EX. NO: 4

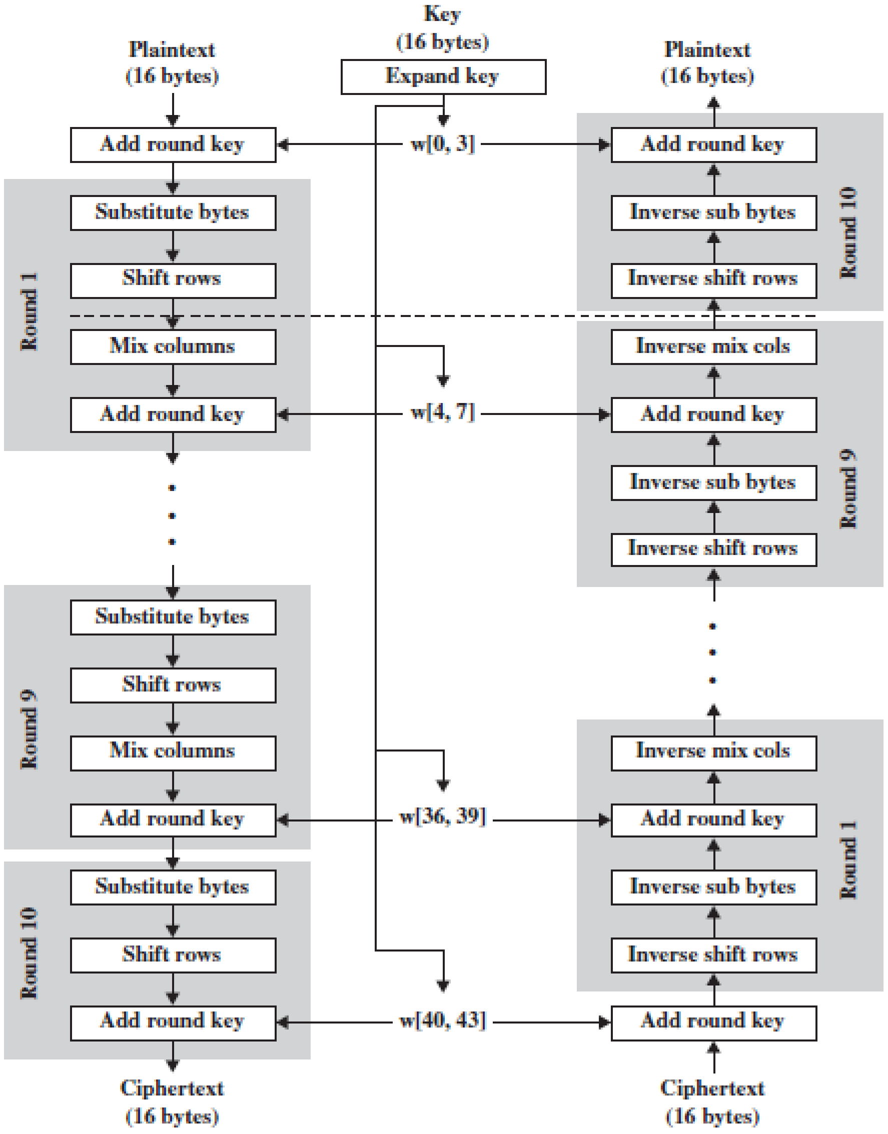
IMPLEMENTATION OF AES

AIM:

To write a program to implement Advanced Encryption Standard (AES)

DESCRIPTION:

The cipher takes a plaintext block size of 128 bits, or 16 bytes. The key length can be 16, 24, or 32 bytes (128, 192, or 256 bits). The algorithm is referred to as AES-128, AES-192, or AES-256, depending on the key length.



(a) Encryption (b) Decryption

The input to the encryption and decryption algorithms is a single 128-bit block. In FIPS PUB 197, this block is depicted as a 4 * 4 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 matrix. Similarly, the key is depicted as a square matrix of bytes. This key is then expanded into an array of key schedule words. Each word is four bytes, and the total key schedule is 44 words for the 128-bit key. Note that the ordering of bytes within a matrix is by column. So, for example, the first four bytes of a 128-bit plaintext input to the encryption cipher occupy the first column of the **in** matrix, the second four bytes occupy the second column, and so on. Similarly, the first four bytes of the expanded key, which form a word, occupy the first column of the **w** matrix.

The cipher consists of *N* rounds, where the number of rounds depends on the key length: 10 rounds for a 16-byte key, 12 rounds for a 24-byte key, and 14 rounds for a 32-byte key. The first *N* - 1 rounds consist of four distinct transformation functions: SubBytes, ShiftRows, MixColumns, and AddRoundKey. The final round contains only three transformations, and there is a initial single transformation (AddRoundKey) before the first round, which can be considered Round 0. Each transformation takes one or more 4 * 4 matrices.

PROGRAM:

```
    package com.includehelp.stringsample;

    import java.util.Base64; import java.util.Scanner; import
    javax.crypto.Cipher;
    import javax.crypto.spec.IvParameterSpec; import
    javax.crypto.spec.SecretKeySpec;

    /**
     * Program to Encrypt/Decrypt String Using AES 128 bit Encryption
     Algorithm
     */
    public class EncryptDecryptString
    {
          private static final String encryptionKey = "ABCDEFGHIJKLMNOP";
          private static final String characterEncoding         = "UTF-8";
          private static final String cipherTransformation=
          "AES/CBC/PKCS5PADDING"; private static final String
          aesEncryptionAlgorithem = "AES";
      /**
      * Method for Encrypt Plain String Data
      * @param plainText
      * @return encryptedText
      */
          public static String encrypt(String plainText)
          {
                String encryptedText = ""; try
                {
                   Cipher cipher = Cipher.getInstance(cipherTransformation);
                   byte[] key= encryptionKey.getBytes(characterEncoding);
                   SecretKeySpec secretKey = new SecretKeySpec(key,
                   aesEncryptionAlgorithem); IvParameterSpec ivparameterspec
                   = new IvParameterSpec(key);
                   cipher.init(Cipher.ENCRYPT_MODE, secretKey,
                   ivparameterspec);
byte[] cipherText = cipher.doFinal(plainText.getBytes("UTF8"));
Base64.Encoder encoder = Base64.getEncoder();
encryptedText = encoder.encodeToString(cipherText);
} catch (Exception E)
{
System.err.println("Encrypt Exception : "+E.getMessage());
}
return encryptedText;
}
         public static String decrypt(String encryptedText)
         {
               String decryptedText = ""; try
               {
                     Cipher cipher =
                     Cipher.getInstance(cipherTransformation); byte[] key =
                     encryptionKey.getBytes(characterEncoding);
                   SecretKeySpec secretKey = new SecretKeySpec(key,
                     aesEncryptionAlgorithem); IvParameterSpec
                     ivparameterspec = new IvParameterSpec(key);
                     cipher.init(Cipher.DECRYPT_MODE, secretKey,
                     ivparameterspec); Base64.Decoder decoder =
                     Base64.getDecoder();
                     byte[] cipherText =
                     decoder.decode(encryptedText.getBytes("UTF8"));
                     decryptedText = new String(cipher.doFinal(cipherText),
                     "UTF-8");
               } catch (Exception E)
               {
                     System.err.println("decrypt Exception :
                     "+E.getMessage());
               }
               return decryptedText;
         }

         public static void main(String[] args)
         {
               Scanner sc = new Scanner(System.in); System.out.println("Enter
               String : "); String plainString = sc.nextLine();

               String encyptStr = encrypt(plainString); String decryptStr =
               decrypt(encyptStr);
               System.out.println("Plain     String     :     "+plainString);
               System.out.println("Encrypt     String     :     "+encyptStr);
               System.out.println("Decrypt String : "+decryptStr);
}
```

OUTPUT:

```
Enter String : Hello World
Plain String : Hello World
Encrypt String : IMfL/ifkuvkZwG/v2bn6Bw==
Decrypt String : Hello World
```

VIVA QUESTIONS (PRELAB and POSTLAB):

1.AES follows

a)Hash Algorithm b)Caesars Cipher c)Feistel Cipher Structure d)SP Networks

2. The AES Algorithm Cipher System consists of _______ _ rounds (iterations) each with a round key

a) 12 b) 18 c) 9 d) 16

3. The AES algorithm has a key length of

a) 128 Bits b) 32 Bits c) 64 Bits d) 16 Bits

4. In the AES algorithm, although the key size is 64 bits only 48bits are used for the encryption procedure, the rest are parity bits.

a) True b) False

5. In the AES algorithm the round key is ______ ___bit and the Round Input is ____ bits.

a) 48, 32 b) 64,32 c) 56, 24 d) 32, 32

6. In the AES algorithm the Round Input is 32 bits, which is expanded to 48 bits via _____ _ __

a) Scaling of the existing bits b) Duplication of the existing bits

c) Addition of zeros d) Addition of ones

7. The Initial Permutation table/matrix is of size

a) 16×8 b) 12×8 c) 8×8 d) 4×8

8. The number of unique substitution boxes in AES after the 48 bit XOR operation are a) 8 b) 4 c) 6 d) 12

9. In the AES algorithm the 64 bit key input is shortened to 56 bits by ignoring every 4th bit.

a) True b) False

RESULT:

The program to implement AES encryption technique was developed and executed successfully.