Implementation of AES Encryption Algorithm

Supervisor	Assignment by,
Dr. Narendran Rajagopalan,	Parupati Abhinav
Associate Professor,	(CS22B1041)
NITPY.	

Overview:

The AES (Advanced Encryption Standard) algorithm is a symmetric key encryption algorithm widely used for securing data in applications ranging from file encryption to secure communications. This repository provides an implementation of AES in C, showcasing its functionality and potential integration into projects requiring cryptographic operations.

Methodology:

The methodology used in this AES implementation follows a structured approach adhering to the AES standard:

Key Size: A 256-bit key is used for encryption and decryption, providing a high level of security.

Input Division into Blocks: The input data is divided into 16-byte (128-bit) blocks, as required by the AES standard. If the input is not a multiple of 16 bytes, padding is likely applied.

Key Expansion: The 256-bit key is expanded into multiple round keys using a key schedule algorithm. These round keys are used in each round of encryption and decryption.

Data Transformation: The encryption process involves several transformations performed on each 16-byte block:

- **SubBytes Transformation:** A non-linear substitution step where each byte is replaced using a predefined substitution box (S-Box).
- **ShiftRows Transformation:** A transposition step where bytes in each row of the state are cyclically shifted to the left.
- **MixColumns Transformation:** A mixing operation combining bytes within each column of the state to produce new values.
- **AddRoundKey:** A bitwise XOR operation between the state and the round key.

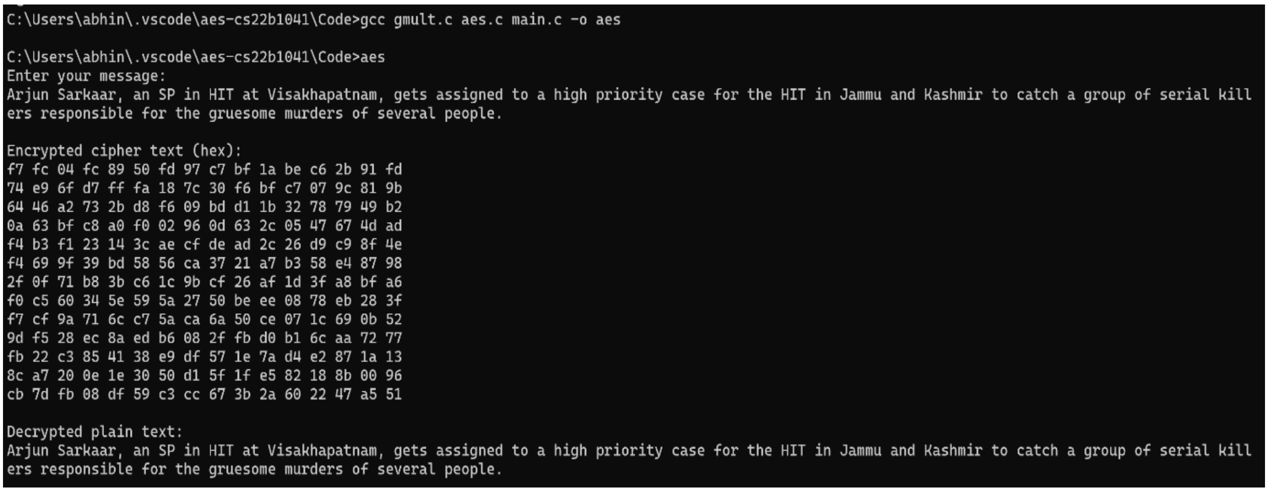
Rounds: For a 256-bit key, the algorithm applies 14 rounds of transformation. Each round consists of SubBytes, ShiftRows, MixColumns, and AddRoundKey steps. The final round omits the MixColumns step.

Decryption: Decryption reverses the encryption process using the inverse of the transformations: InvSubBytes, InvShiftRows, InvMixColumns, and AddRoundKey.

Applications :

- **Secure Communications:** Encrypting messages for secure transmission.
- **File Encryption:** Protecting sensitive files from unauthorized access.
- **Embedded Systems:** Implementing AES in constrained environments for secure operations.

Result :



```
C:\Users\abhin\.vscode\aes-cs22b1041\Code>gcc gmult.c aes.c main.c -o aes

C:\Users\abhin\.vscode\aes-cs22b1041\Code>aes
Enter your message:
Arjun Sarkaar, an SP in HIT at Visakhapatnam, gets assigned to a high priority case for the HIT in Jammu and Kashmir to catch a group of serial kill
ers responsible for the gruesome murders of several people.

Encrypted cipher text (hex):
f7 fc 04 fc 89 50 fd 97 c7 bf 1a be c6 2b 91 fd
74 e9 6f d7 ff fa 18 7c 30 f6 bf c7 07 9c 81 9b
64 46 a2 73 2b d8 f6 09 bd d1 1b 32 78 79 49 b2
0a 63 bf c8 a0 f0 02 96 0d 63 2c 05 47 67 4d ad
f4 b3 f1 23 14 3c ae cf de ad 2c 26 d9 c9 8f 4e
f4 69 9f 39 bd 58 56 ca 37 21 a7 b3 58 e4 87 98
2f 0f 71 b8 3b c6 1c 9b cf 26 af 1d 3f a8 bf a6
f0 c5 60 34 5e 59 5a 27 50 be ee 08 78 eb 28 3f
f7 cf 9a 71 6c c7 5a ca 6a 50 ce 07 1c 69 0b 52
9d f5 28 ec 8a ed b6 08 2f fb d0 b1 6c aa 72 77
fb 22 c3 85 41 38 e9 df 57 1e 7a d4 e2 87 1a 13
8c a7 20 0e 1e 30 50 d1 5f 1f e5 82 18 8b 00 96
cb 7d fb 08 df 59 c3 cc 67 3b 2a 60 22 47 a5 51

Decrypted plain text:
Arjun Sarkaar, an SP in HIT at Visakhapatnam, gets assigned to a high priority case for the HIT in Jammu and Kashmir to catch a group of serial kill
ers responsible for the gruesome murders of several people.
```

Conclusion :

The AES implementation in this repository is an excellent resource for understanding and utilizing symmetric encryption in C. It has practical applications in enhancing data security and serves as a foundation for any of further cryptographic projects.