensitive data and is employed in various applications worldwide.

Key features and components of the AES algorithm are as follows:

Symmetric Key Algorithm: AES is a symmetric key algorithm, meaning the same key is used for both encryption and decryption. This key is kept secret and must be shared securely between the communicating parties.

Block Cipher: AES operates on fixed-size blocks of data. The block size for AES is 128 bits (16 bytes). This means that the input data is divided into blocks of 128 bits, and each block is processed independently.

Key Lengths: AES supports key lengths of 128, 192, and 256 bits. The longer the key, the more secure the encryption. AES-128, AES-192, and AES-256 refer to the different key lengths used.

Substitution-Permutation Network: AES employs a substitution-permutation network (SPN) structure. It consists of multiple rounds (10 rounds for AES-128, 12 rounds for AES-19, and 14 rounds for AES-256), each performing a series of substitutions and permutations on the data.

Substitution: AES uses a substitution step known as the S-box. The S-box substitutes each byte of the data block with a corresponding value from a pre-defined lookup table. This non-linear substitution provides confusion and enhances security.

Permutation: AES employs a permutation step known as the ShiftRows operation. It cyclically shifts the rows of the data block, ensuring that the output of the encryption process has no simple patterns

Mixing: Another step in AES is the MixColumns operation. It operates on the columns of the data block, performing a mathematical mixing of the bytes to provide diffusion and further strengthen the encryption

Key Expansion: AES requires an expanded key derived from the original secret key. The key expansion process generates a set of round keys, one for each round of encryption. These round keys are derived using a combination of mathematical operations

Security: AES is considered highly secure, and no practical attacks have been discovered against the full AES algorithm. Its security is based on the key length and the complexity of the encryption process, including the non-linearity of the S-box and the diffusion provided by the mixing operations.

1.4 Applications of the AES algorithm

AES efficiency, security, and wide industry adoption make it an essential encryption algorithm for safeguarding sensitive information in various domains, including communication, storage, and digital content protection.

- Secure Communication Protocols: AES is a fundamental component of secure communication protocols such as Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL). AES ensures that data transmitted over networks, such as the internet, remains confidential and protected from unauthorized access.
- File and Disk Encryption: AES is commonly used for encrypting files and entire disks. File encryption applications utilize AES to encrypt individual files, ensuring their confidentiality and integrity. Disk encryption software, such as

BitLocker (Windows) and File Vault (Mac), use AES to encrypt entire storage devices, protecting the data stored on them.

- Virtual Private Networks (VPNs): AES is widely employed in VPNs, which provide secure connections between remote users or networks over the internet. AES ensures that the data transmitted between VPN endpoints remains encrypted and secure from eavesdropping and unauthorized access.
- Wireless Networks: AES is a part of the encryption standards used in securing wireless networks, such as Wi-Fi. The Wi-Fi Protected Access 2 (WPA2) protocol, which is based on AES, ensures the confidentiality and integrity of data transmitted over wireless networks, protecting against unauthorized access.
- Secure Email and Messaging: AES is utilized in secure email protocols, such as Pretty Good Privacy (PGP) and Secure/Multipurpose Internet Mail Extensions (S/MIME). It enables the encryption of email messages and attachments, ensuring that only the intended recipients can decrypt and read the contents.
- Cloud Storage and Backup: AES encryption is employed in cloud storage and backup services to protect data stored in the cloud. It ensures that data remains encrypted and secure, both during transit to the cloud and while at rest in cloud storage facilities.
- Database Encryption: AES is used in database systems to encrypt sensitive data stored in databases. This helps protect sensitive information, such as personal or financial data, from unauthorized access or theft.
- Digital Rights Management (DRM): AES encryption is utilized in DRM systems to protect copyrighted content from unauthorized copying or distribution. It ensures that digital media, such as music, movies, and e-books, can only be accessed by authorized users.

1.5 Limitations of the AES algorithm

While the Advanced Encryption Standard (AES) is widely regarded as a secure and efficient encryption algorithm, it does have some limitations that should be taken into consideration. One limitation is the need for proper key management. The security of AES heavily relies on the secrecy and integrity of the encryption keys. If the keys are compromised or mishandled, it can undermine the overall security of the encrypted data. Key generation, distribution, storage, and revocation must be carefully managed to maintain the confidentiality of AES-encrypted information. Organizations need to establish robust key management practices and protocols, including secure key storage mechanisms and procedures for key rotation and revocation. Additionally, the complexity of managing large numbers of encryption keys can present logistical challenges, requiring effective key management systems to be in place.

Another potential vulnerability lies in side-channel attacks. Side-channel attacks exploit information leaked through physical characteristics of the system, such as power consumption, electromagnetic radiation, or timing. By analyzing these side-channel signals, an attacker can potentially infer information about the encryption keys, compromising the security of AES. Countermeasures like constant-time implementations, which ensure that the execution time of cryptographic operations is independent of sensitive data, can help mitigate these risks. Implementing hardware

and software protections to minimize side-channel leakage, such as tamper-resistant modules or secure microcontrollers, can also enhance the resilience of AES against such attacks.

Figure 1.6 Side channel Attack

A significant limitation to consider is the potential threat posed by quantum computing. AES, in its current form, is not resistant to attacks by quantum computers. Quantum computers have the potential to break symmetric key algorithms, including AES, using algorithms like Shor's algorithm. This threat arises from the ability of quantum computers to perform certain calculations exponentially faster than classical computers. As quantum computing continues to advance, organizations should stay informed about the progress and developments in post-quantum cryptography. It is essential to evaluate and adopt post-quantum cryptographic algorithms that provide resistance against attacks from quantum computers. Migrating to these new algorithms may require significant changes to cryptographic infrastructure and protocols, as well as careful consideration of performance and compatibility issues.

Furthermore, AES, particularly when using longer key lengths like 192 or 256 bits, can require significant computational resources compared to shorter key lengths. Encrypting and decrypting large amounts of data using AES may be computationally expensive, especially in resource-constrained environments such as embedded systems or low-power devices. Organizations need to carefully evaluate the computational capabilities of their target systems and select appropriate key lengths based on the trade-off between security and computational efficiency. Optimizations such as hardware acceleration or algorithmic improvements can also be considered to mitigate the computational burden.

AES, in its basic form, does not provide authentication of the encrypted data. While AES ensures confidentiality by encrypting the data, it does not guarantee the integrity or authenticity of the data. If data integrity and authentication are crucial, additional mechanisms such as message authentication codes (MACs) or digital signatures should be used alongside AES to provide these security properties. MACs ensure data integrity by generating a tag that is securely attached to the encrypted data, allowing the recipient to verify the integrity of the received data. Digital signatures, on the other hand, provide

both integrity and authentication by adding a cryptographic signature to the data, ensuring its integrity and verifying the identity of the sender.

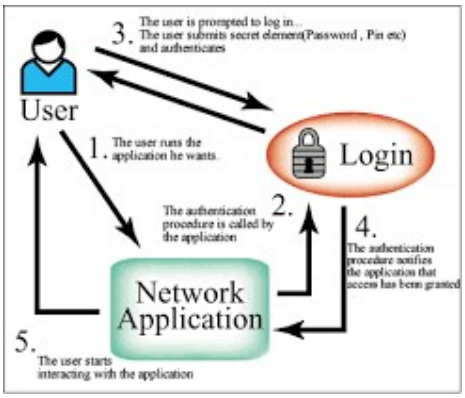


Figure 1.7 Authentication Mechanisms

Lastly, the security of AES can be compromised if there are vulnerabilities or weaknesses in the implementation of the algorithm. Implementation flaws, software bugs, or insecure key management practices can introduce vulnerabilities that attackers can exploit. It's crucial to ensure that AES implementations adhere to secure coding practices, undergo thorough security testing, and receive regular updates and patches to address any discovered vulnerabilities. Organizations should follow industry best practices, such as employing code reviews, performing penetration testing, and staying up to date

Despite these limitations, AES remains widely used and trusted for its robust security and performance. It's important to stay informed about ongoing research and advancements in cryptographic algorithms to address these limitations and adapt to emerging threats in the field of cryptography.

1.6 Cryptanalysis Techniques and Attacks on AES

Cryptanalysis is the science of analyzing cryptographic systems to uncover weaknesses and vulnerabilities that can be exploited to break the encryption. Although AES is widely regarded as a secure symmetric encryption algorithm, various cryptanalysis techniques have been developed to analyze its security. Brute-force attacks involve exhaustively trying all possible keys to decrypt the ciphertext, but the strength of AES lies in its key length, making brute-force attacks computationally infeasible. Differential and linear cryptanalysis are statistical techniques that exploit patterns and biases in the encryption process to deduce information about the key. Timing attacks aim to gather information about the key by measuring the time taken to perform encryption or decryption operations. Side-channel attacks, such as power analysis and electromagnetic analysis, focus on exploiting physical information leaked during the cryptographic operations, such as power consumption or electromagnetic radiation. While AES has undergone extensive scrutiny and analysis, it is crucial to

remain vigilant against new cryptanalysis techniques and continually evaluate its security to ensure its resistance to attacks.

Meet-in-the-Middle Attacks: Meet-in-the-middle attacks exploit the vulnerability of block ciphers like AES by breaking the encryption process into two stages. By precomputing the first half of the encryption process and the second half of the decryption process, an attacker can match the intermediate results to find the correct key. This attack can significantly reduce the time complexity of brute-forcing AES, especially when the key size is not utilized to its full extent.

Side-Channel Attacks: Side-channel attacks target the physical implementation of cryptographic systems rather than the underlying algorithms. They exploit information leaked through side channels such as power consumption, electromagnetic radiation, or even sound. By monitoring these side channels during cryptographic operations, an attacker can gather information about the internal states and secret keys used in AES. Countermeasures against side-channel attacks include techniques like power analysis resistance, electromagnetic shielding, and constant-time implementations.

Related-Key Attacks: Related-key attacks take advantage of weaknesses in AES when multiple keys are related to each other. In such attacks, the attacker gains access to pairs of plaintext and ciphertext encrypted under related keys and analyzes the relationship between them to deduce information about the key. These attacks are particularly relevant in scenarios where keys are derived from a common master key using weak key derivation functions.

Algebraic Attacks: Algebraic attacks leverage algebraic properties of the AES algorithm to uncover the secret key. By formulating AES as a set of algebraic equations, attackers can solve these equations to obtain the key. However, practical algebraic attacks on AES are currently limited due to the complexity of solving these equations, especially with the use of larger key sizes.

Fault Attacks: Fault attacks manipulate the execution of cryptographic operations to induce errors in the computation and exploit them to deduce information about the key. By injecting faults, such as voltage glitches or clock manipulations, into the system during encryption or decryption, an attacker can cause the algorithm to malfunction and reveal sensitive information. Countermeasures against fault attacks include error detection and correction techniques, redundancy mechanisms, and secure hardware implementations.

As of the current state of research, AES remains a robust and secure encryption algorithm. The cryptanalytic community has extensively analyzed AES since its adoption as the standard encryption algorithm in 2001. Despite numerous efforts, no practical attacks have been discovered against the full AES algorithm with its recommended key sizes. While some theoretical attacks, such as related-key attacks and algebraic attacks, have been proposed, they often require unrealistic assumptions or are not applicable to practical scenarios. The extensive scrutiny and absence of practical attacks have instilled confidence in the security of AES. However, the field of cryptanalysis is constantly evolving, and new techniques and advancements may

emerge in the future. Ongoing research aims to explore novel attack vectors, potentially exploiting implementation flaws or side-channel vulnerabilities. Therefore, it is essential to remain vigilant and continue to evaluate the security of AES through regular updates, advancements in cryptanalysis, and improvements in hardware and software implementations.

1.7 Modern Developments in Symmetric Key Algorithms

The field of symmetric key algorithms has seen significant developments in recent years, aimed at addressing specific requirements and emerging challenges. One area of advancement is the development of AES variants and modes of operation. These variants, such as AES-GCM (Galois/Counter Mode) and AES-XTS (XEX-based Tweaked CodeBook Mode with Cipher Text Stealing), provide enhanced functionalities and security properties for specific applications. AES-GCM, for instance, combines the AES encryption algorithm with a universal hash function for authenticated encryption, offering both confidentiality and data integrity in a single operation. AES-XTS is commonly used in disk encryption scenarios, providing protection against certain types of attacks.

Another noteworthy development in symmetric key algorithms is the emergence of lightweight encryption algorithms. These algorithms are designed to provide efficient and secure encryption in resource-constrained environments, such as embedded systems, IoT devices, and wireless sensor networks. Lightweight encryption algorithms prioritize small code size, low power consumption, and fast execution while maintaining acceptable levels of security. Examples of lightweight encryption algorithms include PRESENT, SIMON, and Speck, which have gained attention for their ability to provide robust encryption while minimizing computational and memory requirements.

In recent years, there has been a growing demand for encryption algorithms that can provide efficient and secure cryptographic solutions in resource-constrained environments. This demand has led to the development of lightweight encryption algorithms. Lightweight encryption algorithms prioritize small code size, low power consumption, and fast execution while maintaining acceptable levels of security. These algorithms are specifically designed to meet the requirements of embedded systems, Internet of Things (IoT) devices, and other constrained environments where computational and memory resources are limited. Examples of lightweight encryption algorithms include PRESENT, SIMON, and Speck. PRESENT is a lightweight block cipher that offers a balance between security and efficiency, making it suitable for applications with limited resources. SIMON and Speck are families of lightweight block ciphers that provide a range of key sizes and performance trade-offs, allowing users to choose the most appropriate configuration based on their specific needs. Lightweight encryption algorithms have gained significant attention due to their ability to provide secure encryption while minimizing computational and memory requirements. They enable secure communication and data protection in environments where traditional encryption algorithms may be too resource-intensive.

Furthermore, authenticated encryption schemes have gained prominence as a means to ensure both confidentiality and data integrity simultaneously. Authenticated encryption algorithms, such as GCM, CCM (Counter with CBC-MAC), and OCB (Offset CodeBook mode), offer built-in mechanisms for data authentication and encryption. These schemes protect against unauthorized modification of encrypted data and provide assurance that the decrypted data is authentic and unaltered.

The Advanced Encryption Standard (AES) has been widely adopted as a secure symmetric encryption algorithm. Over time, various AES variants and modes of operation have emerged to cater to different security requirements and application scenarios. One notable AES variant is AES-GCM (Galois/Counter Mode), which combines the AES encryption algorithm with a universal hash function. AES-GCM not only provides confidentiality but also ensures data integrity through the use of authenticated encryption. It has become a popular choice for securing communication channels and is widely used in protocols like Transport Layer Security (TLS). Another AES variant is AES-XTS (XEX-based Tweaked CodeBook Mode with Cipher Text Stealing), primarily employed in disk encryption applications. AES-XTS is designed to offer robust protection against certain types of attacks, including known-plaintext attacks and watermarking attacks. It divides the data into blocks and encrypts each block separately, making it suitable for encrypting data stored on disk or other storage media. These AES variants, along with other modes of operation like Cipher Block Chaining (CBC), Counter (CTR), and Electronic CodeBook (ECB), provide flexibility in terms of security features, performance trade-offs, and compatibility with different systems and protocols. By utilizing these AES variants and modes of operation, organizations can tailor their encryption solutions to meet specific requirements and ensure robust data protection.

Authenticated encryption schemes have emerged as an important cryptographic solution for ensuring both confidentiality and data integrity in secure communications. These schemes provide a combined approach where encryption and authentication are performed together, offering protection against unauthorized modification of encrypted data. Authenticated encryption schemes eliminate the need for separate encryption and message authentication codes (MACs), simplifying the implementation and reducing computational overhead. Several authenticated encryption schemes have been developed, including GCM (Galois/Counter Mode), CCM (Counter with CBC-MAC), and OCB (Offset CodeBook mode). GCM is widely used and recommended in various protocols, such as TLS, due to its efficiency and security. CCM combines counter mode encryption with a CBC-MAC to provide both confidentiality and authentication, making it suitable for resource-constrained environments. OCB is another efficient authenticated encryption mode that combines encryption with a proprietary message authentication mechanism. These authenticated encryption schemes ensure the integrity and authenticity of encrypted data, protecting against unauthorized modifications or tampering. They are essential in applications where data integrity is critical, such as secure communication channels, storage systems, and authenticated encryption of sensitive data.

These modern developments in symmetric key algorithms demonstrate the ongoing efforts to meet evolving security requirements, accommodate resource-constrained environments, and provide efficient solutions for secure communication and data protection. By expanding the range of cryptographic options available, these advancements contribute to the versatility and adaptability of symmetric key algorithms in various applications and computing environments.

1.8 Motivation for the Project

The motivation behind our project, "Improved AES Algorithm with Added Extra AddRoundKey in between ShiftRows and MixColumns," stems from the constant pursuit of enhancing the security and efficiency of cryptographic algorithms. AES (Advanced Encryption Standard) is a widely adopted symmetric encryption algorithm known for its robustness and effectiveness. However, as technology evolves and new cryptographic attacks emerge, it is essential to continuously improve and strengthen existing algorithms.

Our project aims to address a specific aspect of the AES algorithm by introducing an enhancement that inserts an additional AddRoundKey operation between the ShiftRows and MixColumns steps. The motivation for this modification is to explore the potential benefits it may bring to the overall security and performance of the AES algorithm.

By inserting an extra AddRoundKey operation at this specific stage, we seek to examine the impact on the diffusion and confusion properties of AES. The added operation can potentially introduce additional security against certain cryptanalytic techniques, enhancing the algorithm's resistance to attacks. Moreover, we anticipate that this modification may contribute to improved diffusion, resulting in a more efficient and effective encryption process.

Furthermore, our project not only aims to improve the security and performance of the AES algorithm but also aligns with the broader objective of advancing the field of cryptography as a whole. By exploring modifications and enhancements to existing encryption algorithms, we contribute to the collective knowledge and understanding of cryptographic techniques. Our work has the potential to inspire further research and development in this domain, igniting a cycle of innovation and improvement.

By sharing our findings and insights with the cryptographic community, we can foster collaboration and encourage the exploration of new ideas and approaches. The examination of modifications to AES, such as inserting an additional AddRoundKey operation, opens up avenues for experimentation and discovery. Other researchers and cryptographic experts may build upon our work, offering their perspectives and proposing alternative enhancements. This collaborative spirit allows for a deeper understanding of the underlying principles and mechanisms behind encryption algorithms, ultimately leading to more robust and secure cryptographic solutions.

Moreover, our project's potential impact extends beyond AES itself. The knowledge gained from our research can be applied to other encryption algorithms, guiding future iterations and improvements. As we push the boundaries of what is currently known and understood, we pave the way for new encryption techniques and protocols that can

address emerging security challenges. The potential impact of our project extends beyond theoretical research. Improved encryption algorithms have wide-ranging applications in various domains, including secure communication protocols, data storage, financial transactions, and sensitive information protection. By enhancing the security and performance of AES through the introduction of an additional AddRoundKey operation, we aim to provide a practical solution that can be integrated into real-world systems. The implications of our project could result in more secure data transmission, increased confidence in encryption mechanisms, and enhanced protection against unauthorized access or tampering.

Overall, the motivation behind our project is driven by the desire to enhance the security and performance of the AES algorithm. By introducing an extra AddRoundKey operation between ShiftRows and MixColumns, we aim to explore the potential benefits it brings to the diffusion and confusion properties of AES, contributing to the advancement of cryptographic techniques and furthering our understanding of secure encryption algorithms.

1.9 Overview of the Project

Our project, titled "Improved AES Algorithm with Added Extra AddRoundKey in between ShiftRows and MixColumns," aims to enhance the security and performance of the Advanced Encryption Standard (AES) algorithm. AES is a widely used symmetric encryption algorithm known for its strength and reliability. However, with the continuous advancements in computing power and cryptanalysis techniques, it is crucial to explore enhancements to further strengthen the algorithm.

The main objective of our project is to investigate the impact of inserting an additional AddRoundKey operation between the ShiftRows and MixColumns steps of the AES algorithm. This modification introduces an extra layer of confusion and diffusion, potentially improving the algorithm's resistance to cryptanalytic attacks. By evaluating the effects on security properties, such as diffusion, confusion, and resistance against attacks, we aim to assess the feasibility and effectiveness of this enhancement.

Our project involves a comprehensive analysis and evaluation of the modified AES algorithm. We will conduct rigorous testing, including cryptographic strength analysis, performance evaluation, and comparative studies with the original AES algorithm. Through extensive experimentation and analysis, we seek to determine the potential benefits and trade-offs introduced by this modification.

Additionally, our project involves implementing the improved AES algorithm and validating its performance and security characteristics. This includes assessing its suitability for various real-world applications, such as secure communication protocols, data storage systems, and sensitive information protection. We will also consider the compatibility and interoperability aspects of the modified AES algorithm with existing cryptographic infrastructures and protocols.

Throughout the project, we will document our research findings, methodologies, and experimental results. The project report will serve as a comprehensive resource, contributing to the body of knowledge in the field of cryptography. It will provide

insights into the effectiveness of the proposed modification, offering guidance for further research and development in this area.

In summary, our project aims to enhance the AES algorithm by introducing an additional AddRoundKey operation between the ShiftRows and MixColumns steps. Through rigorous analysis, testing, and evaluation, we seek to improve the algorithm's security and performance characteristics. Ultimately, our project contributes to the advancement of encryption techniques, with the potential to inspire further research and development in the field of cryptography.

CHAPTER - 2

LITERATURE SURVEY

CHAPTER -2

LITERATURE SURVEY

1.Ako Muhamad Abdullah discussed about Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data: Advanced Encryption Standard (AES) algorithm is one on the most common and widely symmetric block cipher algorithm used in worldwide. This algorithm has an own particular structure to encrypt and decrypt sensitive data and is applied in hardware and software all over the world. It is extremely difficult to hackers to get the real data when encrypting by AES algorithm. Till date is not any evidence to crake this algorithm. AES has the ability to deal with three different key sizes such as AES 128, 192 and 256 bit and each of these ciphers has 128-bit block size. This paper will provide an overview of AES algorithm and explain several crucial features of this algorithm in details and demonstration some previous researches that have done on it with comparing to other algorithms such as DES, 3DES, Blowfish etc.

2.Hasanen S. Abdullah , Maha A.Hamood Al-Rawi, Dalal N. Hammod have done Analysis of AES Algorithm Effects on the Diffusion Property: In cryptography, confusion and diffusion are two very important characteristics, which must be achieved these characteristics in the ciphertext to obtain a strong cipher and avoid any attacker for attacked. This research aims to propose three methods based on a different sequence of operations of Advanced Encryption Standard (AES) algorithm. In this research is used hamming distance ,which it is a number of different symbols between two strings of equal length, for calculating diffusion. The proposed methods obtained approximately (63 bits) change in each round corresponding to the total number of bits (128 bits), but the standard AES obtains approximately (65 bits). The proposed methods use hamming distance for calculated diffusion property the (49.5%) percentage value of the proposed methods and the (50.9%) percentage value of the standard AES. After testing and verifying, it was concluded the AES algorithm is the best sequence of operations to achieve the best confusion and diffusion of data.

3.S.Radhika, A.Chandra Sekar proposed AES Algorithm using 512 Bit Key Implemented for Secure Communication : The main aim of this paper is to provide stronger security for communication over the Internet by enhancing the strength of the

AES algorithm. Rijndael's algorithm was selected as the Advanced Encryption Standard. The AES algorithm was believed to provide much more security without any limitations. But, recently some breaking methods on the AES have been found by cryptanalysts. In AES algorithm, the number of rounds involved in the encryption and decryption depends on the length of the key and the number of block columns. So, the number of rounds is increased to improve the strength of the AES. The strength of the AES algorithm is enhanced by increasing the key length to 512 bit and thereby the number of rounds is increased in order to provide a stronger encryption method for secure communication. Code optimization is done in order to improve the speed of encryption and decryption using the 512-bit AES.

4. Abdullah Al- Mamun , Shawon S. M. Rahman , Tanvir Ahmed Shaon and Md Alam Hossain proposed security analysis of AES and enhancing its security by modifying s-box with an additional byte: Secured and opportune transmission of data always is a significant feature for any organization. Robust encryption techniques and algorithms always facilitate in augmenting secrecy, authentication and reliability of data. At present, Advanced Encryption Standard (AES) patronized by NIST is the most secure algorithm for escalating the confidentiality of data. This paper mainly focuses on an inclusive analysis related to the security of existing AES algorithm and aim to enhance the level security of this algorithm. Through some modification of existing AES algorithm by XORing an additional byte with s-box value, we have successfully increased the Time Security and Strict Avalanche Criterion. We have used random additional key for increasing security. Since this key is random, result of security measurement sometimes fluctuates.

5. Heidilyn V. Gamido, Ariel M. Sison, Ruji P. Medina proposed Modified AES for Text and Image Encryption: Advanced Encryption Standard (AES) is one of the most frequently used encryption algorithms. In the study, the Advanced Encryption Standard is modified to address its high computational requirement due to the complex mathematical operations in MixColumns Transformation making the encryption process slow. The modified AES used Bit Permutation to replace the MixColumns Transformation in AES since bit permutation is easy to implement and it does not have any complex mathematical computation. Results of the study show that the modified AES algorithm exhibited increased efficiency due to the faster encryption time and

reduced CPU usage. The modified AES algorithm also yielded higher avalanche effect which improved the performance of the algorithm.

6. Junjie Yan* and Feng Chen proposed An Improved AES Key Expansion Algorithm: Advanced Encryption Standard (AES) has been widely used in wireless communications with advantage of the small amount of computation and fast speed. In order to overcome the drawback of typical expansion algorithm whose key is easily attacked by Square, an improved AES algorithm is proposed. In this algorithm, the double S-box model, the only non-linear structure, is employed to increase the diffusivity of data. Experimental results indicate that this AES algorithm can improve the security of the key in the same condition of algorithm efficiency.

7. Jie Cui1,2 , Liusheng Huang , Hong Zhong , Chinchen Chang and Wei Yang proposed An Improved Aes S-Box and Its Performance Analysis: S-box is a unique nonlinear operation in Rijndael, one encryption algorithm chosen as AES, and it determines the performance of AES. In this paper, the weaknesses in complexity and security of AES S-box are analysed. We propose to increase the complexity and security of AES S-box by modifying the affine transformation and adding an affine transformation. Performance analysis demonstrates that the improved AES S-box has following cryptographic properties: the affine transformation period is increased from 4 to the most 16, the iterative period is increased from less than 88 to the most 256, and the distance to SAC is reduced from 432 to 372. Moreover, the number of terms in the improved AES S-box algebraic expression is increased from 9 to 255, and its inverse Sbox keeps almost the same as AES inverse S-box. Comparison results suggest that the improved AES S-box has better performance and can readily be applied to AES.

8. Priyanka Sharma proposed A New Image Encryption using Modified AES Algorithm and its Comparison with AES: In the present world optimisation is the only thing to be asked for. The relentless growth of Internet and Communication technologies has made the extensive use of images unavoidable. The specific characteristics of image like high transmission rate with limited bandwidth, redundancy, bulk capacity and correlation among pixels makes standard algorithms not suitable for image encryption. In order to overcome these limitations for real time applications, design of new algorithms that require less computational power while preserving a sufficient level of security has always been a subject of interest. This paper proposes an algorithm based on Modified

AES Key Expansion in which the encryption process is a bitwise exclusive or operation of a set of image pixels along with the a 128 bit key which changes for every set of pixels . The keys to be used are generated independently at the sender and receiver side based on Modified AES Key Expansion process hence the initial key is alone shared rather than sharing the whole set of keys. The algorithm has been experimented with standard bench mark images proposed in USC-SIPI database. Experimental results and security analysis of the proposed algorithm shows that the proposed algorithm offers good resistance against brute force attack, key sensitivity tests and statistical crypt analysis.

9. M. Zeghid, M. Machhout, L. Khriji, A. Baganne, and R. Tourki proposed A Modified AES Based Algorithm for Image Encryption: With the fast evolution of digital data exchange, security information becomes much important in data storage and transmission. Due to the increasing use of images in industrial process, it is essential to protect the confidential image data from unauthorized access. In this paper, we analyse the Advanced Encryption Standard (AES), and we add a key stream generator (A5/1, W7) to AES to ensure improving the encryption performance; mainly for images characterised by reduced entropy. The implementation of both techniques has been realized for experimental purposes. Detailed results in terms of security analysis and implementation are given. Comparative study with traditional encryption algorithms is shown the superiority of the modified algorithm.

10. Eman Mohammed Mahmoud, Ahmed Abd El Hafez, Talaat A. Elgarf, AbdelhalimZekry proposed Dynamic AES-128 with Key-Dependent S-box : Advanced Encryption Standard (AES) block cipher system is widely used in cryptographic applications. The main core of AES block cipher is the substitution table or SBox. This S-box is used to provide confusion capability for AES. The aim of this paper is to design dynamic S-box which depends on the secret key. The parameters of the new created SBOXes have characteristics equal to those in the original algorithm AES. This algorithm is suitable to exchange keys on insecure communication channels in order to achieve secure communications. In this paper, a dynamic AES-128 with key dependent S-box is designed and implemented. Also, the quality of the implemented S-boxes is experimentally investigated. Also, the designed AES is compared with original AES in

terms of security analysis, and simulation time. Key words-Advanced encryption standard (AES), dynamic S-box, S-box, security analysis.

11. Edjie M. De Los Reyes , Dr. Ariel M. Sison , Dr. Ruji P. Medina proposed Modified AES cipher round and key schedule: In this paper, Advanced Encryption Standard was modified to address the low diffusion rate at the early rounds by adding additional primitive operations such as exclusive OR and modulo arithmetic in the cipher round. Furthermore, byte substitution and round constant addition were appended to the key schedule algorithm. The modified AES was tested against the standard AES by means of avalanche effect and frequency test to measure the diffusion and confusion characteristics respectively. The results of the avalanche effect evaluation show that there was an average increase in diffusion of 61.98% in round 1, 14.79% in round 2 and 13.87% in round 3. Consequently, the results of the frequency test demonstrated an improvement in the randomness of the ciphertext since the average difference between the number of ones to zeros is reduced from 11.6 to 6.4 along with better-computed p-values. The results clearly show that the modified AES has improved diffusion and confusion properties and the ciphertext can still be successfully decrypted and recover back the original plaintext.

12. Ako Muhammad Abdullah et al., Proposed Nowadays, network has important roles for transferring data accurately and fast from source to a destination. The data is not secure enough to transfer highly confidential. The security of information has become one of the principle challenges of resource sharing with data communication over computer network. Cryptography and Steganography are two methods for protecting data from intruders while transferring over an open channel network. Cryptography is a method to encrypt data and steganography is the art and science of hiding secret message in a cover image. In this paper a Hash Least Significant Bit (H-LSB) with Affine cipher algorithm has been proposed for providing more security to data in a network environment. First we encrypt the data with the new cryptography algorithm and then embed in the image. Eight bits of the secret message are divided into 3, 3, 2 and embedding into the RGB pixels values of the cover image respectively. A hash function is used to select the particular position of insertion in LSB bits. This system allows a message sender to select keys to encrypt the secret message before embedding into the image and a receiver is used the keys to decrypt the message. Receiver can be

decrypted the encrypt message with incorrect the keys but to a different form from the original message. This system has the ability to provide better security while transferring the secret message from one end to the other end in network environment.

13. Singh, G. and Supriya proposed A study of encryption algorithms (RSA, DES, 3DES and AES) for information security: Encryption is the process of scrambling a message so that only the intended recipient can read it. Encryption can provide a means of securing information. As more and more information is stored on computers or communicated via computers, the need to insure that this information is invulnerable to snooping and/or tampering becomes more relevant. With the fast progression of digital data exchange in electronic way, Information Security is becoming much more important in data storage and transmission. Information Confidentiality has a prominent significance in the study of ethics, law and most recently in Information Systems. With the evolution of human intelligence, the art of cryptography has become more complex in order to make information more secure. Arrays of Encryption systems are being deployed in the world of Information Systems by various organizations. In this paper, a survey of various Encryption Algorithms is presented.

14. Lu, C. C., & Tseng, S. Y proposed Integrated design of AES (Advanced Encryption Standard) encrypter and decrypter: This paper proposed a method of integrating the AES encrypter and the AES decrypter into a full functional AES crypto-engine. This method can make it a very low-complexity architecture, especially in saving the hardware resource in implementing the AES (Inv)SubBytes module and (Inv)Mixcolumns module, etc. Most designed modules can be used for both AES encryption and decryption. Besides, the architecture can still deliver a high data rate in both encryption/decryption operations. The proposed architecture is suited for hardware-critical applications, such as smart cards, PDAs, and mobile phones, etc.

15. Nadeem, H described A performance comparison of data encryption algorithms: The principal goal guiding the design of any encryption algorithm must be security against unauthorized attacks. However, for all practical applications, performance and the cost of implementation are also important concerns. A data encryption algorithm would not be of much use if it is secure enough but slow in performance because it is a common practice to embed encryption algorithms in other applications such as e-commerce, banking, and online transaction processing applications. Embedding of

encryption algorithms in other applications also precludes a hardware implementation, and is thus a major cause of degraded overall performance of the system. In this paper, the four of the popular secret key encryption algorithms, i.e., DES, 3DES, AES (Rijndael), and the Blowfish have been implemented, and their performance is compared by encrypting input files of varying contents and sizes, on different Hardware platforms. The algorithms have been implemented in a uniform language, using their standard specifications, to allow a fair comparison of execution speeds. The performance results have been summarized and a conclusion has been presented. Based on the experiments, it has been concluded that the Blowfish is the best performing algorithm among the algorithms chosen for implementation.

CHAPTER - 3

EXISTING METHOD

CHAPTER -3

EXISTING METHOD

3.1 INTRODUCTION

As DES is widely used symmetric encryption algorithm but there is always a need to improve the security of cryptographic algorithms to meet the increasing demands of secure communication and data protection. A replacement for DES was needed because DES Key size is too small and DES is slow, so we can use Triple-DES – but slow and small block size. US NIST issued call for ciphers in 1997.15 candidates accepted in Jun 98 and 5 were shortlisted in Aug 99. AES (Advanced Encryption Standard) is a widely used symmetric encryption algorithm that is used to encrypt and decrypt electronic data. It was developed by the National Institute of Standards and Technology (NIST) in 2001 as a replacement for the older DES (Data Encryption Standard) algorithm. AES is a block cipher, which means it encrypts data in fixed-size blocks (128 bits in the case of AES). It uses a series of mathematical operations, including substitution, permutation, and bitwise operations, to transform the plaintext into ciphertext. AES supports three key sizes: 128-bit, 192-bit, and 256-bit. The key size determines the strength of the encryption, with larger key sizes providing stronger security. AES-256, which uses a 256-bit key, is currently the most widely used and recommended version of AES and is used in a wide range of applications, including secure communications, financial transactions, and data storage. It has become the de facto standard for encryption and is used by many organizations, governments, and security-conscious individuals around the world.

The Advanced Encryption Standard (AES) was published by NIST (National Institute of Standards and Technology) in 2001. AES is a symmetric block cipher that is intended to replace DES as the approved standard for a wide range of applications. The AES cipher (& other candidates) form the latest generation of block ciphers, and now we see a significant increase in the block size - from the old standard of 64-bits up to 128-bits; and keys from 128 to 256-bits. In part this has been driven by the public demonstrations of exhaustive key searches of DES. Whilst triple-DES is regarded as secure and well understood, it is slow, especially in s/w. In a first round of evaluation, 15 proposed

algorithms were accepted. A second round narrowed the field to 5 algorithms. NIST completed its evaluation process and published a final standard (FIPS PUB 197) in November of 2001. NIST selected Rijndael as the proposed AES algorithm. The two researchers who developed and submitted Rijndael for the AES are both cryptographers from Belgium: Dr. Joan Daemen and Dr. Vincent Rijmen.

3.2 RIJNDAEL CIPHER

Rijndael was selected as the AES in Oct-2000, designed by Vincent Rijmen and Joan Daemen in Belgium and issued as FIPS PUB 197 standards in Nov-2001.

Figure 3.1: Vincent Rijmen andJoan Daeman

An iterative rather than Feistel cipher: processes data as block of 4 columns of 4 bytes (128 bits) and operates on entire data block in every round, he Feistel Cipher is a symmetric encryption scheme that was invented by Horst Feistel in the early 1970s. It is widely used in modern cryptographic algorithms such as the Data Encryption Standard (DES) and its successor, the Advanced Encryption Standard (AES). The Feistel Cipher operates by dividing the plaintext into blocks and then repeatedly applying a round function that mixes the plaintext with a subkey derived from the main key. The output of each round is fed back into the next round as input, with the final output being the ciphertext. The Feistel Cipher has several desirable properties, including resistance to differential and linear cryptanalysis and the ability to easily encrypt and decrypt data in parallel. It also allows for the use of different types of round functions and subkey generation techniques, making it adaptable to a variety of encryption needs. Overall, the Feistel Cipher is a powerful and flexible encryption

scheme that has stood the test of time and remains an important tool in the field of cryptography.

Rijndael design has simplicity, has 128/192/256-bit keys, 128 bits data and resistant against known attacks and speed and code compactness on many CPU. Rijndael is a symmetric key block cipher that was selected as the Advanced Encryption Standard (AES) by the National Institute of Standards and Technology (NIST) in 2001. The design was created by two Belgian cryptographers, Joan Daemen and Vincent Rijmen, and was selected from a pool of five finalists in the AES competition. The Rijndael cipher operates on blocks of plaintext and produces blocks of ciphertext of the same size.

The block size and the key length can vary, but the most common combinations are 128-bit block size with key lengths of 128, 192, or 256 bits. The Rijndael proposal for AES defined a cipher in which the block length and the key length can be independently specified to be 128,192, or 256 bits. The AES specification uses the same three key size alternatives but limits the block length to 128 bits. Rijndael is an academic submission, based on the earlier square cipher, from Belgium academics Dr Joan Daemen and Dr Vincent Rijmen. It is an iterative cipher (operates on entire data block in every round) rather than Feistel (operate on halves at a time), and was designed to have characteristics of: Resistance against all known attacks, Speed and code compactness on a wide range of platforms, & Design simplicity.

Rijndael is a substitution-permutation network (SPN) cipher, which means that it combines a series of substitutions and permutations to transform the plaintext into ciphertext. The substitution step substitutes each byte of the plaintext with another byte based on a lookup table. The permutation step shuffles the order of the bytes in the block. This process is repeated for multiple rounds, with the number of rounds depending on the block size and key length. Rijndael has a high security margin and has been extensively studied and analyzed by the cryptographic community. It has been widely adopted and is used in many applications that require strong encryption, such as secure communications, digital rights management, and financial transactions.

3.3 BASICS OF AES

The input to the AES encryption and decryption algorithms is a single 128-bit block, depicted in FIPS PUB 197, as a square matrix of bytes. This block is copied into the State array, which is modified at each stage of encryption or decryption. After the final stage, State is copied to an output. The key is expanded into 44/52/60 lots of 32-bit words (see later), with 4 used in each round.

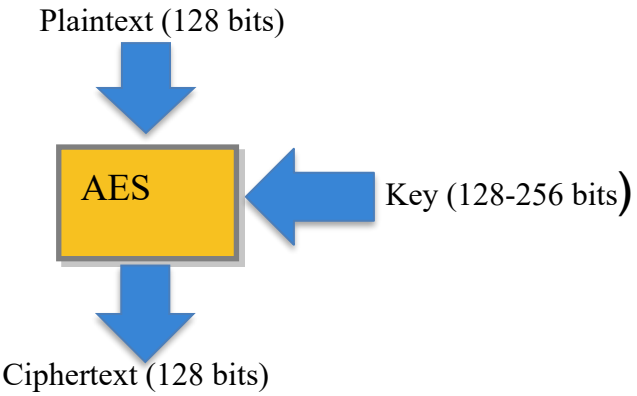
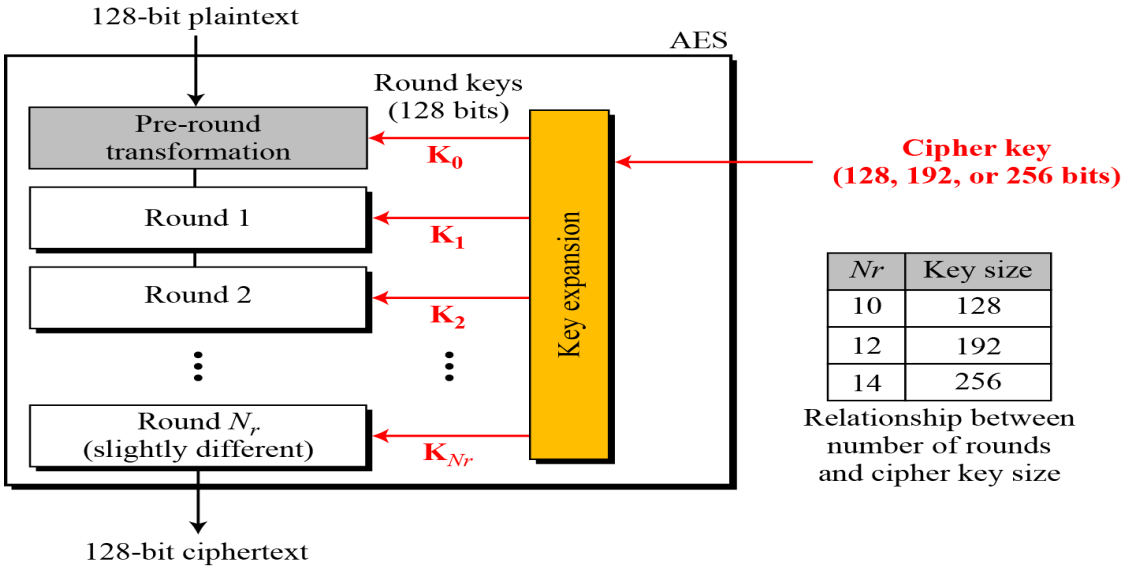


Figure 3.2: Basic view of AES



Nr	Key size
10	128
12	192
14	256

Relationship between number of rounds and cipher key size

Figure 3.3: Multiple rounds in AES

The data computation then consists of an "add round key" step, then 9/11/13 rounds with all 4 steps, and a final $10^{th}/12^{th}/14^{th}$ step of byte subs + mix cols + add round key.

This can be viewed as alternating XOR key & scramble data bytes operations. All of the steps are easily reversed, and can be efficiently implemented using XOR's & table lookups.

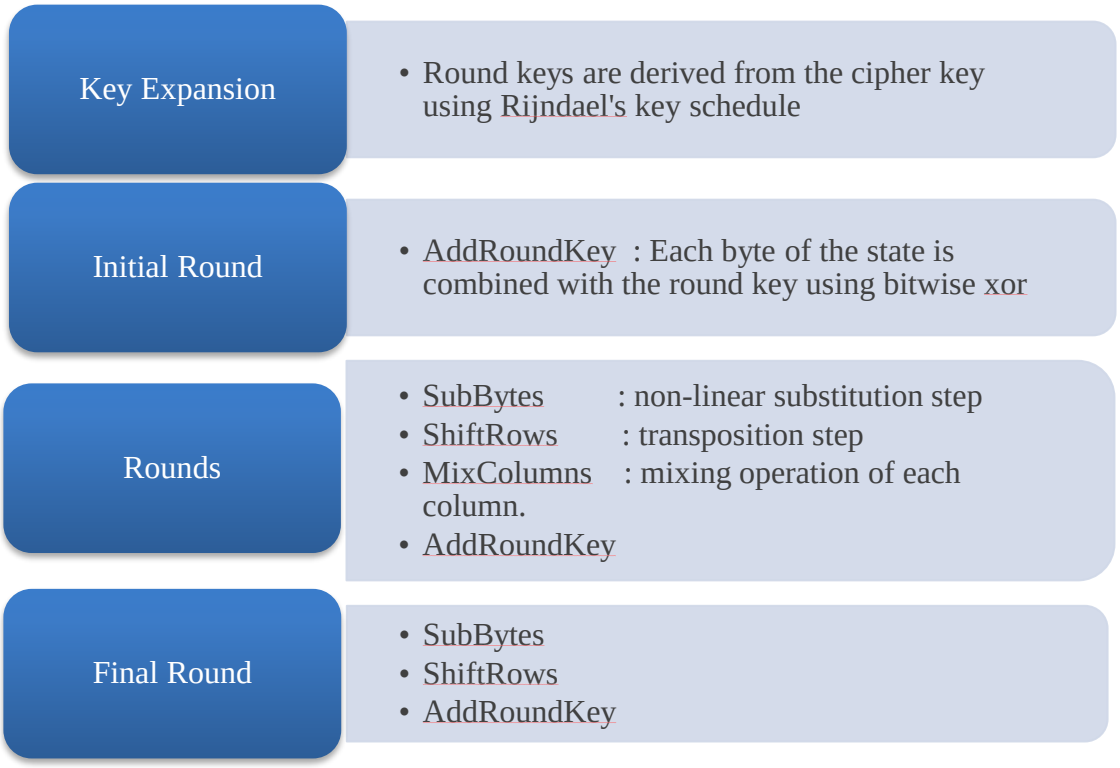


Figure 3.4: Description of AES

AES divides data into blocks of fixed size, where the block size is determined by the key size used for encryption. For AES-128, the most common key size, the block size is 128 bits or 16 bytes. For AES-192, the block size is 192 bits or 24 bytes, and for AES-256, the block size is 256 bits or 32 bytes. The use of a fixed block size is important for the algorithm's security, as it helps prevent attacks that exploit patterns in the data. Matrix Structure: Each block of data is divided into a matrix of 4 rows and 4 columns, where each element of the matrix represents one byte of data. This matrix is known as the State matrix and is used in the encryption process. During encryption, the State matrix undergoes a series of transformations, such as Sub Bytes, ShiftRows, MixColumns, and AddRoundKey, which scramble the data to make it difficult to decipher without the correct key.

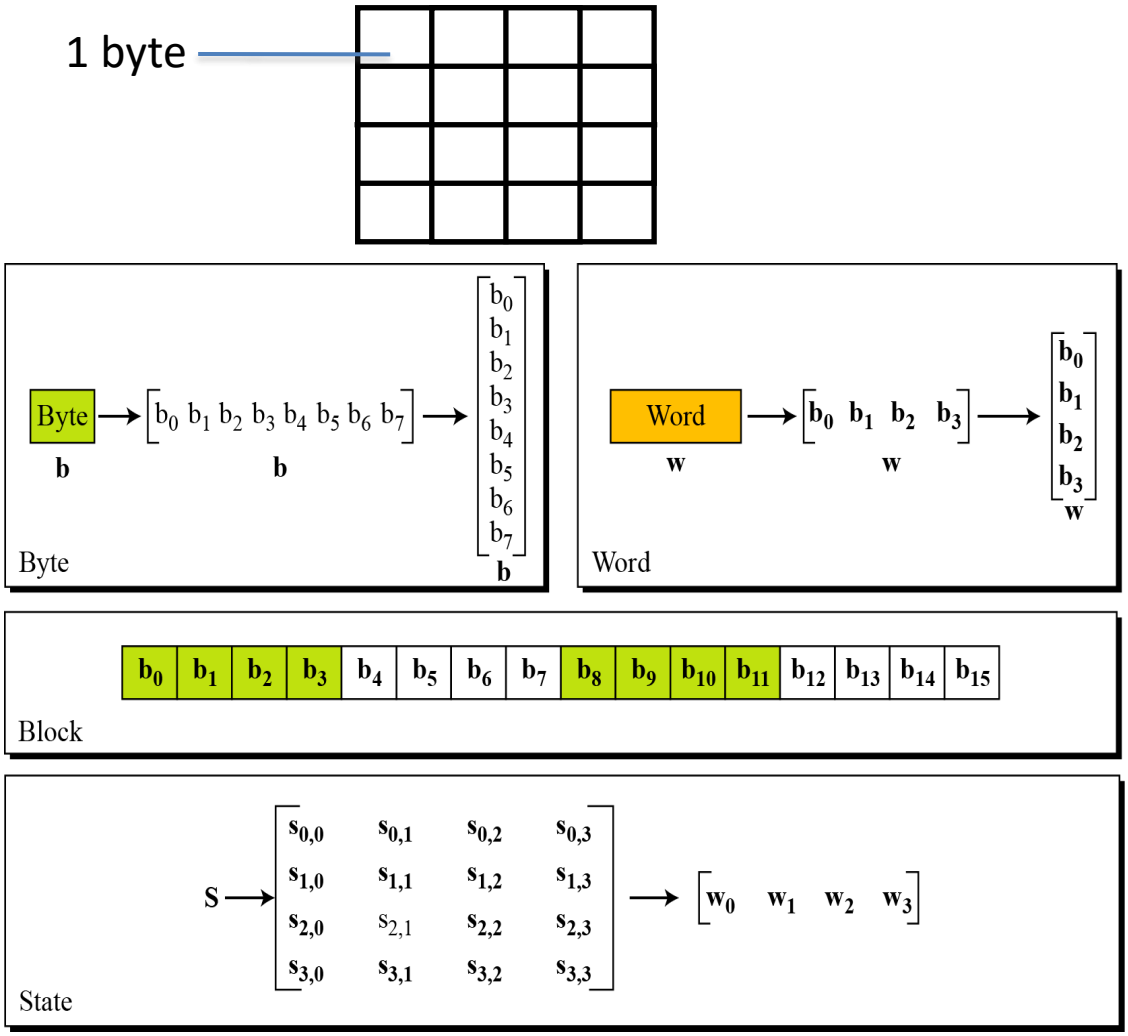


Figure 3.5: Data Unit of AES

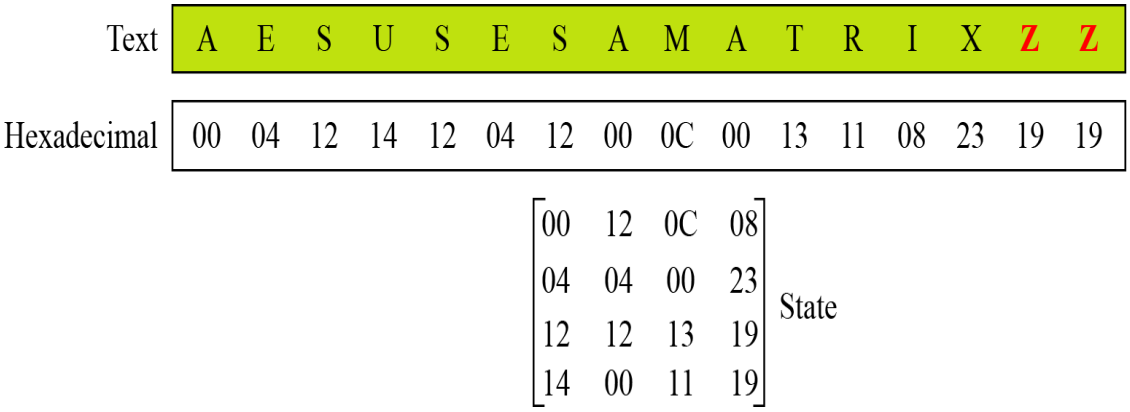


Figure 3.6: Changing plaintext to state

3.4 AES ARCHITECTURE

The architecture of the AES algorithm plays a crucial role in providing a high level of security for sensitive data. The AES algorithm uses a fixed block size, a key schedule and a round-based structure to encrypt and decrypt data efficiently and securely. The AES algorithm is widely used in various applications, such as online banking, e-commerce, and secure communication, where the protection of sensitive information is critical.

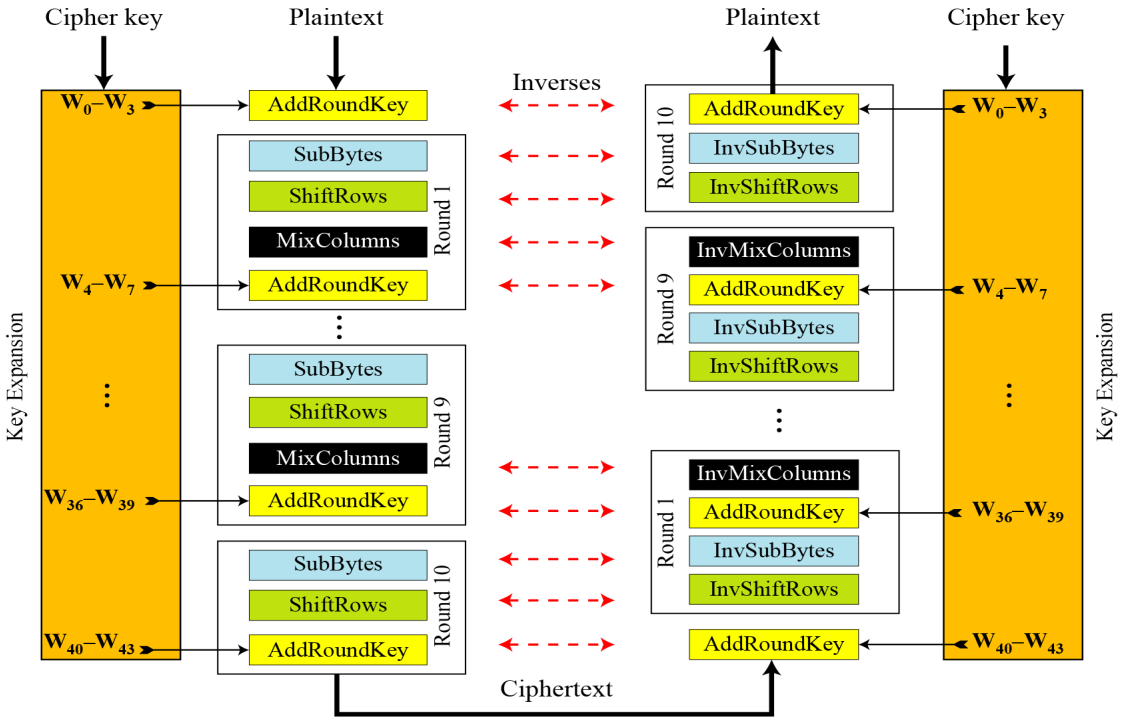


Figure 3.7: Architecture of AES

3.5 AES Key Scheduling

Takes 128-bits (16-bytes) key and expands into array of 44 32-bit words. The AES key expansion algorithm takes as input a 4-word (16-byte) key and produces a linear array of words, providing a 4-word round key for the initial AddRoundKey stage and each of the 10/12/14 rounds of the cipher. It involves copying the key into the first group of 4 words, and then constructing subsequent groups of 4 based on the values of the previous & 4th back words. The first word in each group of 4 gets “special treatment” with rotate + S-box + XOR constant on the previous word before XOR’ing

the one from 4 back. In the 256-bit key/14 round version, there's also an extra step on the middle word.

3.5.1 Key Expansion

Key expansion is a crucial step in the Advanced Encryption Standard (AES) algorithm, which is used to expand the original key into a larger set of round keys for use in the encryption and decryption. process.

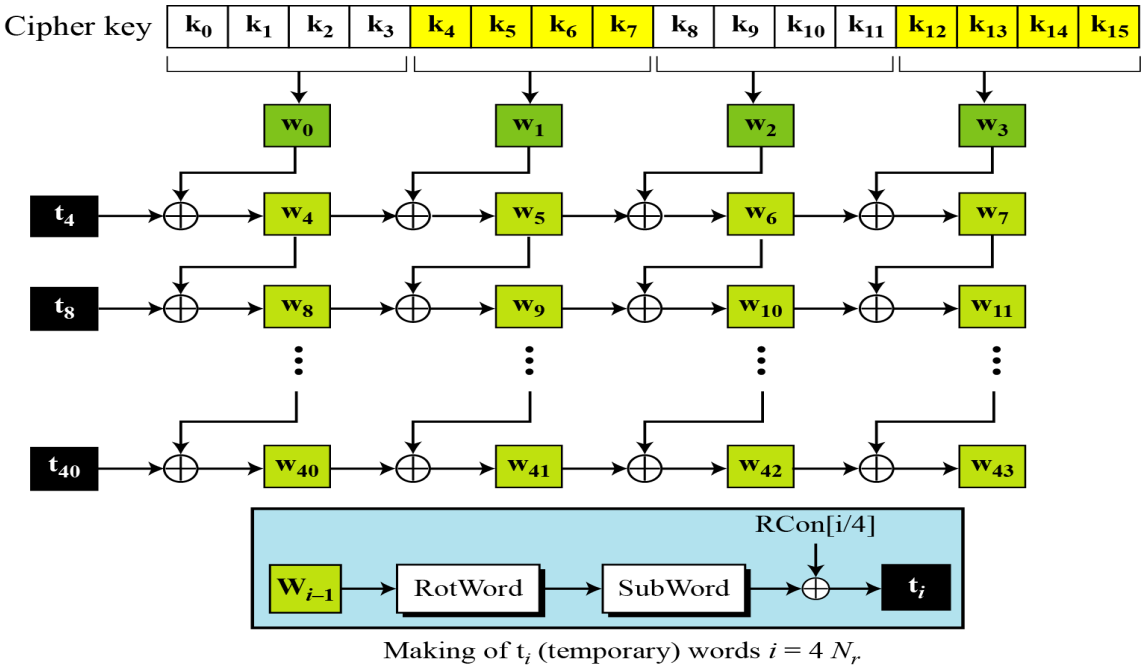


Figure 3.8: key Expansion scheme

The key expansion process in AES involves several steps, as follows:

Key Size Determination: The AES algorithm supports three different key sizes, namely 128 bits, 192 bits, and 256 bits. The key size determines the number of rounds used in the encryption and decryption process.

Key Padding: The original key is padded with zeroes to ensure that it matches the required key size. For example, if the original key is 128 bits and the algorithm requires a 256-bit key, then the key is padded with 128 bits of zeroes.

Round Constant Generation: A set of round constants are generated based on the key size. The round constants are used in the key expansion process to add additional complexity to the encryption process.

Key Expansion: The key expansion process involves applying a series of transformations to the original key to generate a larger set of round keys. The number of round keys generated is equal to the number of rounds required by the key size. For example, if the key size is 128 bits, then 11 round keys are generated for use in the 11 rounds of encryption and decryption. The key expansion process in AES is designed to create a set of round keys that are independent and unique for each round, which makes it difficult for attackers to recover the original key or decrypt the encrypted data. The key expansion process also adds additional complexity to the encryption and decryption process, which makes it more difficult to break the encryption.

3.5.2 Key Expansion submodule

RotWord performs a one-byte circular left shift on a word for example:

RotWord[b0,b1,b2,b3] = [b1,b2,b3,b0]

SubWord performs a byte substitution on each byte of input word using the S-box

SubWord(RotWord(temp)) is XORed with RCon[j] – the round constant

3.5.3 Round Constant (RCon)

RCON is a word in which the three rightmost bytes are zero It is different for each round and defined as: RCon[j] = (RCon[j],0,0,0) , where RCon[1] =1 , RCon[j] = 2 * RCon[j-1], Multiplication is defined over GF(2^8) but can be implement in Table

Round	*Constant (RCon)*	*Round*	*Constant (RCon)*
1	$(\underline{\mathbf{01}}\ 00\ 00\ 00)_{16}$	6	$(\underline{\mathbf{20}}\ 00\ 00\ 00)_{16}$
2	$(\underline{\mathbf{02}}\ 00\ 00\ 00)_{16}$	7	$(\underline{\mathbf{40}}\ 00\ 00\ 00)_{16}$
3	$(\underline{\mathbf{04}}\ 00\ 00\ 00)_{16}$	8	$(\underline{\mathbf{80}}\ 00\ 00\ 00)_{16}$
4	$(\underline{\mathbf{08}}\ 00\ 00\ 00)_{16}$	9	$(\underline{\mathbf{1B}}\ 00\ 00\ 00)_{16}$
5	$(\underline{\mathbf{10}}\ 00\ 00\ 00)_{16}$	10	$(\underline{\mathbf{36}}\ 00\ 00\ 00)_{16}$

Figure 3.9: Round constant values

Example of expansion of a 128-bit cipher key

Cipher key = 2b7e151628aed2a6abf7158809cf4f3c

w0=2b7e1516 w1=28aed2a6 w2=abf71588 w3=09cf4f3c

I	w_{i-1}	RotWord	SubWord	Rcon[i/4]	t_i	w[i-4]	w_i
4	09cf4f3c	cf4f3c09	8a84eb01	01000000	8b84eb01	2b7e1516	a0fafe17
5	a0fafe17	-	-	-	-	28aed2a6	88542cb1
6	88542cb1	-	-	-	-	Abf71588	23a33939
7	23a33939	-	-	-	-	09cf4f3c	2a6c7605

Table 3.1: Example of key generation

3.6 AES Round Operations

AES rounds are based on key size as 10 rounds for 128bit key ,12 rounds for 192 bit key and 14 rounds for 256 bit key and for round 1 to Nr-1 there will be 4 operations namely SubBytes , ShiftRows, MixedColumns and AddRoundkey , in the final round N Mixcolumn operation is excluded.

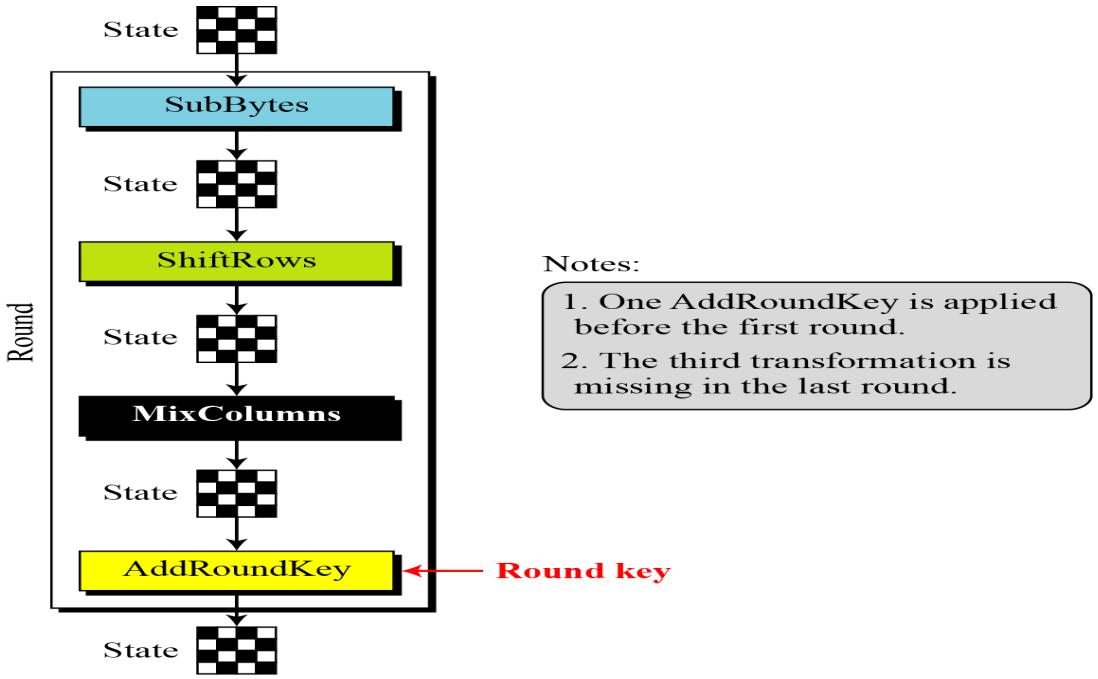


Figure 3.10: AES Round Operations

3.6.1 SubBytes: Byte Substitution

A simple substitution of each byte which provide a confusion andUses one S-box of 16x16 bytes containing a permutation of all 256 8-bit values,Each byte of state is replaced by byte indexed by row (left 4-bits) & column (right 4-bits).

eg. byte {95} is replaced by byte in row 9 column 5 which has value {2A}

S-box constructed using defined transformation of values in Galois Field- GF(2^8), SubBytes and inverse SubBytes are two fundamental operations used in the Advanced Encryption Standard (AES) cipher. They are part of the substitution layer in the AES cipher. The SubBytes operation is a nonlinear byte substitution that operates on each byte of the input data block. It substitutes each byte of the input block with another byte from a fixed 256-byte lookup table called the S-box. The S-box is created using a combination of mathematical operations such as modular arithmetic and inversion in the Galois field. The substitution process helps to provide diffusion and confusion in the cipher.

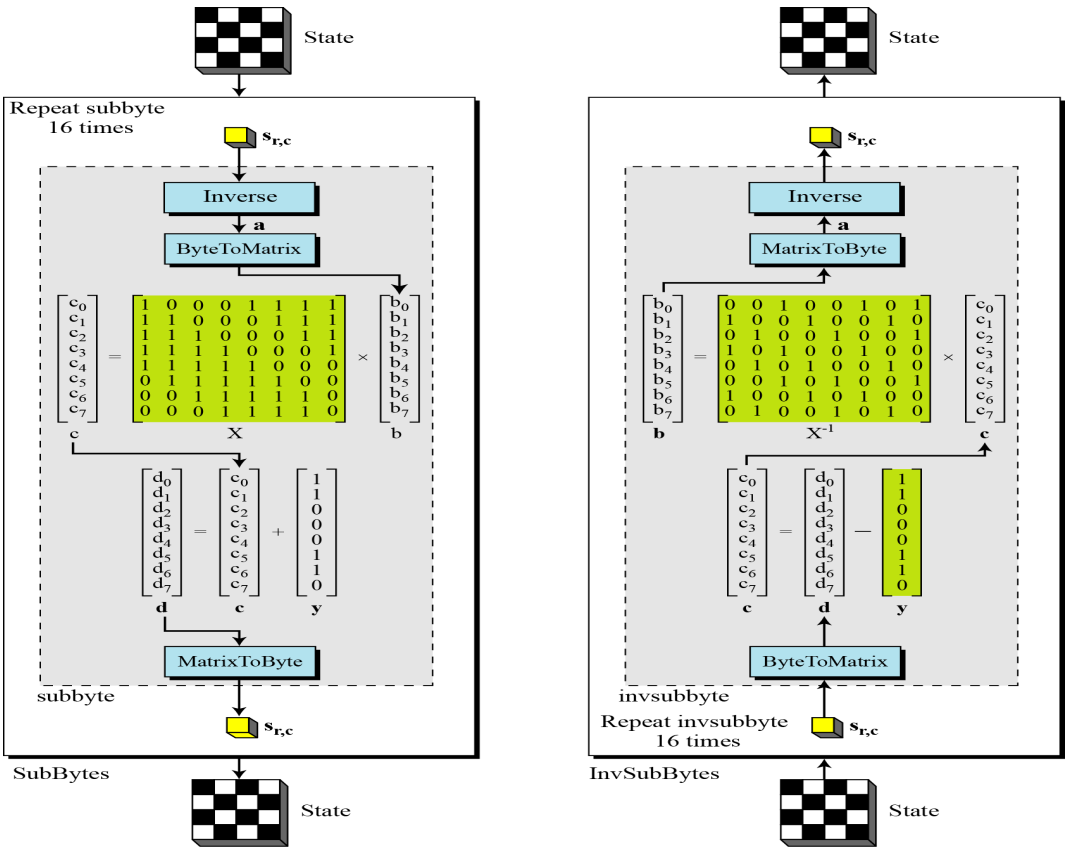


Figure 3.11: SubBytes and Inverse SubBytes

Both SubBytes and inverse SubBytes operations are important for the security of the AES cipher as they add an additional layer of complexity and nonlinearity to the cipher. The SubBytes operation involves 16 independent byte-to-byte transformations.

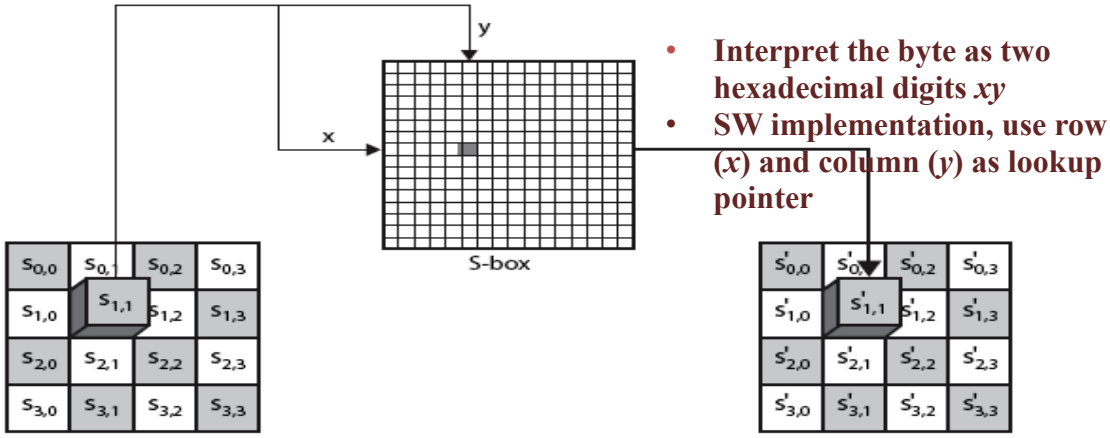


Figure 3.12:SubByte operation

		y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
	1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
	2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
	3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
	4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
	5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
	6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
	7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
	8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
	9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
	A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
	B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
	C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
	D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
	E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
	F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

		y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
	1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
	2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
	3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
	4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
	5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
	6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
	7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
	8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
	9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
	A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
	B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
	C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
	D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
	E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
	F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

Figure 3.13:SubByte and InvSubBytes Lookup Tables

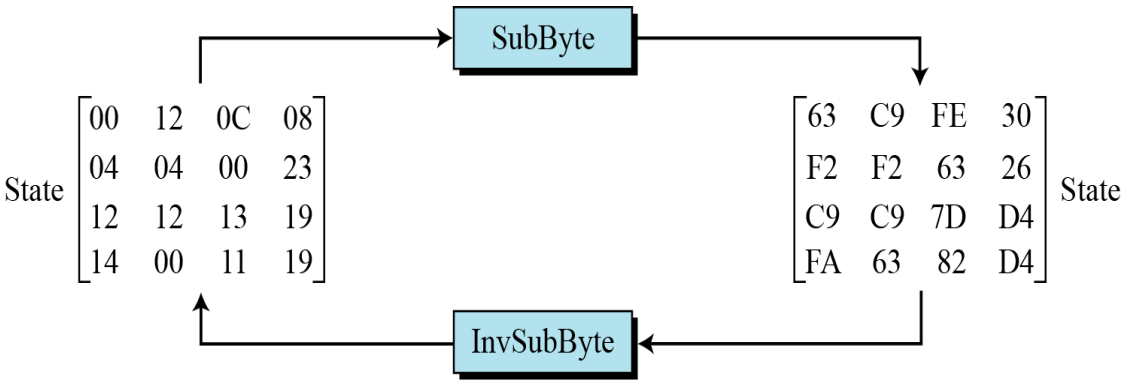


Figure 3.14: Sample Subbyte Transformation

3.6.2 ShiftRows

Shifting, which permutes the bytes. A circular byte shift in each each 1st row is unchanged,2nd row does 1 byte circular shift to left,3rd row does 2 byte circular shift to left,4th row does 3 byte circular shift to left. In the encryption, the transformation is called ShiftRows. In the decryption, the transformation is called InvShiftRows and the shifting is to the right.

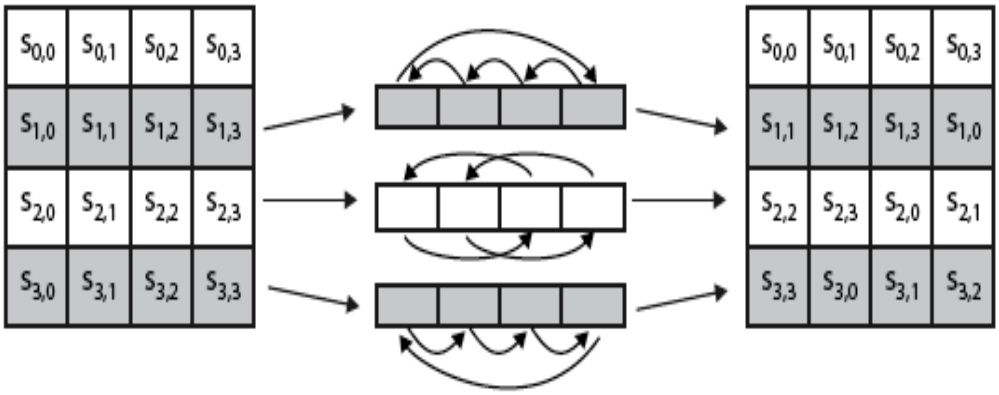


Figure 3.15: Shiftrows Scheme

The ShiftRows stage provides a simple “permutation” of the data, whereas the other steps involve substitutions. Further, since the state is treated as a block of columns, it is this step which provides for diffusion of values between columns. It performs a circular rotate on each row of 0, 1, 2 & 3 places for respective rows. When decrypting it performs the circular shifts in the opposite direction for each row. This row shift moves an individual byte from one column to another, which is a linear distance of a

multiple of 4 bytes, and ensures that the 4 bytes of one column are spread out to four different columns.

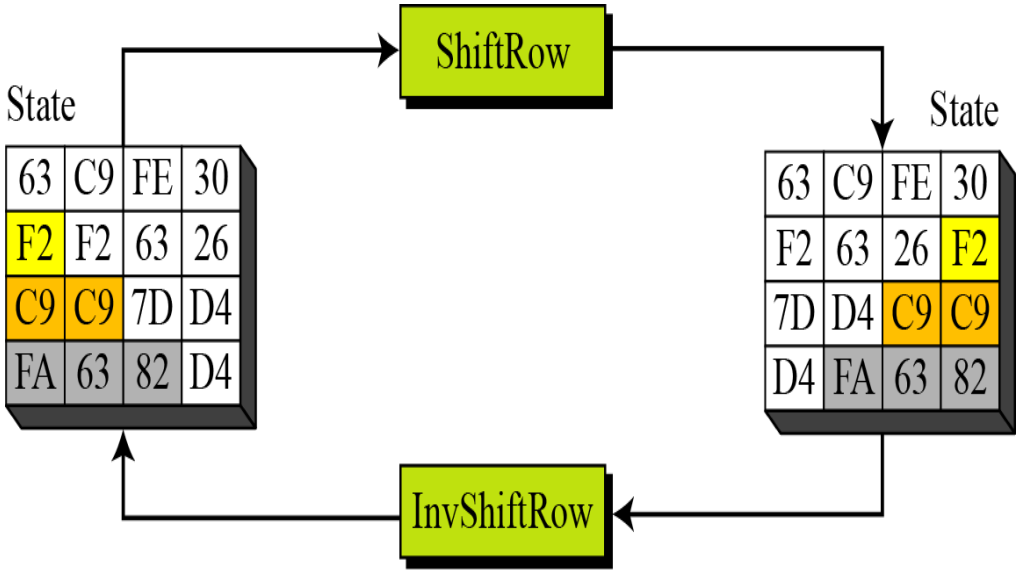


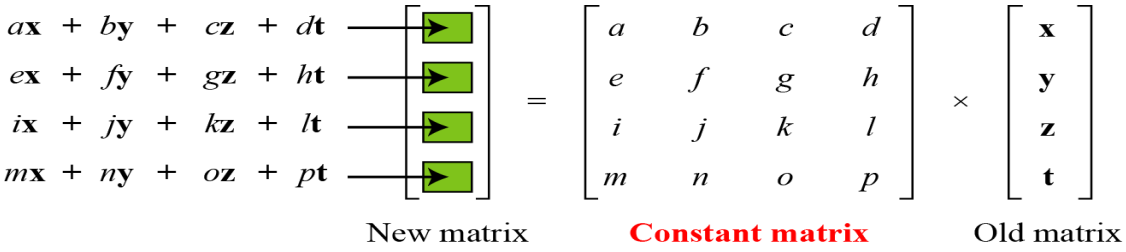
Figure 3.16: ShiftRows and InverseShiftRows

3.6.3 MixColumns

ShiftRows and MixColumns provide diffusion to the cipher. Each column is processed separately. Each byte is replaced by a value dependent on all 4 bytes in the column.Effectively a matrix multiplication in GF(2^8) using prime poly m(x) =x^8+x^4+x^3+x+1. The MixColumns stage is a substitution that makes use of arithmetic over GF(2^8). Each byte of a column is mapped into a new value that is a function of all four bytes in that column. It is designed as a matrix multiplication where each byte is treated as a polynomial in GF(2^8). The inverse used for decryption involves a different set of constants. The constants used are based on a linear code with maximal distance between code words – this gives good mixing of the bytes within each column. Combined with the "shift rows" step provides good avalanche, so that within a few rounds, all output bits depend on all input bits.In practise, Mix Columns is implemented by expressing the transformation on each column as 4 equations to compute the new bytes for that column. This computation only involves shifts, XORs & conditional XORs (for the modulo reduction).

The decryption computation requires the use of the inverse of the matrix, which has larger coefficients, and is thus potentially a little harder & slower to implement. The

designers & the AES standard provide an alternate characterisation of Mix Columns, which treats each column of State to be a four-term polynomial with coefficients in GF(2^8). Each column is multiplied by a fixed polynomial a(x) ..



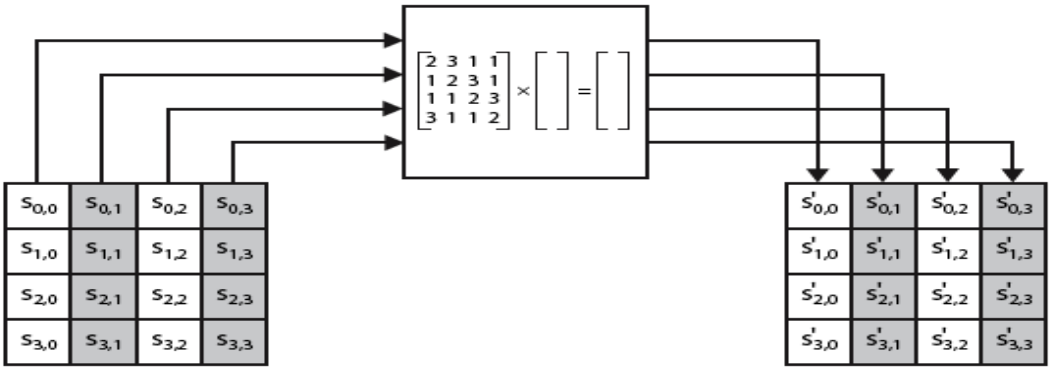


Figure 3.17: MixColumns scheme

3.6.4 AddRoundKey

The AddRoundKey operation is a crucial part of the AES (Advanced Encryption Standard) algorithm, to provide a high level of security for protecting sensitive data. It is one of the four operations that are performed in each round of encryption or decryption in AES, along with SubBytes, ShiftRows, and MixColumns. The AddRoundKey operation XORs each byte of data in the State matrix with a corresponding byte from the round key generated by the key schedule. The round key is derived from the original encryption key using the key schedule, which generates a series of round keys based on the original encryption key. The AddRoundKey operation helps to ensure that each round of encryption or decryption uses a unique key, which is essential for providing a high level of security. Without the AddRoundKey operation, the encryption or decryption process would be vulnerable to attacks that exploit patterns in the data or the key. The AddRoundKey operation is performed on a 128-bit State matrix that is divided into a 4x4 matrix of bytes. Each byte of the State matrix is XORed

with a corresponding byte from the round key. The round key is also divided into a 4x4 matrix of bytes, which corresponds to the 4x4 matrix of bytes in the State matrix.

The AddRoundKey operation is reversible, which means that it can be used for both encryption and decryption. During encryption, the round key is added to the State matrix, while during decryption, the round key is subtracted from the State matrix. This is possible because XOR is a bitwise operation that is its own inverse.

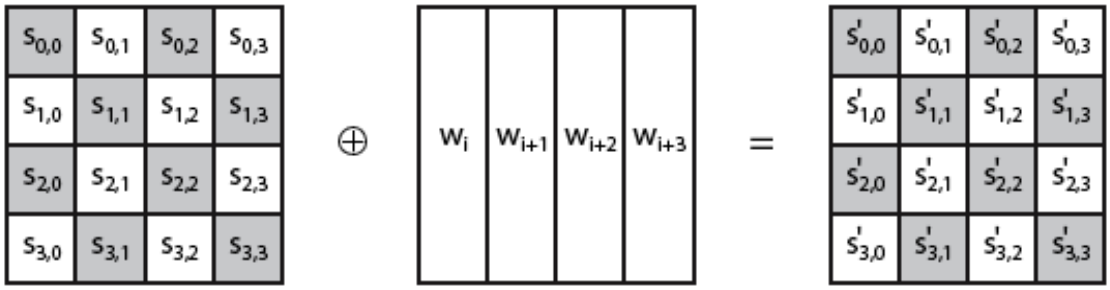


Figure 3.18: AddRoundKey Scheme

CHAPTER 4

PROPOSED METHOD

CHAPTER-4

PROPOSED METHOD

4.1 Introduction

The Advanced Encryption Standard (AES) is a widely used symmetric encryption algorithm that provides strong security and high performance. It has been adopted as a standard by many organizations and is used to secure sensitive data such as financial transactions, healthcare records, and government communications. While AES is highly secure, it is not immune to attacks, and researchers are constantly looking for ways to enhance its security.

In this project, we propose a modification to the AES algorithm that involves adding an extra AddRoundKey transformation between the ShiftRows and MixColumns transformations for rounds 1 to 9 only. This modification is intended to improve the avalanche effect of the AES algorithm, which is a measure of how much a small change in the plaintext input affects the ciphertext output.

Overall, we believe that our proposed modification to the AES algorithm has the potential to enhance the security of AES and make it even more resistant to attacks. By adding an extra AddRoundKey transformation, we aim to increase the diffusion of the input plaintext throughout the algorithm, resulting in a more robust and secure encryption process. We look forward to presenting our findings and contributing to the ongoing research on AES and its variants.

4.2 Modified AES

The proposed method of AES has similar structure as standard AES but with an Extra AddRoundKey between ShiftRows and MixColumns for rounds 1 to 9. Modified AES has the following steps:

1.Key Expansion: The step generates a set of round keys from the secret key input using a key expansion algorithm. Key generation involves 4 methods that will produce unique round keys, number of keys to be generated is based on number of rounds.

2.Initial Round: The input plaintext is first XORed with the first-round key. This can be implemented using the bitwise XOR operator (^).

3.Round Transformation: The modified AES algorithm consists of 9 rounds similar to standard AES but with an extra AddRoundKey. Each round consists of the following steps:

A. SubBytes: In this step, each byte of the input is substituted with a corresponding byte from a pre-defined S-box. This can be implemented using a lookup table or a substitution function.
B. ShiftRows: In this step, the bytes in each row of the input are cyclically shifted by a certain number of bytes.
C. AddRoundKey: This is an extra AddRoundKey transformation, in this step, the input is XORed with the round key corresponding to the current round.
D. MixColumns: In this step, each column of the input is transformed using a pre-defined matrix operation. This can be implemented using matrix multiplication or bitwise operations.
E. AddRoundKey: It is a default AddRoundKey operation as in standard AES. In this step, the input is XORed with the round key corresponding to the current round.

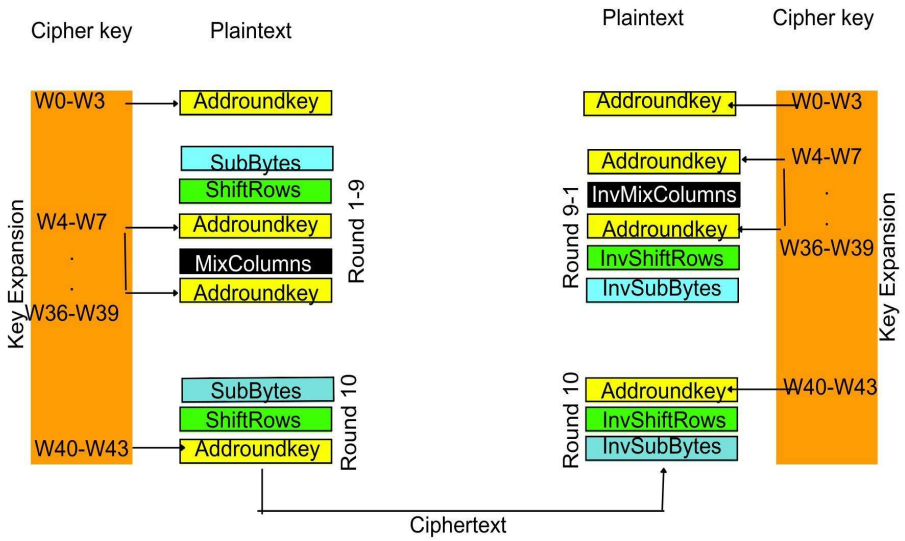


Figure 4.1 : Architecture of Modified AES

4. Final Round: The final round of the modified AES algorithm is the same as in the standard AES algorithm, except that there is no MixColumns transformation. The final round consists of the following steps:

A. SubBytes: In this step, each byte of the input is substituted with a corresponding byte from a pre-defined S-box. This can be implemented using a lookup table or a substitution function.

B. ShiftRows: In this step, the bytes in each row of the input are cyclically shifted by a certain number of bytes.

C. AddRoundKey: In this step, the input is XORed with the final round key. This can be implemented using the bitwise XOR operator (^).

Input and Output Formats: The input to the modified AES algorithm is a block of plaintext with a length of 128 bits (16 bytes), while the output is a block of ciphertext with the same length.

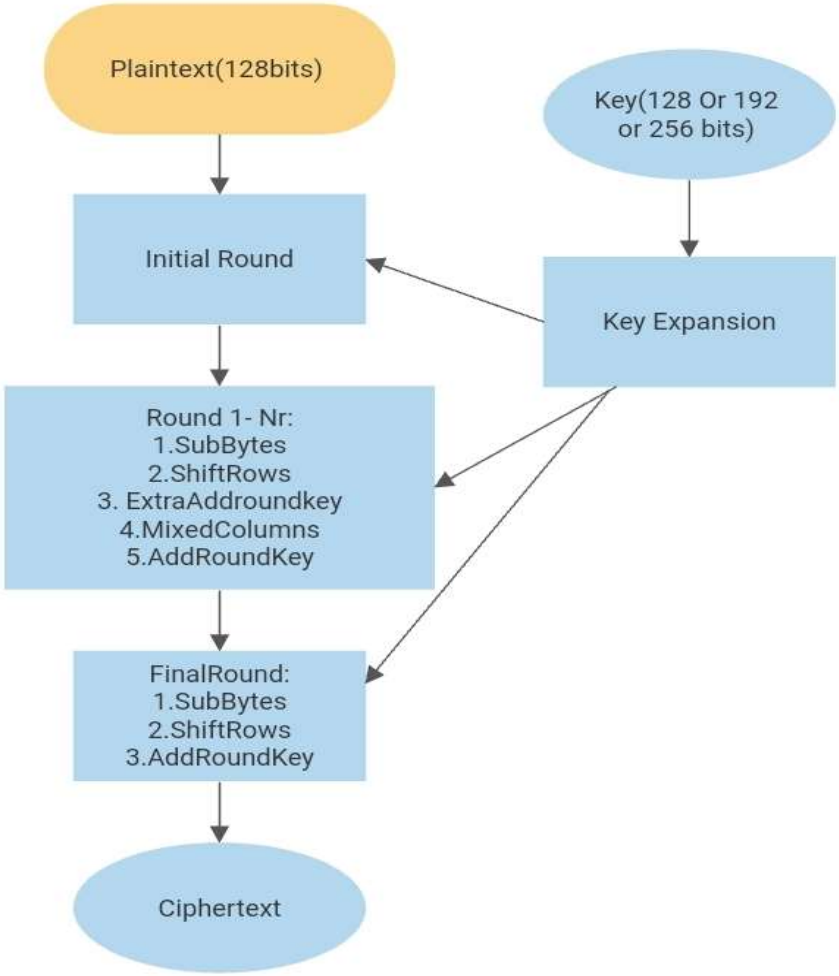


Figure 4.2: Flowchart of Modified AES Encryption

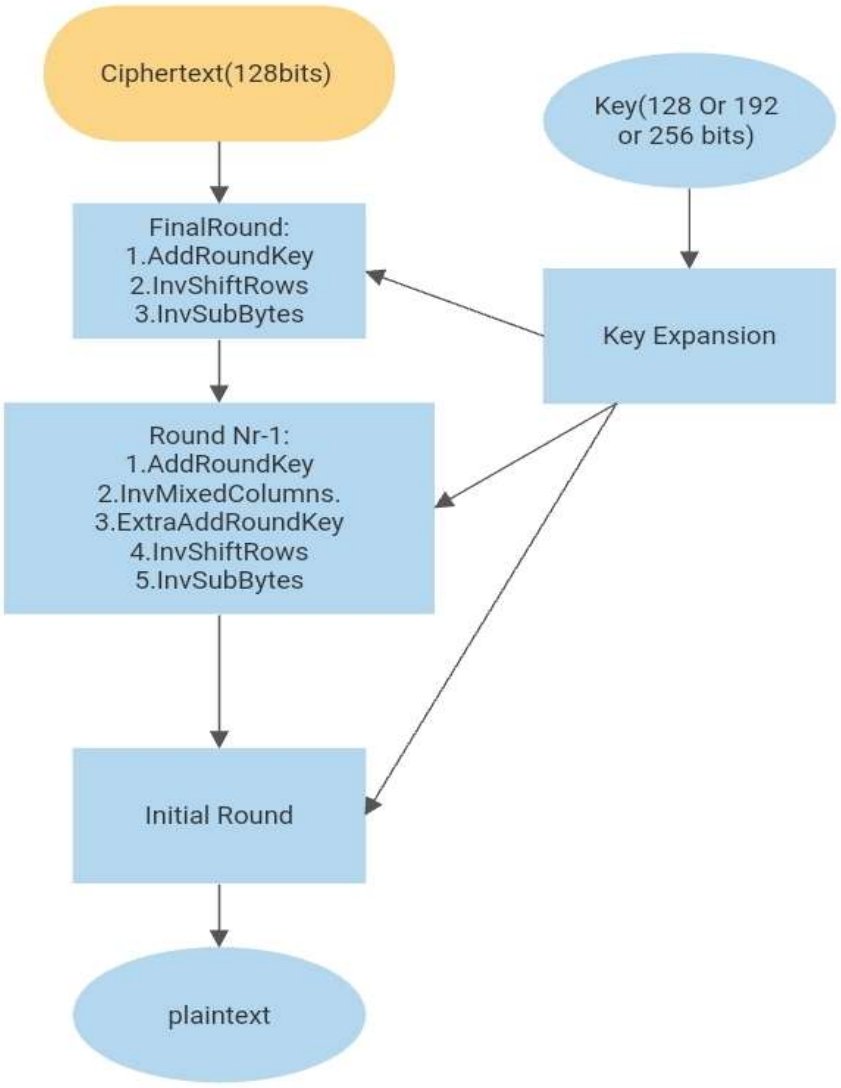


Figure 4.3: Flowchart of Modified AES Decryption

4.3 Implementation Details

For this project AES has been implemented in pure Python without any dependencies of using any External packages and proposed method is implemented by modifying the AES code with proposed methods and tested against some test cases.

OS: Windows 11

Processor: 12th Gen Intel(R) Core (TM) i7-1260P 2.10 GHz

RAM: 32.0 GB (31.7 GB usable)

System type:64-bit operating system, x64-based processor

4.4 Effects of Adding Extra AddRoundKey Transformation

The addition of an extra AddRoundKey transformation between the ShiftRows and MixColumns transformations in proposed method can have several effects on the security and performance of the AES algorithm. Here are some of the key effects:

Improved Avalanche Effect: The avalanche effect is a measure of how much the output of a cryptographic algorithm changes when the input is changed slightly. A good cryptographic algorithm should exhibit a strong avalanche effect, meaning that even small changes in the input should result in significant changes in the output. The addition of an extra AddRoundKey transformation in proposed method can improve the avalanche effect of the algorithm, as it adds an extra layer of confusion to the input before the MixColumns transformation.

Increased Security: The main motivation for adding an extra AddRoundKey transformation in the proposed method is to increase the security of the AES algorithm. By adding an extra layer of confusion to the input, the modified AES algorithm is less susceptible to attacks that exploit statistical properties of the input or key.

In conclusion, the addition of an extra AddRoundKey transformation between the ShiftRows and MixColumns transformations in the proposed method can improve the avalanche effect, increase the security, thus making modified AES algorithm secure against known attacks and exploits.

CHAPTER - 5

RESULTS

CHAPTER-5

RESULTS

5.1 Avalanche effect:

The avalanche effect is a measure of how much a small change in the input of an encryption algorithm affects the output. Avalanche effect is equal to number of bits changed in ciphertext due a bit change in plaintext or key to total number of bits in the ciphertext. A good encryption algorithm should have a high avalanche effect, meaning that even a small change in the input will cause a significant change in the output. In this experiment, Standard AES and Modified AES are compared using Avalanche Effect for 1-bit change and n- random bit changed in the key and plaintext.

5.1.1 Avalanche Effect due to 1-bit change in plaintext

It is primary objective to apply avalanche effect on modified AES algorithm and standard AES algorithm due to one bit change in the plaintext. 20 testcases have been taken with different key and plaintext . For each test case plaintext is encrypted using key and the plaintext is changed by 1 bit ,original ciphertext and changed ciphertext are compared to find percentage of bit flipped. Test results are evaluated to compare avalanche effect due to 1 bit change in plaintext between standard AES and modified AES as seen in the following table 5.1.

Sl.no	Plaintext	Key	Std AES cipher due to 1bit change in plaintext	MAES cipher Due to 1bit Change in plaintext	Std AES avalanche %	MAES avalanche%
1	57CDDD3D3A8A6828 10CA0DA088C99956	99FAFBBE766CD2C6 2BD1886209305EDE	1574C663F1F3AA34 4DDC944EFC33F4C0	9B6F513453EC26FB 892A24CFB2A0B086	48.4	59.4
2	0462A3030F59435C E0BEC73259D7E79E	AC8D053896EA55DA 573EEF93A280DE4C	80177093DBB598A3 C9155314FA201E1C	4EAA6E1CECB4D76A 76E366517F3346BA	49.2	51.6
3	97B085DF0CB2F436 09DF3CE62938F47D	D678D14ABA848000 CF5782B1C6FCDD8D	EABE88388EC8793D 73E2B2C9240DD792	E16CFF0032ED8E3A 8083C8419C65D694	46.1	55.5
4	D6C55981CF8E8D5A 5E87C4AD96B6441D	9E39E526EA504B11 9A8A9C771A3F010F	B9DC8016B60351D3 17DF2928E5CFC137	23CA43AC32D660F2 7485E84878CBF7DE	46.1	47.7
5	8C707E118B719841 D4CFF00B154808A5	6265EA9EB75C229E 08F9DE2C59D40314	FE1EA81E7E48C537 7AB269D8DD82450F	7744FED91B859F39 1B4960C1812C4F57	43.8	45.3

6	AB76FBB78AA13BF BC7872AA741616D1	2B5EB0C5C5DCF75D 186FAF5347D0968D	B43B20053AAA4AA9 4E302A8387009E38	AF11C5ABA4EE85E8 50E1F80D7468B306	46.9	50.0
7	5051BC450D6961A1 AC2A621057C2CCF5	5CEAB897D44EB925 EE43215BF16AEBF1	28E7BF1F3895E5594 2990DAD874C0C78	56E681C299250168 051E1F7C8924F60F	47.7	54.7
8	5ABA31C838FF5117 278AC7C4DEA2C283	2F59832DB44DD720 68D0D84D7FF4E2D3	7A925D65349299F0 8DA49DE4E188B12E	00C3399FED71A24A 6EE2989C1C23C0A9	52.3	55.5
9	C53705C34F76FB9F8 B376EAADE174DAA	42BDF4E1B4BAB0A 2303B877DCD8958F	93408A0324D51565 073DCA06A008B771	585DF770EA162C30 BE4C2F119E3F5D07	42.2	46.9
10	630B84E5ACB37C36 A39F73FDD51C826D	9175D1243E2EA0AF 0CF7E675C0851877	31B9CB8FB591AB8C AD8CB2A111785610	C232F740BB419164 3AE2E848A8A63565	46.1	47.7
11	A3454EC2E7E6A19A E1F6B155BF73AFBE	32CC4C0D258520D4 25FBEFDD33527BC4	78846927C5D08636 D9B0847C39EAA8D3	009BF3C0F7D87CB9 FF2ADB953C264351	49.2	58.6
12	6871D09F052DB2CB EEEB55D08ED4FD1	F28B17FF2C1BD57E D5026DA69AC13C24	453369C89488356B DC962E4802EB54ED	069ADD35AEAD539 86C6E91F1D8C29D6	43.8	53.1
13	2837F725D1783221 F78145D68FCDF1FE	3EA4565DBC5DF9A4 862A1A0F504336E7	00CA3C6ACE1BCA9 5C42812C43F3F8DE	86FAE9F55574E2B8 08BDFD3CF39F0038	46.1	51.6
14	71E633D22D2677F1 463E619B3018C9CD	3C539652AF487472 2CC7C49B53B8AEF4	8FFF994423A0C4A54 7C60E82BDACEBB3	EF1A794598B08ADA 7D7B7A807D736B7D	45.3	46.1
15	E3D3CFEA4130C82C B0A2485F76920B8D	A649054E04B06223 3804AB6E472D329A	6DCC05D84021AFE0 6C07BF9EEB408058	D2432CE4FB07A418 9C7EF575DF2BB032	52.3	54.7
16	AF2B9C0EA8E0DEF4 1B24481ACFA08CBA	40AD91974428DCA4 F7C6CB099A048AEA	F64353508A940A85 BEE06A0EC984F176	F41803EF75982CEB B640A07A357C09B7	41.4	50.0
17	D711A421DC7B5A9 F42AABEA10158C8A	CB1A603B0B5A6975 68B5BD5F928BD845	C84AA65CF4C28603 6BD9C28B42F1EE05	DEACDAA2D1CFCB 9C6A3CC06C8D97F8	47.7	55.5
18	4F0E24E41DF459F7 F3EFC3FE87A692A7	C219DA2A16E494DD DAC4A8AC564F70A	0940989840540D37 515634BA1F5D654E	8D119D6490EF6D3D 32D8C6438BF29822	45.3	50.0
19	313FBBAE81384B19 C7FE4F888F99AE55	D3FD568D71A491F2 640652DE0C947334	63910B16D142B4B9 B36D5A5B6BAE0066	F7E180CE48F7DF91 5C8A2F0638698C6A	49.2	52.3
20	30AB028DA58124DE 29F45ACB93E8850A	8C3CE8BF06CB7CEE 7467B7F2A1EB8926	0AB33F3CB6C84BC8 0E9D91AC11C30452	E32840BDB33F6612 769C8F8947E7EACC	45.3	46.9

Table 5.1: Avalanche Effect of MAES-128 and Std AES-128 due to 1bit change in plaintext

The above table consists of plaintexts , keys ,ciphertext due to 1-bit change in plaintext by Std AES-128,ciphertext due to 1-bit change in plaintext by MAES-128 and their respective Avalanche effects . Graphs are drawn between avalanche effect of Std AES-128 and MAES-128.

Results show that Modified AES-128 has better avalanche effect which means that it is difficult to cryptanalysis to make prediction s about the input.

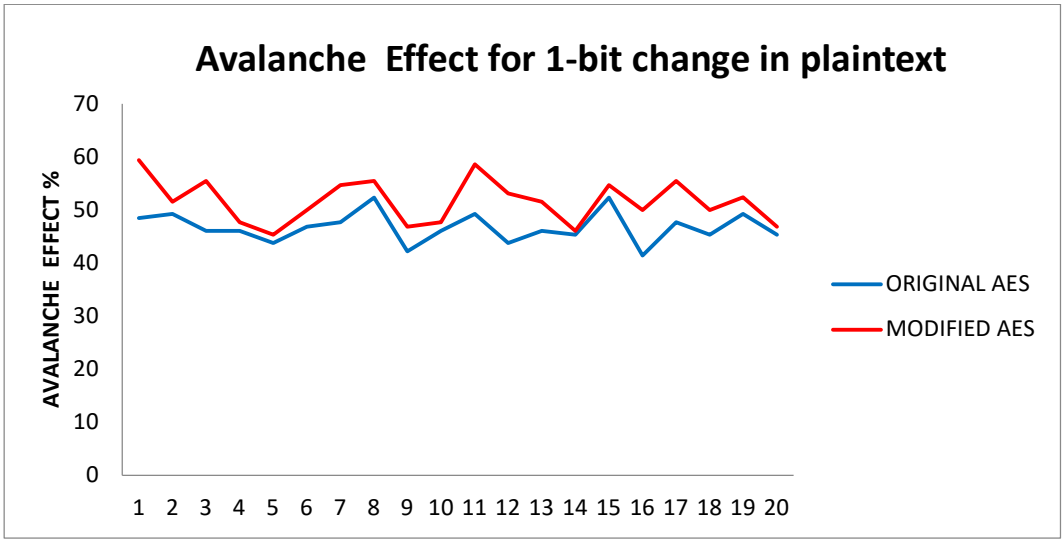


Figure 5.1: Avalanche Effect of MAES -128 and Std AES-128 due to 1-bit change in plaintext

5.1.2 Avalanche Effect due to 1- bit change in key

A comparision has to be made on modified AES algorithm and standard AES algorithm due to one bit change in the key. 20 testcases have been taken with different key and plaintext . For each test case plaintext is encrypted using key and the key is changed by 1-bit ,original ciphertext and changed ciphertexts are compared to find percentage of bits flipped. Test results are evaluated to compare avalanche effect due to 1-bit change in key between standard AES and modified AES as seen in the following table 5.2.

Sl. no	Plaintext	key	Std AES cipher due to 1bit change in key	MAES cipher due to 1bit change in key	Std AES avalanche %	MAES avalanche%
1	57CDDD3D3A8A6828 10CA0DA08899956	99FAFBBE766CD2C62 BD1886209305EDE	F26844233E5A48576D 2548EDAECF2CE7	226BB633E216310A32 413D8C1EAF74A5	44.5	53.9
2	0462A3030F59435CE0 BEC73259D7E79E	AC8D053896EA55DA 573EEF93A280DE4C	7D1166D18E9A981ED BF4058B93812521	A56D11FE1DE420B1C A1B5A31657E202	43.0	46.9
3	97B085DF0CB2F4360 9DF3CE62938F47D	D678D14ABA848000C F5782B1C6FCDD8D	3340D838FEA90B2F9 48D706307DE2D4F	C65752A3135030E0B B2F72AF83E755CE	46.9	53.1

4	D6C55981CF8E8DA5 E87C4AD96B6441D	9E39E526EA504B119 A8A9C771A3F010F	5129F2351855F588AB 31D4BC0C687C91	CE95C4615770F2AAB CAC415797C8E186	51.6	53.1
5	8C707E118B719841D4 CFF00B154808A5	6265EA9EB75C229E0 8F9DE2C59D40314	B5B4ADFEBA6B7A19 7BDD57A3B7A0984	A58872B92163990104 51387D77BF6726	50.0	51.6
6	AB76FBB78AA13BFB C7872AA741616D1	2B5EB0C5C5DCF75D 186FAF5347D0968D	87C53BBCC6EAD4A4 B1917992ECE0DA22	5348BC416BE4623BD A20CA4538E10135	46.1	51.6
7	5051BC450D6961A1A C2A621057C2CC5	5CEAB897D44EB925 EE43215BF16AEBF1	4BD23E6310840F0B27 61822B1BBDF59F	90EED358D3F29B5AF 1AAF290C1C50FD	45.3	50.0
8	5ABA31C838FF51172 78AC7C4DEA2C23	2F59832DB44DD7206 8D0D84D7FF4E2D3	A826A30EC3D4AD8C DD757B804101A781	89DCBFBA9DF6C8D E2AB4EE1494D758C0	43.0	50.0
9	C53705C34F76FB9F8 B376EAADE174DAA	42BDF4E1B4BAB06A 2303B877DCD8958F	BB7DA805A6E1B222 FB082E8687B1BFB4	F8A118D1FE94C3E01 98F81C5CAAF3FC2	43.8	46.9
10	630B84E5ACB37C36A 39F73FDD51C826D	9175D1243E2EA0AF0 CF7E675C0851877	0CBE7085C28A2B166 5D9AD979B188C23	B4949385C34589BC8F 386869BD632098	43.8	46.9
11	A3454EC2E7E6A19A E1F6B155BF73AFBE	32CC4C0D258520D42 5FBEFDD33527BC4	5532AE5BC55DA52E4 32B1AA0564D8272	D9DA62A8AC7BC9E CDAD23E6A708762F9	38.3	53.1
12	6871D09F052DB2C3B EEEB55D08ED4FD1	F28B17FF2C1BD57ED 5026DA69AC13C24	3BF0506FDA5B0E09F EF5936425834C73	A47367DE6285B6D66 1E8F2411FF6A066	46.1	51.6
13	2837F725D1783221F7 8145D68FCDF1FE	3EA4565DBC5DF9A4 862A1A0F504336E7	A5530DBCD3EEE479 50271C1AF0EC9EBC	40A90517ED451DCD7 64E7781B8BB46CA	47.7	48.4
14	71E633D22D2677F146 3E619B3018C9CD	3C539652AF4874722C C7C49B53B8AEF4	F2072AAA5BF7D16B 0E695C82C9A4AD95	F4C5DE735E5CD943F 8F5B4A421B20773	51.6	54.7
15	E3D3CFEA4130C82C B0A2485F76920B8D	A649054E04B0622338 04AB6E472D329A	4424B0B874A59DEC5 5A82C3518501552	C878604F740C2C02E2 548DEDB8A2DDCF	44.5	53.9
16	AF2B9C0EA8E0DEF4 1B24481ACFA08CBA	40AD91974428DCA4F 7C6CB099A048AEA	645C0583ABD59E692 6CBA08B47CECBCD	D55033EE1BAD748D E16B972B0116FADC	46.1	54.7
17	D711A421DC7B5AA9 F42AABEA10158C8A	CB1A603B0B5A69756 8B5BD5F928BD845	851200403930CACAD 8D25E429CEF6642	9EEA16D1378AC0703 60C88052ACCF707	46.9	47.7
18	4F0E24E41DF459F7F3 EFC3FE87A692A7	C219DA2A16E494DD DAC4A8AC564F70BA	C65C40D51F1560D6C A48300F7217AE65	0A316FC23449C32438 F01F070F0DDCA3	45.3	50.0
19	313FBBAE81384B19C 7FE4F888F99AE55	D3FD568D71A491F26 40652DE0C947334	8FF52A52AF036D4B0 FA59E16025F02DC	7799091D47986FBC4F A98677E2BEF349	49.2	51.6
20	30AB028DA58124DE2 9F45ACB93E8850A	8C3CE8BF06CB7CEE 7467B7F2A1EB8926	61BB6CB043652B386 10E3110B47E9465	D6D5A3440512118CE 0F882CAB1E6953A	43.8	46.1

Table 5.2: Avalanche Effect of MAES-128 and Std AES-128 due to 1bit change in key.

The above table5.2 consists of plaintexts , keys ,ciphertext due to 1-bit change in key by Std AES-128,ciphertext due to 1-bit change in key by MAES-128 and their respective Avalanche effects . Graphs are drawn between avalanche effect of Std AES-128 and MAES-128.Results show that Modified AES-128 has better avalanche effect which means that it is difficult to cryptanalysis to make predictions about the key , in which it is impossible to decrypt data without key.

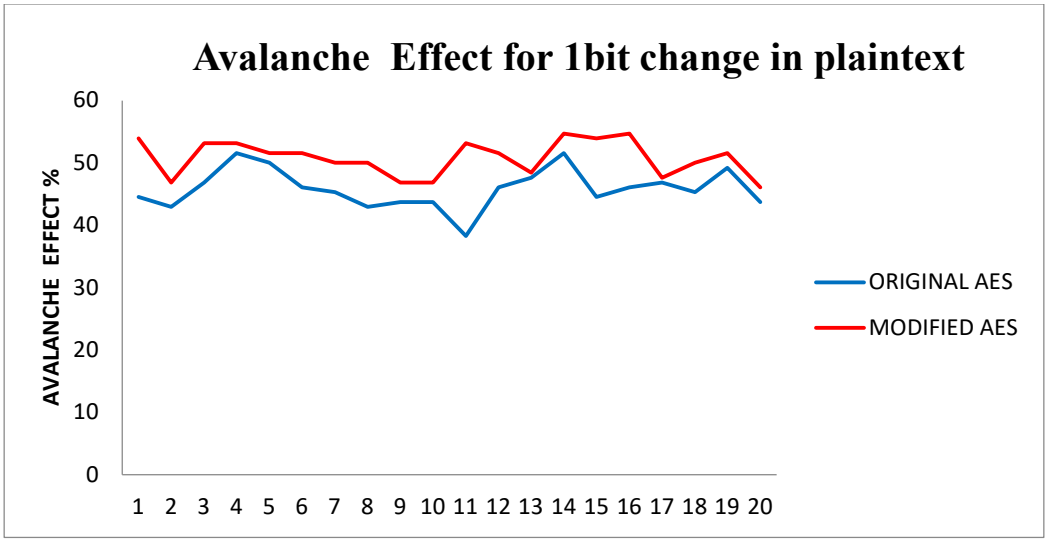


Figure 5.2: Avalanche Effect of Modified AES-128 VS Standard AES-128 due to 1-bit change in key

5.1.3 Avalanche Effect due to n- bits change in key and plaintext

A plaintext and key are selected and avalanche effect is found for standard AES and Modified AES by changing n- random bits in key and plaintext. Here n has taken values n=3,5,7,11,……35,37.

NUMBER OF BITS	MODIFIED KEY	MODIFIED PLAINTEXT	ORIGINAL KEY	ORIGINAL PLAINTEXT
3	54.69	56.25	50.78	53.13
5	48.44	57.81	42.19	48.44
7	42.97	51.56	45.31	49.22
9	57.03	42.97	45.31	43.75
11	43.75	48.44	50.00	58.59

13	51.56	46.09	47.66	50.78
15	53.13	54.69	54.69	54.69
17	52.34	47.66	50.78	53.91
19	47.66	53.91	57.03	50.00
21	42.97	44.53	43.75	39.84
23	47.66	53.13	52.34	48.44
25	56.25	47.66	51.56	51.56
27	55.47	52.34	50.78	50.00
29	45.31	46.88	52.34	52.34
31	43.75	50.00	43.75	53.91
33	50.00	50.00	48.44	49.22
35	50.00	50.00	46.88	49.22
37	58.59	42.97	46.88	52.34

Table 5.3 : Avalanche effect for n-random bits changed for Std AES-128 and Modified AES-128 for the key: D678D14ABA848000CF5782B1C6FCDD8D and plaintext: 97B085DF0CB2F43609DF3CE62938F47D

Above table 5.3 consists of Avalanche effects due n-bits changed in key and plaintext for Std AES-128 and Modified AES-128.N bits are changed randomly among 128 bits. Graphs are drawn for avalanche effect vs number of bits changed. Results show that Modified AES has better avalanche effect, which shows that Modified AES is more Secure than Std AES.

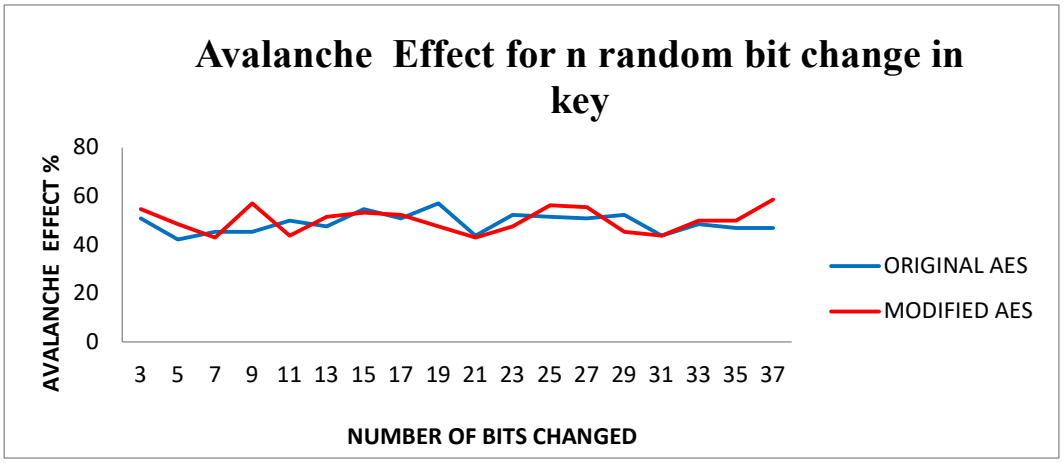


Figure 5.3: Avalanche Effect of MAES-128 VS Std AES-128 due to n random bits changed in key

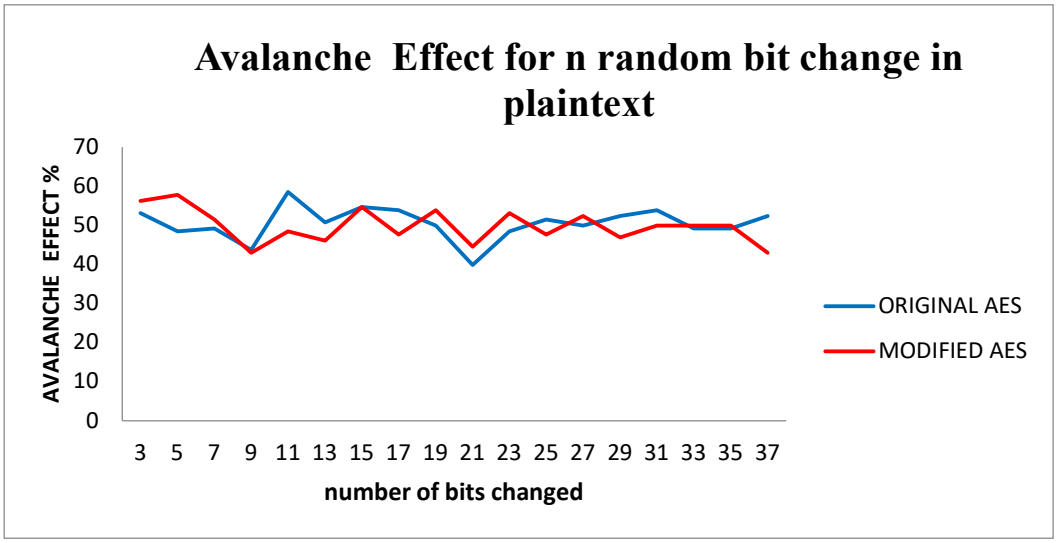


Figure 5.3: Avalanche Effect of Modified AES-128 VS Standard AES-128 due to n-bits change in plaintext.

5.2 Performance

System configurations: Processor :12th Gen Intel(R) Core(TM) i7-1260P 2.10 GHz Installed RAM:32.0 GB(31.7 GB usable) ,System type:64-bit operating system, x64-based processor.

It is necessary to calculate the simulation time of Modified AES and Std AES to evaluate the performance. Performance has been calculated by taking block size of 1 to 5000 for each block-16bytes size and simulated the Std AES and Modified AES in Python. Encryption time and Decryption time has been noted in the below table 5.4.

BLOCK SIZE	OAES ENCRYPTION TIME	OAES DECRYPTION TIME	MAES ENCRYPTION TIME	MAES DECRYPTION TIME
200	0.038	0.063	0.041	0.067
400	0.090	0.109	0.080	0.122
600	0.119	0.174	0.126	0.208
800	0.141	0.227	0.159	0.251
1000	0.183	0.297	0.186	0.309
1200	0.200	0.349	0.219	0.383
1400	0.246	0.415	0.282	0.426

1600	0.279	0.476	0.295	0.502
1800	0.321	0.534	0.342	0.564
2000	0.347	0.594	0.374	0.622
2200	0.389	0.661	0.407	0.692
2400	0.418	0.703	0.453	0.802
2600	0.456	0.781	0.519	0.830
2800	0.507	0.856	0.546	0.870
3000	0.540	0.915	0.587	0.965
3200	0.576	0.972	0.613	1.022
3400	0.596	1.046	0.660	1.074
3600	0.643	1.098	0.703	1.142
3800	0.679	1.153	0.725	1.231
4000	0.723	1.234	0.771	1.292
4200	0.777	1.304	0.833	1.353
4400	0.787	1.376	0.853	1.451
4600	0.846	1.418	0.894	1.505
4800	0.882	1.488	0.935	1.565
5000	0.913	1.554	0.988	1.628

Table 5.4: Represents Encryption and Decryption time of Standard AES-128 and Modified AES-128

Test has been implemented in python environment in which both Std AES and MAES are implemented in python and encryption, decryption times are noted in the above tanle 5.4.

Test results has been visualized through line graphs for Std AES-128 and Modified AES-128 and results show that Modified AES-128 has deviated from Std AES-128 for above 2000 blocks until both have same encryption and decryption time . As below figure 5.4 and 5.5 show that both std AES and Modified AES has same exponential time in which Modified AES got deviated by small variation as it provide extra security , so it can be considered to be a good performance .

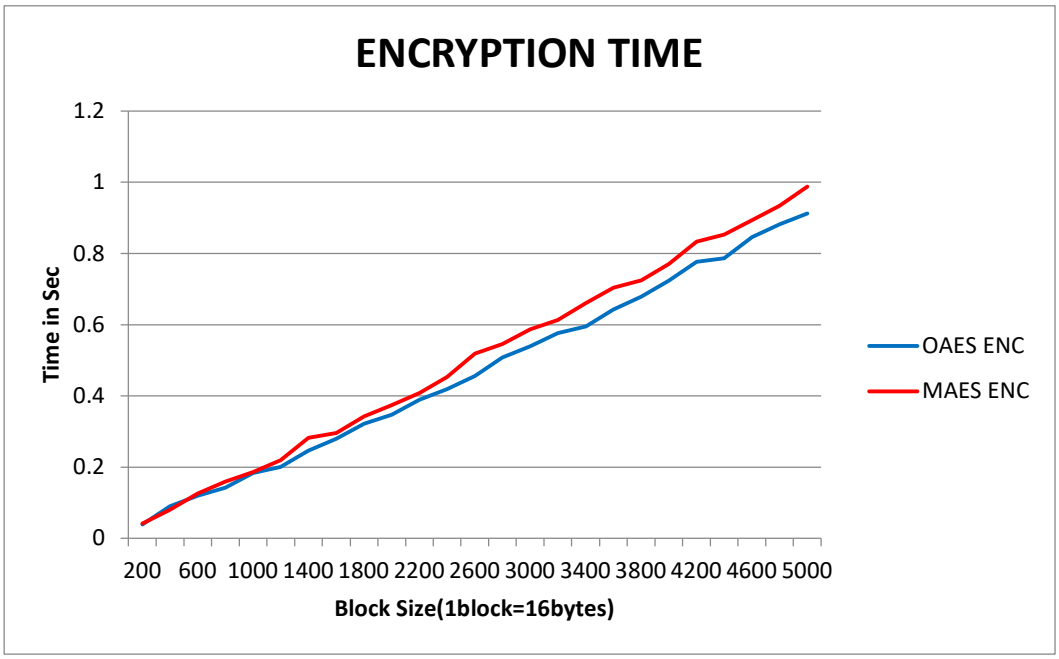


Figure 5.4:Encryption Time of Modified AES-128 and Standard AES-128

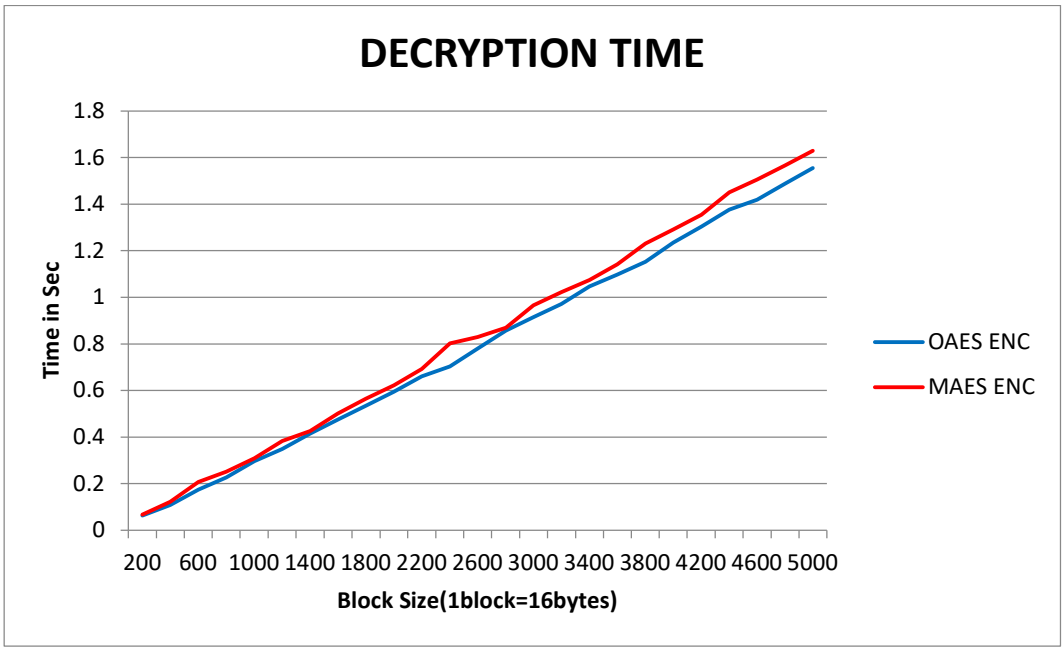


Figure 5.5: Decryption Time of Modified AES-128 and Standard AES-128

CHAPTER - 6

CONCLUSION & FUTURE SCOPE

CHAPTER-6

CONCLUSION AND FUTURE SCOPE

Conclusion:

We have analyzed the avalanche effect and performance of the Advanced Encryption Standard (AES) and Modified Advanced Encryption Standard (MAES) algorithms. The avalanche effect was analyzed for both key and plaintext with varying number of bit changes. Performance analysis was done by measuring the encryption and decryption time for both algorithms. From the analysis of the avalanche effect, it was observed that Modified AES has better avalanche effect than Standard AES which depicts that Modified AES is more secure than Std AES.

The performance analysis revealed that Modified AES has slightly higher in encryption and decryption time compared to Std AES as size of data increased exponentially, which is due to Added AddRoundKey in Modified AES.Overall, the results suggest that Modified AES has better security than Standard AES, which is depicted by higher avalanche effect.

Future Scope:

In the future, more comprehensive tests can be performed on the modified AES algorithm, such as fault attacks and side-channel attacks. Also, the algorithm can be further optimized to reduce the encryption/decryption time while maintaining a good avalanche effect.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Abdullah, A. M., & Aziz, R. H. H. (2016, June). New Approaches to Encrypt and Decrypt Data in Image using Cryptography and Steganography Algorithm., International Journal of Computer Applications, Vol. 143, No.4 (pp. 11-17).

[2] Singh, G. (2013). A study of encryption algorithms (RSA, DES, 3DES and AES) for information security. International Journal of Computer Applications, 67(19).

[3] Gaj, K., & Chodowiec, P. (2001, April). Fast implementation and fair comparison of the final candidates for Advanced Encryption Standard using Field Programmable Gate Arrays. In Cryptographers' Track at the RSA Conference (pp. 84-99). Springer Berlin Heidelberg.

[4] Stallings, W. (2006). Cryptography and network security: principles and practices. Pearson Education India.

[5] Yenuguvanilanka, J., & Elkeelany, O. (2008, April). Performance evaluation of hardware models of Advanced Encryption Standard (AES) algorithm. In Southeastcon, 2008. IEEE (pp. 222- 225).

[6] Lu, C. C., & Tseng, S. Y. (2002). Integrated design of AES (Advanced Encryption Standard) encrypter and decrypter. In Application-Specific Systems, Architectures and Processors, 2002. Proceedings. The IEEE International Conference on (pp. 277-285).

[7] Mohamed, A. A., & Madian, A. H. (2010, December). A Modified Rijndael Algorithm and it's Implementation using FPGA. In Electronics, Circuits, and Systems (ICECS), 2010 17th IEEE International Conference on (pp. 335-338).

[8] Pramstaller, N., Gurkaynak, F. K., Haene, S., Kaeslin, H., Felber, N., & Fichtner, W. (2004, September). Towards an AES crypto-chip resistant to differential power analysis. In Solid-State Circuits Conference, 2004. ESSCIRC 2004. Proceeding of the 30th European IEEE (pp. 307- 310).

[9] Deshpande, H. S., Karande, K. J., & Mulani, A. O. (2014, April). Efficient implementation of AES algorithm on FPGA. In Communications and Signal Processing (ICCSP), 2014 IEEE International Conference on (pp. 1895-1899).

[10] Nadeem, H (2006). A performance comparison of data encryption algorithms," IEEE Information and Communication Technologies, (pp. 84-89).

[11] Diaa, S., E, Hatem M. A. K., & Mohiy M. H. (2010, May) Evaluating the Performance of Symmetric Encryption Algorithms. International Journal of Network Security, Vol.10, No.3, (pp.213-219).

[12] Jain, R., Jejurkar, R., Chopade, S., Vaidya, S., & Sanap, M. (2014). AES Algorithm Using 512 Bit Key Implementation for Secure Communication. International journal of innovative Research in Computer and Communication Engineering, 2(3).

[13] Selmane, N., Guilley, S., & Danger, J. L. (2008, May). Practical setup time violation attacks on AES. In Dependable Computing Conference, 2008. EDCC 2008. Seventh European (pp. 91-96). IEEE.

[14] Berent, A. (2013). Advanced Encryption Standard by Example. Document available at URL http://www. networkdls. com/Articles/AESbyExample. pdf (April 1 2007) Accessed: June.

[15] Benvenuto, C. J. (2012). Galois field in cryptography. University of Washington