AES Extension from 128-bits to 192-bits & 256-bits

MSCS\_630L\_231\_16S

Security Algorithms & Protocols

Created By

Dixita Sharegar

Saranya Radhakrishnan

# Introduction

AES i.e. Advanced Encryption Standard is a specification by NIST to encrypt and decrypt electronic data established by the National Institute of Standards and Technology. It is a block cipher which uses a block length of 128 bits. AES allows for 3 different key lengths 128,192 and 256. AES is symmetric i.e. it uses the same secret key for encryption & decryption, both sender & receiver should share this key. All key lengths are considered good enough to protect classified information up to the “Top Secret” level.

AES 128 bit encryption was accomplished in the previous lab work for this course, which serves as a base for the subsequent decryption techniques implemented as mentioned in this paper. The tasks to be accomplished included:

* Encryption using 192 and 256 bit keys.
* Decryption for 128, 192 and 256 bit keys.
* Padding scheme.

# Program Execution

## Encryption

Encryption technique used for 192 and 256 bits was same as the one used for 128 bit. Common functions were moved to a new class called GlobalObjects. Here based on the key size, operations were performed on the data. Functions like printing

## Decryption

Decryption technique was also the same for 128,192 and 256 bits. The round keys were used in the reverse order while decrypting data.

## GlobalObjects

This file contains common constants, variables and functions that are used by both encryption and decryption classes. It contains common functions like stateXor, nibble substitution, mix column, print data etc.

## Driver

This file contains the main class, it accepts the key and input data. It first checks if the data is in hex format and then splits the input text into blocks of 32 bytes. Each block is padded if the block length is less than 32 bytes with ‘&’ followed by length to be padded in hex. The padded text is sent for encryption, which is then given as input to the decryption method. Padded bits are removed from the decrypted text and then printed on screen.

## Padding

If block size is not a multiple of 32, then it needs to be padded. The end of the string is padded with hex value of ‘&’ followed by number of bytes to pad.

# Input & Output

Ten test cases are created with key lengths varying from 128, 192 and 256 bits. It contains varying length input plaintext so that padding can be tested.

# Suggested Authentication Mechanism

# Conclusion

# References