package com.chatapp;

import javax.crypto.Cipher;

import javax.crypto.KeyGenerator;

import javax.crypto.SecretKey;

import javax.crypto.spec.SecretKeySpec;

import java.util.Base64;

public class AESutil {

private static final String AES\_ALGORITHM = "AES";

private static SecretKey secretKey;

// Generate a secret key using KeyGenerator

public static void generateKey() throws Exception {

KeyGenerator keyGenerator = KeyGenerator.getInstance(AES\_ALGORITHM);

keyGenerator.init(128);

secretKey = keyGenerator.generateKey();

}

// Encrypt the message using AES

public static String encrypt(String message) throws Exception {

Cipher cipher = Cipher.getInstance(AES\_ALGORITHM);

cipher.init(Cipher.ENCRYPT\_MODE, secretKey);

byte[] encryptedBytes = cipher.doFinal(message.getBytes());

return Base64.getEncoder().encodeToString(encryptedBytes);

}

// Decrypt the message using AES

public static String decrypt(String encryptedMessage) throws Exception {

Cipher cipher = Cipher.getInstance(AES\_ALGORITHM);

cipher.init(Cipher.DECRYPT\_MODE, secretKey);

byte[] decodedBytes = Base64.getDecoder().decode(encryptedMessage);

byte[] decryptedBytes = cipher.doFinal(decodedBytes);

return new String(decryptedBytes);

}

// Get the Secret Key in Base64 for easy transfer

public static String getSecretKey() {

return Base64.getEncoder().encodeToString(secretKey.getEncoded());

}

// Set the Secret Key from a received Base64 string

public static void setSecretKey(String base64Key) {

byte[] decodedKey = Base64.getDecoder().decode(base64Key);

secretKey = new SecretKeySpec(decodedKey, AES\_ALGORITHM);

}

}