



Practice activity – RSA Encryption/Decryption

In this practice activity, you will apply your knowledge of public key cryptography to securely transmit a sensitive piece of information from your company to a business partner. You will first generate a private/public key. You will then encrypt some text using the public key of your business partner and subsequently decrypt it (acting as the business partner) using the private key.

RSA (Rivest–Shamir–Adleman) is one of the most popular asymmetric (public key) cryptographic algorithms. RSA supports several different key sizes including 1024, 2048 and 4096 bits.

Resources:

- **Text to Encrypt:** *“This is a confidential sentence that needs to be encrypted”*
- **Public Key Cryptography Algorithm to be used:** RSA
- **Key length:** 1024 bits
- **RSA Key Generator and Encrypt/Decrypt:**
Online: <https://www.javainuse.com/rsagenerator>

Step 1

Please generate a 1024-bit private/public key pair based on RSA and save both the private and public keys. Please note that this step would normally be performed by your business partner, and they will only share their public key with your company.

Step 2

Please encrypt the text *“This is a confidential sentence that needs to be encrypted”* using the public key of your business partner and save the generated ciphertext.

Step 3

Acting as the business partner, please decrypt the ciphertext that you generated in Step 2 using the private key and verify if the message contents match the original text.

Solution on next page.



Solution

Step 1

Acting as the business partner, we first generate a 1024-bit private/public key pair.

Online: <https://www.javainuse.com/rsa-generator>

RSA Generate Keys

This tool generates RSA public key as well as the private key of sizes - 512 bit, 1024 bit, 2048 bit, 3072 bit and 4096 bit with Base64 encoded. The generated private key is generated in PKCS#8 format and the generated public key is generated in X.509 format.

Public Key

```
MIGfMA0GCsGSIb3DQEBAQUAA4GNADCBiQKBgQCTngiZ9WBYbyu40+z4sI4607dYxViR4QeUEE0tFuKlnBXtablw470Ei+ZxxPbwwzw2wd2N0I3I6jAnykdxr6ytAb3NUVve+sodGk2t9SY8M+1Bmtn41mJU4tjvANjMT/qAah7ixo+f6MKnchXXE
```

Private Key

```
MIICdQIBADANBgkqhkiG9w0BAQEFAASCAl8wgGJbAgEAAoGBAJ0eCJn1YFhvK7g77PijwjjrTt1jFWJHhB5QR7S1+4qWcFe1puVbjvQSL5nHE9vDDPDbB3Y04jcyjMCfKR3GvrK0Bvc1RW976yh0aTa31Jjwz7UGa2fjWYLTi208A2MxP+ABqHuLGi
```

Key Size: 1024

Generate Keys

Step 2

We provide the plaintext and use the public key to encrypt it.

Online: <https://www.javainuse.com/rsa-generator>

RSA Encryption

Enter Plain Text to Encrypt -

Enter Public/Private key

Cipher Type: RSA

Key Type: Public Key

Encrypt

The String which is to be encrypted using RSA

The public/private key we have created above. As RSA is asymmetric encryption technique, if text is encrypted using public key then for decryption we should use the private key and vice versa.

Select the Encryption Algorithm. Some Algorithms need to have key size greater than 512 bits.

Specify if the entered key is a public key or private key. As RSA is asymmetric encryption technique, if text is encrypted using public key then for decryption we should use the private key and vice versa.

SAyEzgo0JUBAHDTTXkCwGPznPCfWY0b8m040TjnJVLMI0H3Ye615r1vKS1Wa4la+7//ZZ+hwXUMun00YExPaS3zWa0Hzj0Iz2z2bbwPsBSG6p8shq8xnrMIUwongqkcYJL0x43obFZ10V/0z9hLTFoWZE//FpuvFjhL3bv+sMmhQ=

Please note this in this case, we are using the **public key** of our business partner to encrypt so that no one else can decrypt it except for our business partner who has the **private key**. As discussed in the lecture, the public key **cannot** be used again to decrypt the message which was encrypted with the public key.



Step 3

Assuming we are the business partner, we provide the ciphertext and use the private key to decrypt it.

Online: <https://www.javainuse.com/rsagenerator>

The screenshot shows the 'RSA Decryption' web application. It has a pink header. On the left, there are three input fields: 'Enter Encrypted Text to Decrypt (Base64)' containing a long Base64 string, 'Enter Public/Private key' containing a long key string, and 'Cipher Type' set to 'RSA'. Below these is a 'Key Type' dropdown set to 'Private Key'. A green 'Decrypt' button is centered. Below the button is a text area containing the decrypted message: 'This is a confidential sentence that needs to be encrypted'. On the right side, there are three explanatory paragraphs: 'The RSA Encrypted String which we want to decrypt', 'The public/private key we have created above. As RSA is asymmetric encryption technique, if text is encrypted using public key then for decryption we should use the private key and vice versa.', and 'Specify if the entered key is a public key or private key. As RSA is asymmetric encryption technique, if text is encrypted using public key then for decryption we should use the private key and vice versa.'

As a practice activity, you can try decrypting the ciphertext using the public key, but you will see that it will not give us the correct output and will display error, thus ensuring the confidentiality of the message and providing assurance that only the destination with the private key can decrypt the message.

Congratulations! You have successfully generated RSA key pairs as well as seen public key cryptography in action.