

## 2.2 THUẬT TOÁN MÃ HÓA 3DES

Viết chương trình mã hóa và giải mã văn bản với thuật toán mã hóa 3DES.

Chương trình có thể thực hiện các chức năng sau:

Cho phép nhập văn bản vào hệ thống.

Cho phép nhập khóa bảo vệ văn bản.

Cho phép ghi File và mở File.

### 2.2.1 Hướng dẫn thuật toán TRIPLEDES:

TripleDES một biến thể an toàn hơn của DES còn được gọi là DESede hay 3DES. TripleDES có tính bảo mật cao hơn DES do sử dụng 3 vòng DES với các khóa khác nhau. Vòng đầu tiên và vòng thứ ba là vòng mã hóa, vòng thứ hai là vòng giải mã. DESede có thể dùng hai hoặc ba khóa có độ dài 56, 112 hoặc 168. nếu dùng hai khóa thì khóa đầu tiên được dùng cho vòng thứ nhất và vòng thứ ba, khóa thứ hai dùng cho vòng thứ hai.

Mã hóa với ba khóa 56 bit ( 168 bit).

**Bảng B1:** 3DES Mã hóa với ba khóa 56 bit

NGƯỜI GỬI	NGƯỜI NHẬN
<b>Bước 1:</b> mã hóa plaintext bằng khóa nhất	<b>Bước 1:</b> giải mã bản mã với khóa thứ ba
<b>Bước 2:</b> mã hóa văn bản được tạo ra ở bước 1 bằng khóa thứ hai	<b>Bước 2:</b> giải mã văn bản được tạo ra ở bước 1 bằng khóa thứ hai
<b>Bước 3:</b> mã hóa văn bản được tạo ra ở bước 2 bằng khóa thứ ba, tạo ra bản mã gửi cho người nhận.	<b>Bước 3:</b> giải mã văn bản được tạo ra ở bước 2 bằng khóa thứ nhất, tạo ra bản gốc do người gửi gửi.

Mã hóa với hai khóa 56 bit ( 112 bit)

**Bảng B2:** 3DES Mã hóa với hai khóa 56 bit

NGƯỜI GỬI	NGƯỜI NHẬN
<b>Bước 1:</b> mã hóa plaintext bằng khóa nhất	<b>Bước 1:</b> giải mã bản mã với khóa thứ nhất
<b>Bước 2:</b> giải mã văn bản được tạo ra ở bước 1 bằng khóa thứ hai	<b>Bước 2:</b> mã hóa văn bản được tạo ra bước 1 bằng khóa thứ hai
<b>Bước 3:</b> mã hóa văn bản được tạo ra ở bước 2 bằng khóa thứ nhất, tạo ra bản mã gửi cho người nhận.	<b>Bước 3:</b> giải mã văn bản được tạo ra bước 2 bằng khóa thứ nhất, tạo ra bản gốc do người gửi gửi.

Mã hóa với một khóa 56 bit

**Bảng B3: 3DES Mã hóa với một khóa 56 bit**

NGƯỜI GỬI	NGƯỜI NHẬN
<b>Bước 1:</b> mã hóa plaintext	<b>Bước 1:</b> giải mã bản mã nhận được từ người gửi.
<b>Bước 2:</b> giải mã văn bản được tạo ra ở bước 1	
<b>Bước 3:</b> mã hóa văn bản được tạo ra ở bước, tạo ra bản mã gửi cho người nhận.	

Mặc dù 3DES có tính bảo mật cao hơn DES, nhưng thực tế ít được sử dụng vì để tạo ra được bản mã phải chạy ba lần DES, nên tốc độ chậm, chiếm nhiều tài nguyên.

## 2.2.2 Hướng dẫn thực hành

**Bước 1:** Thiết Kế Form:

**DESEDE**

Input Key :

Plaintext :

CipherText :

## Bước 2: Viết hàm xử lý sự kiện

### B2.1: Khai báo các biến sau

```
private static final String UNICODE_FORMAT = "UTF8";
public static final String DESEDE_ENCRYPTION_SCHEME = "DESede";
private KeySpec myKeySpec;
private SecretKeyFactory mySecretKeyFactory;
private Cipher cipher;
byte[] keyAsBytes;
private String myEncryptionKey;
private String myEncryptionScheme;
SecretKey key;
```

### B2.2: Viết phương thức mã hóa encrypt

```

public String encrypt(String unencryptedString) {
    String encryptedString = null;
    try {
        cipher.init(Cipher.ENCRYPT_MODE, key);
        byte[] plainText = unencryptedString.getBytes(UNICODE_FORMAT);
        byte[] encryptedText = cipher.doFinal(plainText);
        BASE64Encoder base64encoder = new BASE64Encoder();
        encryptedString = base64encoder.encode(encryptedText);
    } catch (Exception e) {
        e.printStackTrace();
    }
    return encryptedString;
}

```

### B2.3: Viết phương thức giải mã decrypt

```

public String decrypt(String encryptedString) {
    String decryptedText=null;
    try {
        cipher.init(Cipher.DECRYPT_MODE, key);
        BASE64Decoder base64decoder = new BASE64Decoder();
        byte[] encryptedText = base64decoder.decodeBuffer(encryptedString);
        byte[] plainText = cipher.doFinal(encryptedText);
        String a= new String(plainText);
        System.out.println("chuoi plaintext : " + a);
        // decryptedText= bytes2String(plainText);
        decryptedText=a;
    } catch (Exception e) {
        e.printStackTrace();
    }
    return decryptedText;
}

```

### B2.4 Viết hàm xử lý sự kiện mã hóa

```

private void btnMaHoaActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        // quot:Sanjaal.comquot
        myEncryptionKey =txtkhoa.getText();
        myEncryptionScheme = DESEDE_ENCRYPTION_SCHEME;
        keyAsBytes = myEncryptionKey.getBytes(UNICODE_FORMAT);
        myKeySpec = new DESedeKeySpec(keyAsBytes);
        mySecretKeyFactory = SecretKeyFactory.getInstance(myEncryptionScheme);
        cipher = Cipher.getInstance(myEncryptionScheme);
        key = mySecretKeyFactory.generateSecret(myKeySpec);
        System.out.println(" khoa ma hoa k : " + " " + key);
        // sử dụng lớp DESEDE_EN
        String plainText=txtvanban.getText();
        // gọi phương thức mã hóa
        String encrypted=encrypt(plainText);
        System.out.println("Encrypted Value : " + encrypted);
        txtmahoa.setText(encrypted);

    } catch( Exception ex){}
}

```

### Bước 3: Kiểm Tra

DESEDE

Input Key : baomathongtinbuvanthieu

Mã Hóa    Hiện Thị    Ghi File

Plaintext :

CipherText :

QVaSWai+0cdC614kjd6qAagUs7ZKkgM4Mzf21Lm1i4fu

Giải Mã    All Show

DESEDE

Input Key : baomathongtinbuvanthieu

Mã Hóa    Hiện Thị    Ghi File

Plaintext :  
mon thuc hanh bao mat thong tin

CipherText :  
0cdC614kjd6qAagUs7ZKkgM4Mzf21Lm1i4fuInTmsvQ==

Giải Mã    All Show

DESEDE

Input Key : baomathongtinbuvanthieu

Mã Hóa    Hiện Thị    Ghi File

Plaintext :

CipherText :  
mon thuc hanh bao mat thong tin

Giải Mã    All Show

DESEDE

Input Key : baomathongtinbuvanthieu

Mã Hóa    Hiện Thị    Ghi File

Plaintext :  
mon thuc hanh bao mat thong tin

CipherText :  
0cdC614kjd6qAagUs7ZKkgM4Mzf21Lm1i4fuInTmsvQ==

Giải Mã    All Show