AES Encryption and Decryption Report

Introduction

This report outlines the implementation and operation of the Advanced Encryption Standard (AES) algorithm. AES is a symmetric key encryption algorithm widely used for securing sensitive data. This report discusses the steps taken in both encryption and decryption processes, along with an explanation of each step.

AES Algorithm Overview

AES operates on 128-bit data blocks and supports key sizes of 128, 192, and 256 bits. The algorithm consists of several key steps, including SubBytes, ShiftRows, MixColumns, AddRoundKey, and KeyExpansion.

Handling Plaintext Data

Selection of Plaintexts:

Three plaintext strings were chosen for demonstration purposes.
Plaintexts include various lengths and content to showcase the flexibility of the AES algorithm.

Conversion to Hexadecimal:

Each plaintext string is converted into its hexadecimal representation using Python's .encode().hex() method.
This conversion ensures uniformity and compatibility with AES, which operates on binary data.

Padding:

Plaintext is divided into 128-bit (16-byte) blocks, as AES operates on blocks of this size.
If the last block is not complete (less than 128 bits), it is padded to reach the required length.
Padding is crucial to ensure that all plaintext blocks are of equal size for encryption.

Encryption Process

Block Encryption:

Each plaintext block (after conversion and padding) is encrypted individually using AES.
The AES encryption process involves multiple rounds of SubBytes, ShiftRows, MixColumns, and AddRoundKey operations.
The final encrypted ciphertext block is obtained for each plaintext block.
Combining Blocks:

Encrypted ciphertext blocks are combined to form the complete ciphertext.
The combined ciphertext represents the encrypted form of the original plaintext.

Decryption Process

Block Decryption:

Encrypted ciphertext is decrypted block by block using the AES decryption process.
Each ciphertext block is decrypted individually, revealing the original plaintext block.
Combining Blocks:

Decrypted plaintext blocks are combined to reconstruct the original plaintext.
The combined plaintext represents the decrypted form of the original ciphertext.

Steps Taken in Encryption

Key Expansion: The provided key is expanded into a set of round keys, one for each round of encryption.
Initial Round: The plaintext is XORed with the first round key.
Main Rounds:
SubBytes: Each byte of the state is substituted with a corresponding byte from the S-box.
ShiftRows: The rows of the state are shifted cyclically to the left.
MixColumns: Each column of the state is multiplied with a fixed polynomial.
AddRoundKey: The round key is XORed with the state.
Final Round: Similar to the main rounds but without the MixColumns step.

Steps Taken in Decryption

Key Expansion: Same as in encryption, the key is expanded into a set of round keys.

Initial Round: The ciphertext is XORed with the last round key.
Main Rounds (in reverse order):
InverseShiftRows: Inverse operation of ShiftRows.
InverseSubBytes: Inverse operation of SubBytes.
AddRoundKey: The round key is XORed with the state.
InverseMixColumns: Inverse operation of MixColumns.
Final Round: Similar to the initial round but using the first round key.

Logging

Throughout both encryption and decryption processes, the state of the data after each round is logged into a file named 'log.txt'. This log provides insights into how the data evolves at each step.

Conclusion

AES is a robust encryption algorithm providing confidentiality and integrity to data. By following a series of well-defined steps and utilizing round keys derived from the initial key, AES ensures secure encryption and decryption of data.
2^{128} complexity to break using brute force

Samples:

```
"This system is made Aditya & Himanshu",
"This project is for AES-128 which takes a 128bit key and has 10 rounds",
"This is a test string of arbitrary length. Let's see how AES handles it!"
```

Output: Logged in file