MALLA REDDY UNIVERSITY

III Year B.Tech– II Semester **L/T/P/C**
-/-/3/1.5

(MR20-1CS0188) CRYPTOGRAPHY & NETWORK SECURITY LAB

Name: Subhapreet Patro
Roll No.: 2211CS010547
Group: 3

WEEK-4

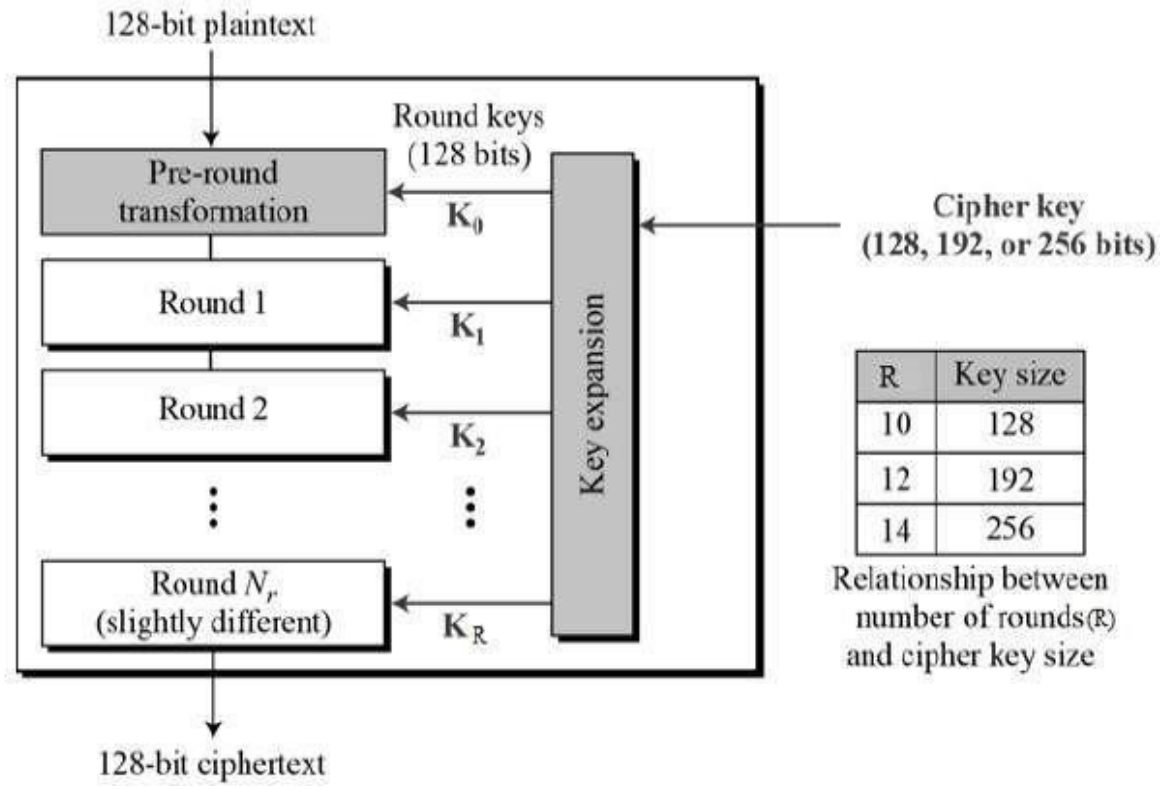
4 . User A want to send message "Welcome to CSE DEPT" to user B by using AES algorithms encrypt it and decrypt it at receiver end.

AIM: To apply Advanced Encryption Standard (AES) Algorithm for a practical application like URL Encryption.

OBJECTIVE: To understand the encryption and decryption the given message by using Advanced Encryption Standard encryption algorithm.

THEORY:

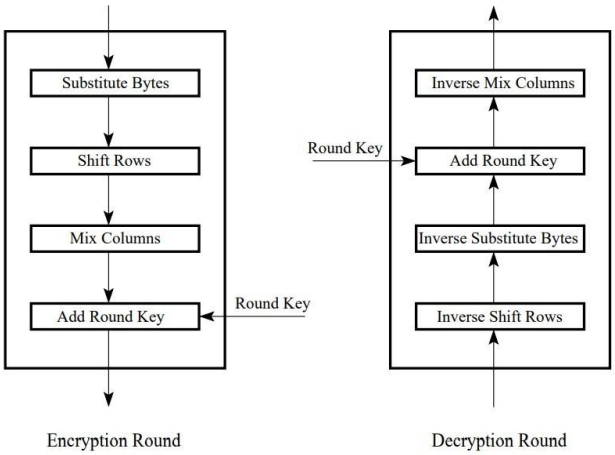
The Advanced Encryption Standard (AES), also called Rijndael, is a symmetric block cipher with a block length of 128 bits and support for key lengths of 128, 192, and 256 bits. It was published by NIST (National Institute of Standards and Technology) in 2001. Here, we assume a key length of 128 bits, which is likely to be the one most commonly implemented.



R	Key size
10	128
12	192
14	256

Relationship between number of rounds(R) and cipher key size

ALGORITHM:



STEP-1: SubBytes for byte-by-byte substitution during the forward process. The corresponding substitution step used during decryption is called InvSubBytes.

STEP-2: ShiftRows for shifting the rows of the state array during the forward process. The corresponding transformation during decryption is denoted InvShiftRows for Inverse ShiftRow Transformation.

STEP-3: MixColumns for mixing up of the bytes in each column separately during the forward process. The corresponding transformation during decryption is denoted InvMixColumns and stands for inverse mix column transformation. The goal is here is to further scramble up the 128-bit input block.
STEP-4: AddRoundKey for adding the round key to the output of the previous step during the forward process. The corresponding step during decryption is denoted InvAddRoundKey for inverse add round key transformation.

PROGRAM:

```
import java.util.*;
import java.io.UnsupportedEncodingException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.Arrays;
import java.util.Base64;
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;

public class AES {
   private static SecretKeySpec secretKey;
   private static byte[] key;

   public static void setKey(String myKey) {
      MessageDigest sha = null;
      try {
         key = myKey.getBytes("UTF-8");
         sha = MessageDigest.getInstance("SHA-1");
         key = sha.digest(key);
         key = Arrays.copyOf(key, 16);
         secretKey = new SecretKeySpec(key, "AES");
      } catch (NoSuchAlgorithmException e) {
         e.printStackTrace();
      } catch (UnsupportedEncodingException e) {
         e.printStackTrace();
      }
   }

   public static String encrypt(String strToEncrypt, String secret) {
      try {
         setKey(secret);
```

```
            Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
            cipher.init(Cipher.ENCRYPT_MODE, secretKey);
            return
Base64.getEncoder().encodeToString(cipher.doFinal(strToEncrypt.getBytes("U
TF-8")));
        } catch (Exception e) {
            System.out.println("Error while encrypting: " + e.toString());
        }
        return null;
    }

    public static String decrypt(String strToDecrypt, String secret) {
        try {
            setKey(secret);
            Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5PADDING");
            cipher.init(Cipher.DECRYPT_MODE, secretKey);
            return new
String(cipher.doFinal(Base64.getDecoder().decode(strToDecrypt)));
        } catch (Exception e) {
            System.out.println("Error while decrypting: " + e.toString());
        }
        return null;
    }

    public static void main(String[] args) {
        String secretKey, originalString;
        System.out.println("Enter the secret key: ");
        Scanner scn = new Scanner(System.in);
        secretKey = scn.nextLine();
        System.out.println("Enter the original Message: ");
        originalString = scn.nextLine();
        String encryptedString = AES.encrypt(originalString, secretKey);
        String decryptedString = AES.decrypt(encryptedString, secretKey);
        System.out.println("Message Encryption Using AES Algorithm\n --------");
        System.out.println("Original Message: " + originalString);
        System.out.println("Encrypted Message: " + encryptedString);
        System.out.println("Decrypted Message: " + decryptedString);
        scn.close();
    }
}
```

OUTPUT:

```
C:\Engineering Third Year\Semester 6\Cryptography and Network Security\2211cs010547\Week-4>java AES
Enter the secret key:
secret
Enter the original Message:
```

```
C:\Engineering Third Year\Semester 6\Cryptography and Network Security\2211cs010547\Week-4>java AES
Enter the secret key:
secret
Enter the original Message:
Welcome to CSE DEPT
Message Encryption Using AES Algorithm
 ----------
Original Message: Welcome to CSE DEPT
Encrypted Message: OMH1q4EBwCsTrdpgUItKFnZua5n6f7fBsXWRuhYAPds=
Decrypted Message: Welcome to CSE DEPT

C:\Engineering Third Year\Semester 6\Cryptography and Network Security\2211cs010547\Week-4>
```