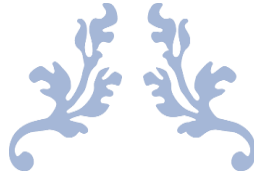




# TED UNIVERSITY

**CMPE325: Information Security and Cryptography**



---

## **RSA – Bob and Alice**

---

**Instructor of the Course: Haydar Çukurtepe**



***Prepared By İhsan Melih Şişman (56725508390)***

## **1. Introduction**

In this assignment, it focuses on integrating the RSA algorithm and the classic version of this problem, the encrypted messaging between Bob and Alice.

## **2. Detail Section**

Assume Alice requests that her friends encrypt their email communications before sending them to her. An email message is merely a very large number since computers encode text as long numbers (01 for "A," 02 for "B," and so on). Electronic communications are frequently encrypted and subsequently decrypted using the RSA Encryption Scheme.

## **3. ASSUMPTIONS**

### **3.1 For Alice**

When you do this conversion, you find the public key. Firstly, you should choose two prime numbers. Then you should calculate the product  $n = P.Q$ . After that, you should calculate  $m = (p-1).(q-1)$ . Then, you choose numbers  $E$  and  $d$  so that  $E.d$  has a remainder of 1 when divided by  $M$ . Finally, you should publish her public key  $(n,e)$ . So, we can reach the public key.

### **3.2 For Bob**

Firstly, you should find Alice's public key  $(n,e)$ . then you should find the remainder  $C$  when  $M^e$  is divided by  $n$ . Finally, you should Send ciphertext  $C$  to Alice. In this way, we can reach the result.

#### **3.2.1 For Alice**

After these operations, you should use her private key  $(n,d)$ . After this, you should find remainder  $R$  when  $C^d$  is divided by  $n$ . Finally,  $R$  matches the message  $M$  that Bob wanted to send to Alice! Eventually, we can complete the encryption process.

As a result, after these stages are done, the keys are revealed, and we do the encryption process accordingly.

## 4. How to Run this Code

The screenshot shows a Java Swing window titled "ECB RSA" with a standard Windows-style title bar (minimize, maximize, close buttons). The window is divided into two main sections: "Sender" and "Receiver".

**Sender Section:**

- It has three input fields labeled "P:", "Q:", and "E:". The values entered are 3, 11, and 7 respectively.
- Below these fields, there are two lines of text: "Public key: After generation, you can see it" and "Private Key: After generation, you can see it."
- A blue button labeled "Generate For Keys" is positioned below the text.
- A text label "Message To Be Encrypted:" is followed by an empty input field.
- A green button labeled "Encrypt" is located below the input field.
- Below the "Encrypt" button, there is a text label "Cipher Text : After clicking encrypt button you can see it."
- A line of text reads: "Please copy the cipher text area to message to be decrypted area and after decrypt it."

**Receiver Section:**

- It has three input fields labeled "P:", "Q:", and "E:". The values entered are 5, 11, and 9 respectively.
- Below these fields, there are two lines of text: "Public key: After generation, you can see it" and "Private Key: After generation, you can see it."
- A blue button labeled "Generate For Keys" is positioned below the text.
- A text label "Message To Be Decrypted:" is followed by an empty input field.
- A red button labeled "Decrypt" is located below the input field.
- Below the "Decrypt" button, there is a text label "Plain Text : After clicking decrypt button, you can see it."

First, we enter the necessary data and press the 'Generate for Keys' button. We can do this for both parties at the same time. At this stage, our public and private keys appear.

**ECB RSA**

**Sender**

P:

Q:

E:

Public key: (3,33)

Private Key: (7,33)

Generate For Keys

Message To Be Encrypted:

Encrypt

Cipher Text : After clicking encrypt button you can see it.

Please copy the cipher text area to message to be decrypted area and after decrypt it.

**Receiver**

P:

Q:

E:

Public key: (23,55)

Private Key: (7,55)

Generate For Keys

Message To Be Decrypted:

Decrypt

Plain Text : After clicking decrypt button, you can see it.

As seen in the example, when the 'Generate for Keys' button was pressed, the keys were revealed.

### ECB RSA

---

#### Sender

P:

Q:

E:

Public key: (3,33)

Private Key: (7,33)

Message To Be Encrypted:

Cipher Text : 23 49 11 2 28

Please copy the cipher text area to message to be decrypted area and after decrypt it.

---

#### Receiver

P:

Q:

E:

Public key: (23,55)

Private Key: (7,55)

Message To Be Decrypted:

Plain Text : After clicking decrypt button, you can see it.

After this stage, we enter the message to be encrypted and press the 'Encrypt' button and the necessary numbers appear.

These numbers, which are seen in the cipher text, will be used for the decryption process below.

**ECB RSA**

**Sender**

P:

Q:

E:

Public key: (3,33)

Private Key: (7,33)

Message To Be Encrypted:

Cipher Text : 23 49 11 2 28

Please copy the cipher text area to message to be decrypted area and after decrypyt it.

**Receiver**

P:

Q:

E:

Public key: (23,55)

Private Key: (7,55)

Message To Be Decrypted:

Plain Text : melih

Then we enter the numbers we obtained in the cipher text into the 'Message to Be Decrypted' section and get the normal form of the message. The process is completed at this stage.

At this stage, the program achieved its goal.