**Rijndael algorithm**

**Code:**

import javax.crypto.Cipher;

import javax.crypto.spec.SecretKeySpec;

import java.util.Base64;

public class RijndaelEncryption {

public static void main(String[] args) {

try {

String plaintext = "Hello, World!";

String key = "abcdefghijklmnop"; // 128-bit key (16 bytes)

// Encrypt plaintext

String encryptedText = encrypt(plaintext, key);

System.out.println("Encrypted: " + encryptedText);

// Decrypt ciphertext

String decryptedText = decrypt(encryptedText, key);

System.out.println("Decrypted: " + decryptedText);

} catch (Exception e) {

e.printStackTrace();

}

}

public static String encrypt(String plaintext, String key) throws Exception {

Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");

SecretKeySpec secretKey = new SecretKeySpec(key.getBytes(), "AES");

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

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

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

}

public static String decrypt(String encryptedText, String key) throws Exception {

Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");

SecretKeySpec secretKey = new SecretKeySpec(key.getBytes(), "AES");

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

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

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

return new String(decryptedBytes);

}

}

Output:

Encrypted: QSSZEzTSBD0hwGv9/H8cmg==

Decrypted: Hello, CSBS!

=== Code Execution Successful ===