SANJANA K

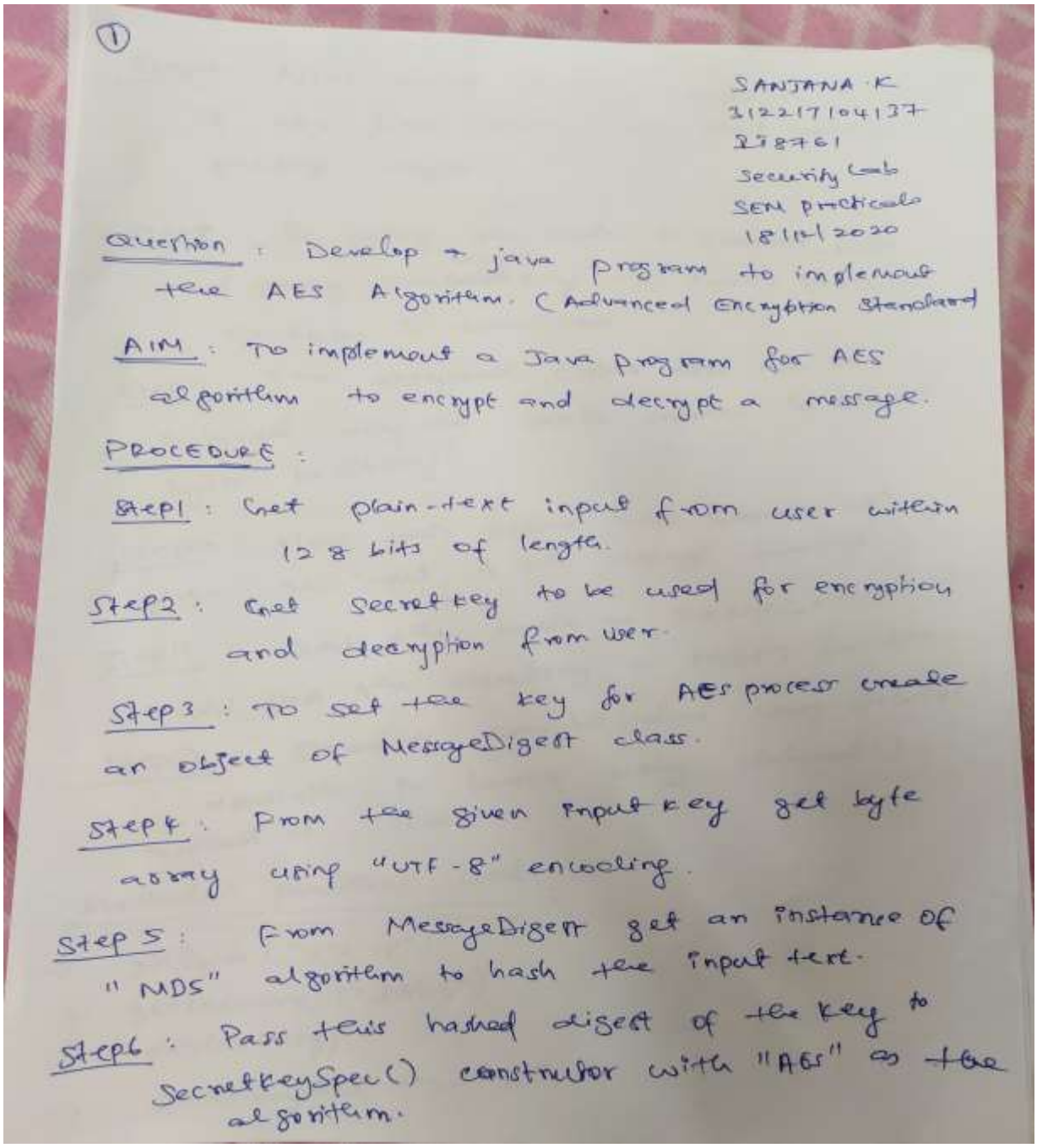
312217104137

IT8761-SECURITY LAB

SEMESTER PRACTICALS

18/12/2020

AIM PROCEDURE AND RESULT:



① SANJANA K 312217104137 IT8761 Security Lab SEM practicals 18/12/2020

Question : Develop a java program to implement the AES Algorithm. (Advanced Encryption Standard)

AIM : To implement a Java program for AES algorithm to encrypt and decrypt a message.

PROCEDURE :

Step1 : Get plain-text input from user within 128 bits of length.

Step2 : Get secret key to be used for encryption and decryption from user.

Step3 : To set the key for AES process create an object of MessageDigest class.

Step4 : From the given input key get byte array using "UTF-8" encoding.

Step5 : From MessageDigest get an instance of "MD5" algorithm to hash the input text.

Step6 : Pass this hashed digest of the key to SecretKeySpec() constructor with "AES" as the algorithm.

Step7: After setting the key create an instance of AES from Cipher class that belongs to package crypto.

Step8: By setting the mode to ENCRYPT_MODE and passing the secret key to Init() function encryption is established.

Step9: The encrypted cipher text can be returned using the Base64 encoder to convent bytes to String.

Step10: Now this cipher text should be given as input to the decrypt function.

Step11: Set the mode as DECRYPT_MODE and pass the secretkey to init() function.

Step12: Decoding can be done using decode() available in Base64 using doFinal() method of cipher.

③

SAMPLE INPUT / OUTPUT:

Enter plain text: SSnce

Enter Secret key: annauniversity

Cipher text: 9ysh WM2J krc 4h bH7JJy G9KA==

Plain text after decryption: SSnce.

METHODS USED:

1. getBytes("UTF-8") - Encoding format used
2. MessageDigest.getInstance("MD5") - to use hashing
3. digest(key) - TO convert key to 128 bits
4. SecretKeySpec(key, "AES") - to generate secret key for AES using constructor of SecretKeySpec class
5. Cipher.getInstance("AES/ECB/PKCS5PADDING") - to get instance of AES with padding if needed
6. cipher.init(Cipher.ENCRYPT_MODE, secretKey) - to initialize cipher to encrypt mode.
7. cipher.init(cipher.DECRYPT_MODE, secretKey) - to initialize cipher to decrypt mode.

RESULT:

AES algorithm was successfully implemented using 128-bit key for encrypting the given text and decrypted successfully.

CODE:

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

public class Main{
  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("MD5");
  key = sha.digest(key);
  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("UTF-
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)
  {
    Scanner sc= new Scanner(System.in);
    System.out.println("\t\t\tAES ALGORITHM");
    String text,secretKey;
    System.out.println("Enter plain text: ");
    text = sc.nextLine();
    System.out.println("Enter secret key: ");
    secretKey = sc.nextLine();
    String cipherText = Main.encrypt(text, secretKey);
    String plainText = Main.decrypt(cipherText, secretKey) ;
    System.out.println("\nCipher Text:" + cipherText);
    System.out.println("\nPlain Text:" + plainText);
    sc.close();
}
}
```

OUTPUT:

AES ALGORITHM

Enter plain text:

ssnce

Enter secret key:

annauniversity

Cipher Text:9yshWMIJlrc4hbH7JyG9kA==

Plain Text:ssnce