

Cryptography Project

**Academic Year : 2024-25**

**Submitted By :**

|  |  |  |
| --- | --- | --- |
| Ishika Sahu | BT22GCS186 | B. Tech CSE |
| Dhruv Temura | BT22GCS191 | B. Tech CSE |
| Drishti Joshi | BT22GCS093 | B. Tech CSE |
| Abhi Chowdary | BT22GCS283 | B. Tech CSE |
| Indurthi Sai Sowgandh | BT22GCS356 | B. Tech CSE |
| Himanshu | BT22GCS071 | B. Tech CSE |

**Contents**

[Abstract 3](#_TOC_250019)

1. [Introduction 3](#_TOC_250018)
   1. [Literature Review 3](#_TOC_250017)
   2. [Contribution 4](#_TOC_250016)
2. [Algorithm Keywords 4](#_TOC_250015)
3. [Algorithm Overview 5](#_TOC_250014)
   1. [AES Pseudocode 5](#_TOC_250013)
   2. [Mathematical Basis 6](#_TOC_250012)
   3. [Four Core Stages in Each Round 7](#_TOC_250011)
      1. [SubBytes (Byte Substitution) 7](#_TOC_250010)
      2. [ShiftRows (Row Shifting) 7](#_TOC_250009)
      3. [MixColumns (Column Mixing) 7](#_TOC_250008)
      4. [AddRoundKey (Key Addition) 8](#_TOC_250007)
   4. [Key Expansion (Key Schedule) 8](#_TOC_250006)
      1. [Key Expansion Algorithm 9](#_TOC_250005)
   5. [Core Structural Elements and AES Security 10](#_TOC_250004)
   6. [AES Encryption and Decryption Process 10](#_TOC_250003)
4. [Conclusion 12](#_TOC_250002)
5. [References 12](#_TOC_250001)

[Plagiarism Report 13](#_TOC_250000)

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Figure Number** | **Name** | **Page Number** |
| **Fig 1** | AES Structure | 6 |
| **Fig 2** | The Key Expansion | 10 |
| **Fig 3** | AES Encryption and Decryption | 11 |

# Abstract

Advanced Encryption Standard is one of the most widely used symmetric block ciphers in existence. The designers, Joan Daemen and Vincent Rijmen, have built a block size of 128 bits and support has been provided for key length as 128, 192, and 256 bits. This consisted of the substitution-permutation network and the method of key mixing for several rounds to provide excellent data confidentiality and resistance in the cryptographic domain to various attacks. Its fundamental transformational blocks consist of subbytes, shift rows, mix columns, and add round keys, which are blocks that provide diffusion and nonlinearity, ensuring complicated transformations improve security. Due to the very high robustness and flexibility, AES has become the standard encryption protocol that is broadly applied to protect sensitive information from any application, whether governmental and financial sectors or personal transactions. In the paper, we are discussing the structure and strength of AES, based on which it serves as a standard for reliable and secure digital communication and data storage.

# Introduction

AES is a block cipher widely trusted and extensively used for encrypting sensitive data. The cipher has fixed blocks of 128 bits and keys of 128, 192, or 256 bits that process them in multiple rounds of substitution, permutation, and key mixing in a way that offers strong confidentiality against modern attacks. AES is known for its efficiency, which makes high-performance operation possible with minimal resource consumption. This has made it suitable for both powerful systems and everyday devices, such as smartphones. Its versatility makes it suitable for use in numerous applications, including online banking that secures funds transactions; secure messaging platforms that guard private communications; the e-commerce sites depend on it for safe payment processing; and government systems safeguarding classified information. More importantly, AES provides protection of data in cloud storage both at rest and in transit, ensuring maximum security. Its reliability, efficiency, and practicality have solidified its role as the cornerstone of modern data security for software and hardware applications.

## Literature Review

The authors Vincent Rijmen et. al. In the year 2001 proposed The Design of Rijndael[1] details the AES algorithm's structure with core transformations (SubBytes, ShiftRows, MixColumns, and AddRoundKey). AES, the Advanced Encryption Standard which is defined by NIST’s FIPS 197[3], majorly adopted the symmetric block cipher designed to secure electronic data. The paper was established in 2001 and updated in 2023, DES got replaced by the AES due to security concerns. In the paper of 2017, Advanced Encryption Standard Algorithm to Encrypt and Decrypt Data[4], author Ako Muhammas Abdullah explores the AES Algorithm along with the concept introduced by NIST. He also compared AES with other algorithms such DES, 3DES, and Blowfish which demonstrated AES's superior performance and security.

## Contribution

In this paper we explained the Advanced Encryption Standard Algorithm (AES), detailing its comprehensive approach. Below are our key contributions :

* + - We dissect the AES algorithm by step, showing in detail that it's easy to understand why each key process - SubBytes, ShiftRows, MixColumns, AddRoundKey, and Key Expansion - is considered easy to understand.
    - We have drawn diagrams of the AES structure and operations including encryption and decryption, rounding, and key expansions. These help elucidate long, difficult concepts in a more easily understood form.

# Algorithm Keywords

|  |  |  |
| --- | --- | --- |
| 1. | **AES** | Advanced Encryption Standard. |
| 2. | **Affine** | A process that consists of multiplication by a matrix, followed transformation by the addition of a vector. |
| 3. | **Bit** | A binary digit, 0 or 1. |
| 4. | **Block** | A fixed length sequence of bits. |
| 5. | **Byte** | A sequence of eight bits. |
| 6. | **Key** | Parameter of a block cipher that determines selection of a permutation from the family of block ciphers. |
| 7. | **Round** | Transformations of the state in the specifications for CIPHER(), INVCIPHER(), and EQINVCIPHER(). The process consists of four transformations, except for the last iteration, where one of the transformations is skipped. |
| 8. | **S-box** | It is a non-linear substitution look up table that is used in SUBBYTES() and KEY EXPANSION() in order to perform one-to-one substitution of a byte. |

# Algorithm Overview

AES is a Symmetric Block Cipher that processes 128 bit data blocks with keys of sizes 128, 192, or 256 bits in length. The encrypting process follows the concept of generating round keys from the cipher key and utilizing them for several rounds of transformations within the process.

1. **KeyExpansion:** This generates the round keys for AES using the cipher key.
2. **Initial AddRoundKey:** In this, the first round key is added to the data.
3. **Main Rounds:** All the rounds except the last round are composed of the operations SubBytes, ShiftRows, MixColumns, and AddRoundKey.
4. **Final Round:** The MixColumns transformation except for the purpose of the decryption.

The number of rounds in relation to the size of the key—10 rounds for 128-bit, 12 for 192-bit, and 14 for 256-bit keys.

## AES Pseudocode

Input = 128-bit Plaintext Block, 128/192/256-bit Cipher Key Output: 128-bit Ciphertext Block

Begin

KeyExpansion(Key) #Generate round keys AddRoundKey(State, RoundKey[0]) #Initial round key addition for Round = 1 to Nr-1 do #Main rounds

SubBytes(State) ShiftRows(State) MixColumns(State)

AddRoundKey(State, RoundKey[Round]) End for

SubBytes(State) #Final round (no MixColumns) ShiftRows(State)

AddRoundKey(State, RoundKey[Nr]) return State as Ciphertext

End

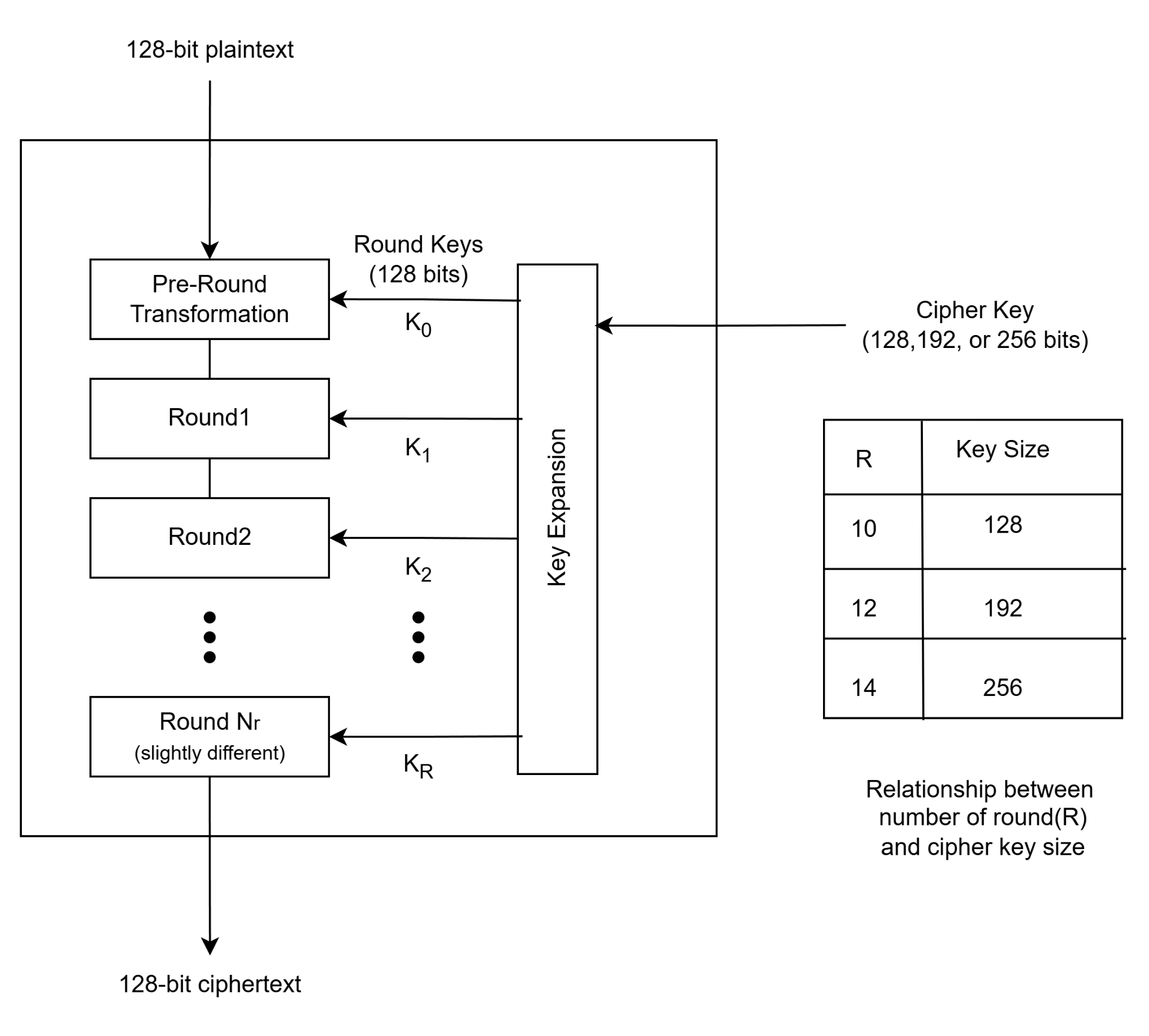


Fig 1. AES structure[1]

# Mathematical Basis

AES operates over a Galois field, specifically GF(28), to carry out the MixColumns transformation and other finite field operations. This mathematical foundation in finite fields ensures secure, repeatable transformations that create complex patterns, providing robust resistance to cryptographic attacks.

## Four Core Stages in Each Round

Each AES encryption round, includes four main transformations:

### SubBytes (Byte Substitution)

The SubBytes step is a nonlinear substitution operation for each byte in an AES state matrix utilizing a substitution box or S-Box. Each entry in this 16x16 look-up table maps an 8-bit input byte to an output byte that is different from the former. This gives the nonlinearity structure that makes both linear and differential cryptanalysis impossible. The S-Box is computed as a multiplicative inverse in the Galois Field GF(2^8), followed by an affine transformation, with perfect diffusion and security at design time.

In encryption, each byte of the state matrix is replaced with its S-Box value, which changes the data significantly, increasing the complexity and making it impossible to decrypt without the key. The S-Box resists popular attacks on cryptographic algorithms; with each substitution properly mapped to guarantee that each round will alter the data significantly while masking any patterns between ciphertext and plaintext so that S-Box is an important component adding confusion and enhancing AES security.

### ShiftRows (Row Shifting)

Here, the rows of the AES state matrix are rotated left by a number of bytes to inject diffusion between rows.

* + - * The first row is left intact.
      * The second row is shifted to the left by one byte.
      * The third row is shifted left to three bytes.
      * The fourth row is moved left by four bytes.

This mixing intermixes the bytes across columns so that, after further rounds, every byte in the state depends on bytes from several positions in the plaintext. It can be easily noticed that by this shifting of rows, AES achieves a type of diffusion where for each input bit, its influence extends over several output bytes, thus making deduction of the original text from compressed output impossible.

### MixColumns (Column Mixing)

This transformation over a mixing is applied to each column of the state matrix, and within such a column, each byte transforms into a new value with regard to all bytes within such a column. In this way, data gets "mixed" within every column.

This operation multiplies the matrices of Galois Field GF(28). In other words, each column is multiplied by the same fixed 4x4 matrix.

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | 3 | 1 | 1 |
| 1 | 2 | 3 | 1 |
| 1 | 1 | 2 | 3 |
| 3 | 1 | 1 | 2 |

This multiplication is performed modulo an irreducible polynomial, x8 + x4 + x3 + x + 1. This step introduces further diffusion with all changes to a single byte in one column influencing all bytes of that column, thereby potentially hindering some types of attacks by spreading the influence of each byte across the column.

### AddRoundKey (Key Addition)

This is the simplest but crucial step at every round of AES. Here, each byte of the state matrix is XORed with the corresponding byte of the round key. The round key generated in the Key Expansion step, discussed below, is different for every round. This implies that each stage of encryption is dependent not only on data transformations but also on key material.

The round key XOR introduces the secret key into the encryption process and makes the output unique for this key. Because XOR is a reversible operation, it follows that AES can maintain the symmetric way of encryption and decryption but still be efficient and secure. AddRoundKey takes place at the start of the first round and then after each round hence finishing up with transformations of each round.

## Key Expansion (Key Schedule)

AES has used a key schedule algorithm to expand the original key to derive a unique round key for each encryption round. This ensures security for the AES because it introduces variations in sub-keys for each round, hence cannot be subjected to direct attacks.

To start the key expansion process, an original encryption key is arranged into a 4 × 4 byte array known as the "state array". For a 128-bit encryption key, this is divided into four 32-bit words, termed as follows: w0, w1, w2, and w3. These words are used as the starting point of the key schedule.

The algorithm expands these initial words into a key schedule consisting of 44 words ( w0, w1 , , w43) for AES-128. Each group of four words constitutes a 128-bit round key.

1. **Initial Round Key:** The first four words (w0, w1, w2, and w3) are XORed with the input block before the first round of encryption starts.
2. **Subsequent Round Keys:** The remaining 40 words of the key schedule are used, four at a time, for the 10 encryption rounds.

### Key Expansion Algorithm

The key expansion proceeds iteratively, generating new words from the previous ones:

### Generating wi+4 from wi :

The first word of each new 4-word group (wi+4) is computed as: wi+4 = wi ⊕ g(wi+3). Here, g() is a function that incorporates diffusion and nonlinearity through the following steps:

* + - * **RotWord:** Perform one-byte left circular rotation on wi+3.
      * **SubWord:** Substitute each byte of the rotated word using the S-box, the same as used for the SubBytes step in AES rounds.
      * **XOR with Round Constant:** XOR the result with a round constant Rcon[i]. The round constant adds randomness at each round and is represented by (RC[i],0x00,0x00,0x00) where,
        + RC[1]=0x01
        + RC[j]=0x02×RC[j−1] in GF(28)\text{GF}(2^8)GF(28)

### Generating Remaining Words:

The remaining words in the new 4-word group are obtained by XORing the preceding word in the group with the corresponding word in the previous group wi+5=wi+4⊕wi+1, wi+6=wi+5⊕wi+2, wi+7=wi+6⊕wi+3

Number of Rounds and Key Lengths

* + - * AES-128: The key is expanded into 11 round keys of 128 bits in length. This is for the 10 rounds of encryption and one initial round key.
      * AES-192: The key will be expanded into 13 round keys to support 12 rounds.
      * AES-256: The key will be expanded into 15 round keys to support 14 rounds.

Every encryption round is using a different 128-bit round key, hence surely giving security by constantly varying the transformation applied to the plaintext.

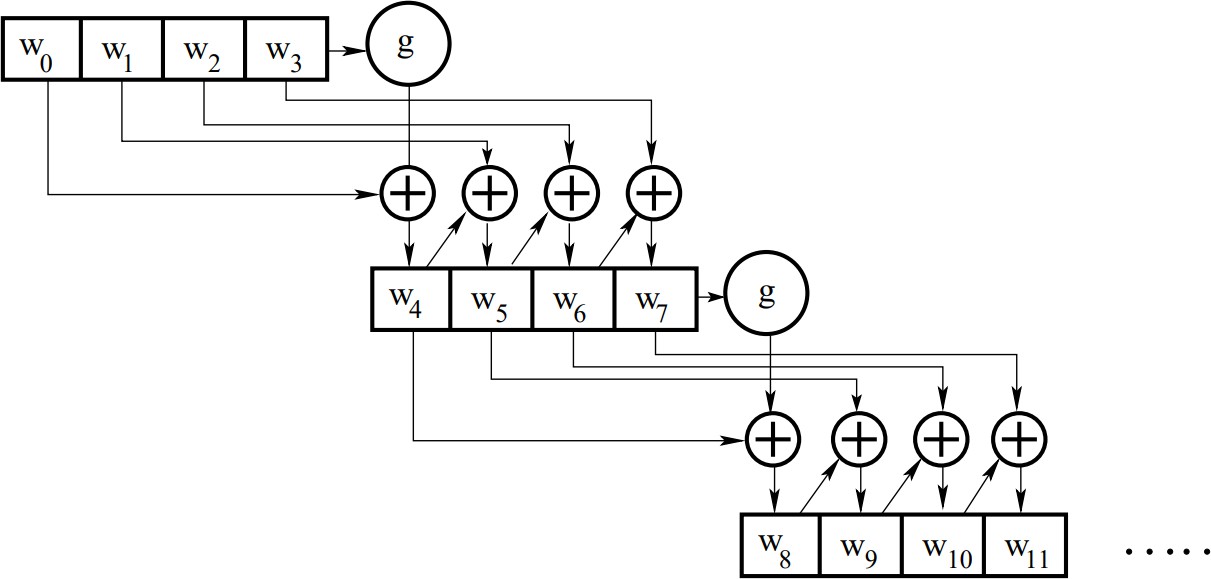


Fig 2. The key expansion[5]

## Core Structural Elements and AES Security

AES is efficient and secure. It has several rounds and each round complicates the encryption process. A round follows an AddRoundKey operation, which consists of nine rounds made up by four transformation stages: Substitute Bytes, Shift Rows, Mix Columns, and AddRoundKey. To ensure ideal invertibility both during encryption and decryption, the last round of AES does not apply the MixColumns step. Only the AddRoundKey step is sensitive to the key so it is essential to AES security, where that encryption key must be present at both initial and final stages. These combined transformations developed by AES through confusion by way of substitution, diffusion by shifting and mixing, and nonlinearity by the work of S-box combinations work in a tight tandem to bring about quite robust encryption that thwarts even brute force, differential, and linear cryptanalysis. Its non-Feistel structure where all data blocks are transformed within each round maximizes complexity as well as diffusion and brings effectiveness to AES not only in software but also in hardware, mainly for secure high-speed data protection.

## AES Encryption and Decryption Process

The AES encryption process initiates the first AddRoundKey with the plaintext. Then it undergoes a series of rounds, each of which is composed of four steps: Substitute Bytes, ShiftRows, MixColumns, and one more AddRoundKey.

* + - **Substitute Bytes:** each byte in the block is replaced by a value from a predefined substitution box S-box to introduce confusion.
    - **ShiftRows:** a cyclic left shift of rows of the block for better diffusion
    - **MixColumns:** Each column of the block mixed by a mathematical transformation for even more diffusion of information
    - **AddRoundKey:** The block is XORed with a round-specific key.

These rounds repeat for 9, 11, or 13 rounds according to the key size - 128, 192, or 256 bits. The last round omits the MixColumns step so only Substitute Bytes, ShiftRows, and AddRoundKey remain, and then the ciphertext is returned.

For decryption, it is paired with the encryption process-except that it has inverse operations. First, round keys have to be applied in reverse order followed by inverse transformations: Inverse ShiftRows, Inverse Substitute Bytes, and Inverse MixColumns, except in the last round. AddRoundKey is not affected, which means symmetric encryption and decryption since the operations have been reversed along with the key schedule. So, this design results in efficiency for both encryption and decryption while attaining high security through reversible steps.

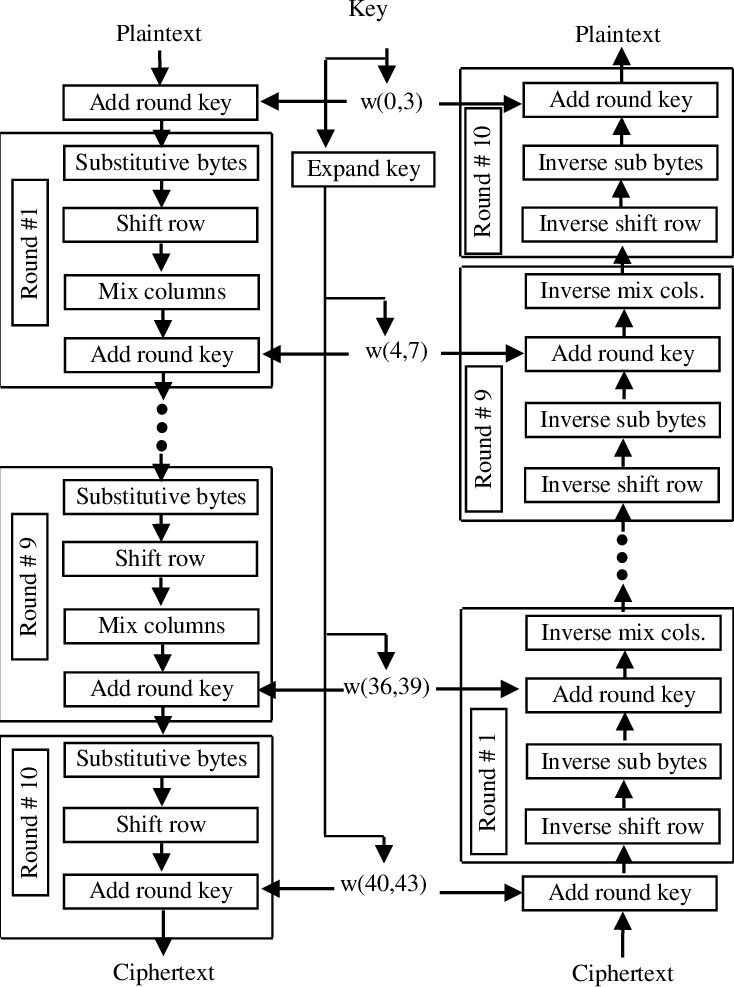


Fig 3. AES Encryption and Decryption[2]

# Conclusion

AES is the strongest symmetric key encryption algorithm; through its key expansion function, a substitution-permutation network, and several rounds of transformations that include SubBytes, ShiftRows, MixColumns, and AddRoundKey, it manages to produce a relatively complex-looking ciphertext from plaintext in a manner that would make unauthorized decryption very tough. The structure of AES, that is its dependency on key expansion and the arithmetic of the Galois field, makes it resistant to brute force and other typical cryptographic attacks. That is why AES has become the encryption standard in securing sensitive data for lots of applications, from financial transactions to government communications.

# References

1. [Daemen, Joan, and Vincent Rijmen. *The Data Encryption Standard.* In *The Design of*](https://link.springer.com/book/10.1007/978-3-662-04722-4)[*Rijndael.* Berlin: Springer, 2002.](https://link.springer.com/book/10.1007/978-3-662-04722-4)
2. [Stallings, William. *Cryptography and Network Security: Principles and Practice.* 8th ed.](https://mrce.in/ebooks/Cryptography%20%26%20Network%20Security%208th%20Ed.pdf) [London: Pearson, 2017.](https://mrce.in/ebooks/Cryptography%20%26%20Network%20Security%208th%20Ed.pdf)
3. [National Institute of Standards and Technology (NIST), Morris J. Dworkin, Elaine Barker,](https://www.nist.gov/publications/advanced-encryption-standard-aes) [James R. Nechvatal, James Foti, Lawrence E. Bassham, E. Roback, and James F. Dray Jr.](https://www.nist.gov/publications/advanced-encryption-standard-aes) [*Recommendation for Block Cipher Modes of Operation: Methods and Techniques.* NIST Special](https://www.nist.gov/publications/advanced-encryption-standard-aes) [Publication 800-38A, 2001.](https://www.nist.gov/publications/advanced-encryption-standard-aes)
4. [Abdullah, Ako. Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data.](https://www.researchgate.net/publication/317615794_Advanced_Encryption_Standard_AES_Algorithm_to_Encrypt_and_Decrypt_Data) [2017.](https://www.researchgate.net/publication/317615794_Advanced_Encryption_Standard_AES_Algorithm_to_Encrypt_and_Decrypt_Data)
5. [Avi Kak The Advanced Encryption Standard Purdue University, 2024](https://engineering.purdue.edu/kak/compsec/NewLectures/Lecture8.pdf)

# Plagiarism Report

