Practice 5: MODERN SYMMETRIC ENCRYPTION (Cont.)

5.1 OVERVIEW

5.1.1 Introduction

- Lab 5: AES Encryption
- Practice time: class: 3 study hours, self-study: 3 study hours.
- Requirements: Students using Netbeans Software

5.1.2 Objective

- This course provides students with knowledge of cryptographic algorithms and how they are used in today's world.
- The content emphasizes the principles, topics, approaches, and problem solving related to the underlying technologies and architectures of the field.

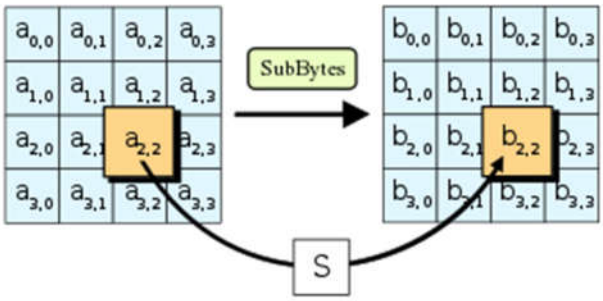
5.2 CONTENTS

5.2.1 Basic knowledge

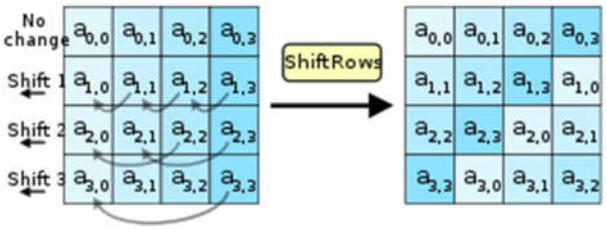
AES is based on a design principle known as a substitution–permutation network, and is efficient in both software and hardware. With a constant block size of 128 bits and a key size of 128, 192, or 256 bits, AES is a Rijndael variation. In contrast, the block and key sizes for Rijndael per se are stated as being any multiple of 32 bits, with a minimum of 128 and a maximum of 256 bits.

Description of the algorithm:

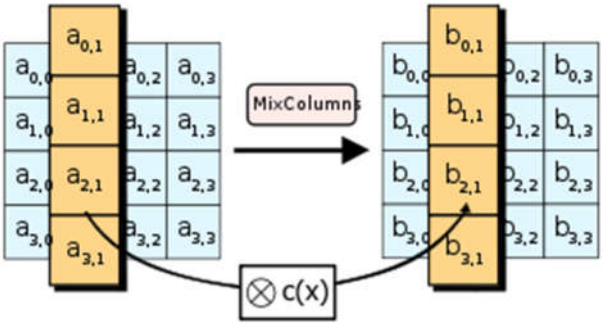
- AddRoundKey - each byte of the block is associated with keys, which are generated from the Rijndael key generation.
- SubBytes - here is allow (nonlinear) in each byte will be replaced by another byte according to the lookup table (Rijndael S-box).
- ShiftRows - the place of change, the rows in the loop are translated.
- MixColumns - too the composite works along the columns in the block according to a linear transformation allowed.



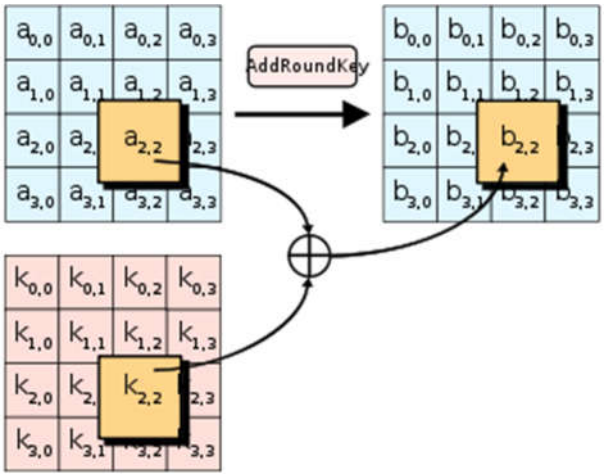
In the SubBytes step, each byte in the state is replaced with its entry in a fixed 8-bit lookup table, S; bij = S(aij).



In the ShiftRows step, bytes in each row of the state are shifted cyclically to the left. The number of places each byte is shifted differs incrementally for each row.



In the MixColumns step, each column of the state is multiplied with a fixed polynomial {\displaystyle c(x)}c(x).



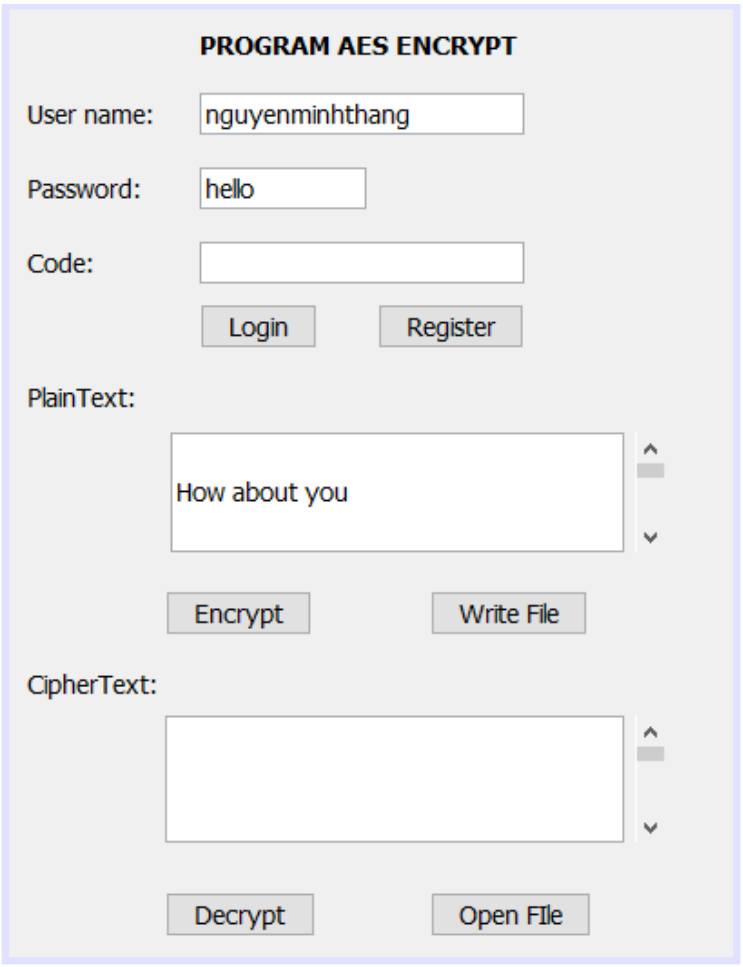
In the AddRoundKey step, each byte of the state is combined with a byte of the round subkey using the XOR operation (⊕).

5.2.2 AES Encrypt

Write a program to encrypt and decrypt text with AES encryption algorithm. The program can perform the following functions:

- Allow Register and Login.
- Allow text input into the system.
- Allows entering text protection keys.
- Allows to write File and open File.

❖ **Step 1: Design Form:**



❖ **Step 2: Write code for initialization function:**

```
public class b3_frmAES extends javax.swing.JFrame {

    public b3_frmAES() {
        initComponents();
    }
    private String user;
    private String pass;
    private String khoa;
    SecretKey secretKey;
    byte[] byteCipherText;
```

❖ **Step 3: Write an event handler function:**

3.1 Button Login

```
private void btnLoginActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        user= txtUser.getText();
        pass= txtPass.getText();
        khoa= user+pass;
        BufferedReader br = null;
        String fileName = "D:\\AES.txt";
        br = new BufferedReader(new FileReader(fileName));
        StringBuffer sb= new StringBuffer();
        char[] ca= new char[5];
        while (br.ready()) {
            int len=br.read(ca);
            sb.append(ca,0,len);
        }
        br.close();
        System.out.println("Key is :"+" "+sb);
        String chuoi=sb.toString();
        Boolean k=khoa.equals(chuoi);
        if(k==true) JOptionPane.showMessageDialog(null,"Login Sucessfull");
        else
            JOptionPane.showMessageDialog(null, "Login Fail");
        txtKey.setText(chuoi.getBytes().toString());
        KeyGenerator keyGen = KeyGenerator.getInstance("AES");
        keyGen.init(128);
        secretKey = keyGen.generateKey();

    } catch (NoSuchAlgorithmException ex) {

    } catch(Exception ex){}
}
```

3.2 button Register

```
private void btnRegisterActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        user= txtUser.getText();
        pass= txtPass.getText();
        khoa= user+pass;
        BufferedWriter bw = null;
        String fileName = "D:\\AES.txt";
        String s=txtPlainText.getText();
        bw = new BufferedWriter(new FileWriter(fileName));
        bw.write(khoa);
        bw.close();
        JOptionPane.showMessageDialog(null,"Registed Sucessfull. Login Please !!!");
        txtKey.setText(khoa.getBytes().toString());
    } catch (IOException ex) {
        Logger.getLogger(b3_frmAES.class.getName()).log(Level.SEVERE,null,ex);
    }
}
```

3.3 Button Encrypt:

```
private void btnEncryptActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        System.out.println("Create Key: "+ secretKey);
        Cipher aesCipher = Cipher.getInstance("AES");
        aesCipher.init(Cipher.ENCRYPT_MODE, secretKey);
        String strData = txtPlainText.getText();
        byte[] byteDataToEncrypt = strData.getBytes();
        byteCipherText = aesCipher.doFinal(byteDataToEncrypt);
        String strCipherText = new BASE64Encoder().encode(byteCipherText);
        System.out.println("Cipher Text generated using AES is "+ strCipherText);
        txtCipherText.setText(strCipherText);
    } catch (Exception ex) {
        System.out.println("Encrypt Error: "+ex);
    }
}
```

3.4 Button Decrypt:

```
private void btnDecryptActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        String cipherText = txtCipherText.getText();
        txtPlainText.setText(cipherText);
        Cipher aesCipher = Cipher.getInstance("AES");
        aesCipher.init(Cipher.DECRYPT_MODE, secretKey,aesCipher.getParameters());
        byte[] byteDecryptedText = aesCipher.doFinal(byteCipherText);
        String strDecryptedText = new String(byteDecryptedText);
        System.out.println("Decypted Text message is "+strDecryptedText);
        txtCipherText.setText(strDecryptedText);
    } catch (Exception ex) {
        System.out.println("Decrypt Error: "+ex);
    }
}
```

3.5 Button Write File:

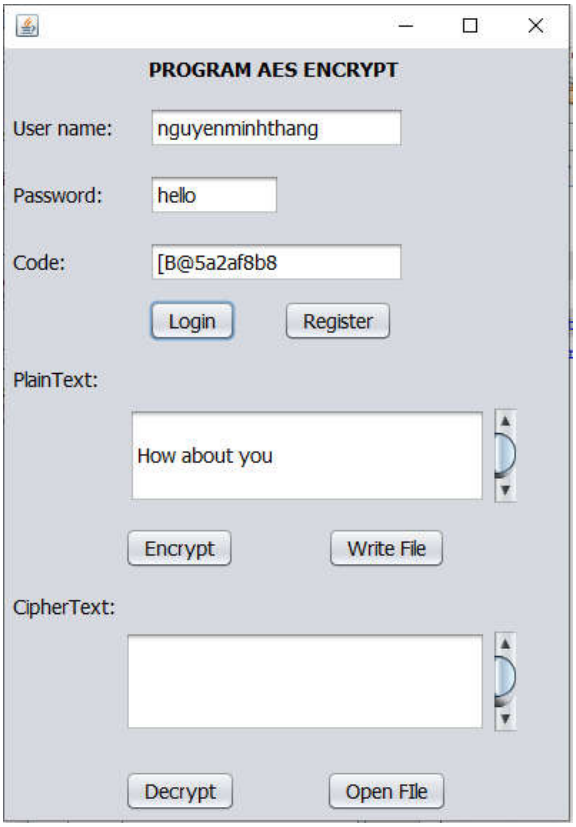
```
private void btnWriteFileActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        BufferedWriter bw = null;
        String fileName = "D:\\WriteAES.txt";
        String s = txtCipherText.getText();
        bw = new BufferedWriter(new FileWriter(fileName));
        bw.write(s);
        bw.close();
        JOptionPane.showMessageDialog(null,"Wrote File D:\\WriteAES.txt");
    } catch (IOException ex) {
        Logger.getLogger(b3_frmAES.class.getName()).log(Level.SEVERE,null,ex);
    }
}
```

3.6 Button Open File:

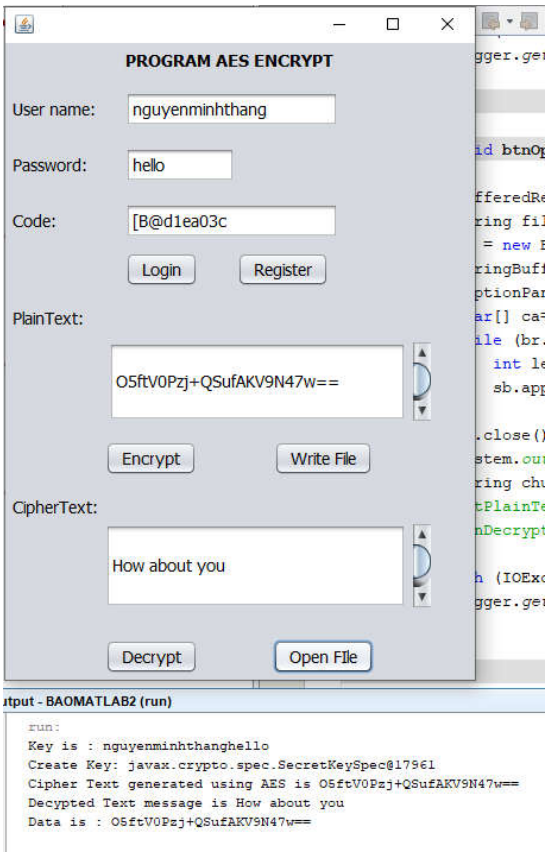
```
private void btnOpenFIleActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        BufferedReader br=null;
        String fileName = "D:\\WriteAES.txt";
        br = new BufferedReader(new FileReader(fileName));
        StringBuffer sb= new StringBuffer();
        JOptionPane.showMessageDialog(null, "Opened File");
        char[] ca= new char[5];
        while (br.ready()) {
            int len = br.read(ca);
            sb.append(ca,0,len);
        }
        br.close();
        System.out.println("Data is :"+" "+sb);
        String chuoi = sb.toString();
        txtPlainText.setText(chuoi);
        btnDecrypt.enable(true);

    } catch (IOException ex) {
        Logger.getLogger(b3_frmAES.class.getName()).log(Level.SEVERE,null,ex);
    }

}
```

Result:


PROGRAM AES ENCRYPT
User name:
nguyenminhthang
Password:
hello
Code:
fit
Login
Register
PlainText:
How about you
Encrypt
Write File
CipherText:
Decrypt
Open File


PROGRAM AES ENCRYPT
User name:
nguyenminhthang
Password:
hello
Code:
[B@5a2af8b8
Login
Register
PlainText:
How about you
Encrypt
Write File
CipherText:
Decrypt
Open File


PROGRAM AES ENCRYPT
User name:
nguyenminhthang
Password:
hello
Code:
[B@5a2af8b8
Login
Register
PlainText:
How about you
Encrypt
Write File
CipherText:
EHk2ZNibvHQCEga6rHjVZQ==
Decrypt
Open File


PROGRAM AES ENCRYPT
User name:
nguyenminhthang
Password:
hello
Code:
[B@d1ea03c
Login
Register
PlainText:
O5ftV0Pzj+QSufAKV9N47w==
Encrypt
Write File
CipherText:
How about you
Decrypt
Open File
tput - BAOMATLAB2 (run)
run:
Key is : nguyenminhthanghello
Create Key: javax.crypto.spec.SecretKeySpec@17961
Cipher Text generated using AES is O5ftV0Pzj+QSufAKV9N47w==
Decypted Text message is How about you
Data is : O5ftV0Pzj+QSufAKV9N47w==