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.

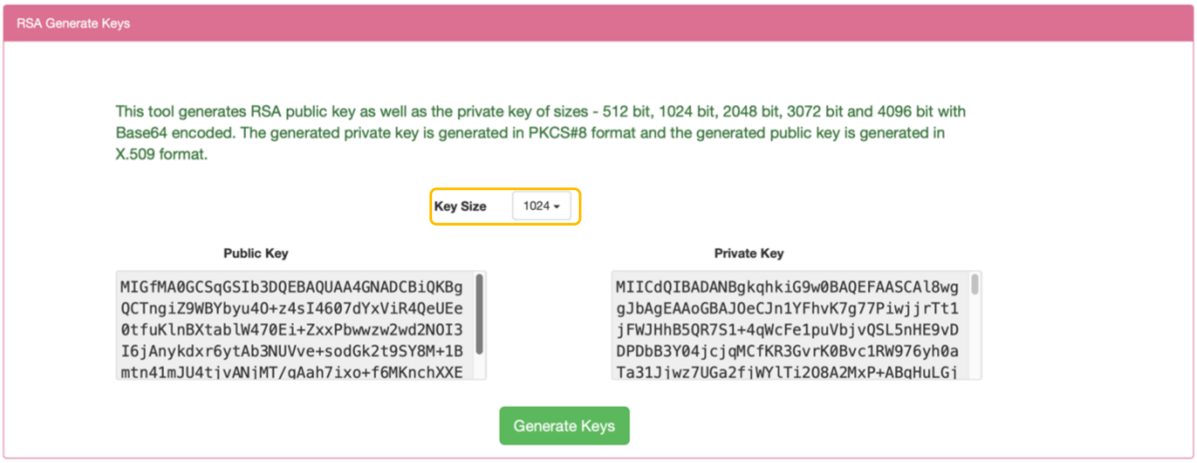
Solution on next page.

Solution

Step 1

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

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



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.

Key Size: 1024

Public Key

```
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBg
QCTngiZ9WBYbyu4O+z4sI4607dYxViR4QeUEe
0tfuKlnBXtablW470Ei+ZxxPbwwzw2wd2NOI3
I6jAnykdxr6ytAb3NUVve+sodGk2t9SY8M+1B
mtn41mJU4tjvANjMT/gAah7ixo+f6MKnchXXE
```

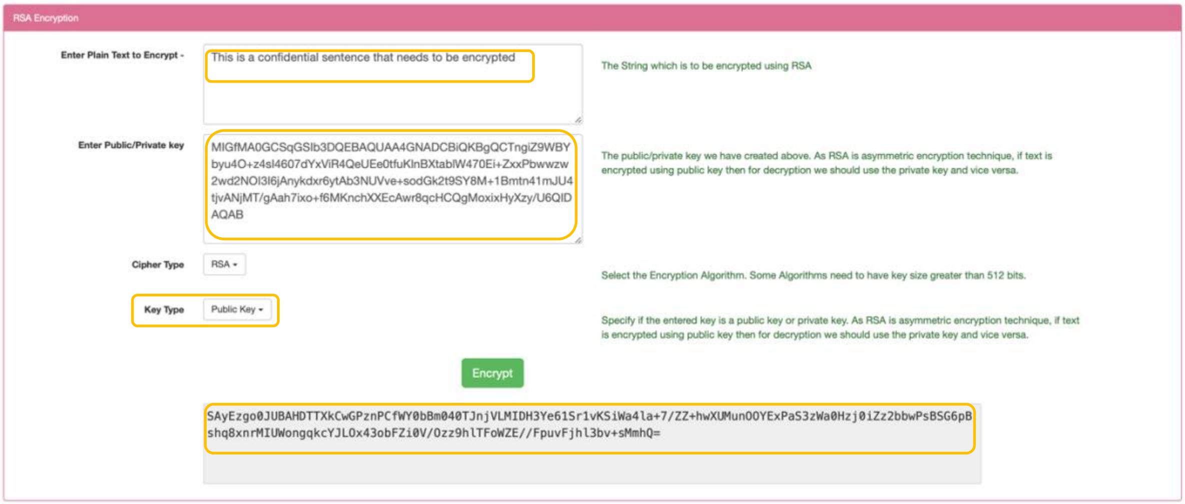
Private Key

```
MIICdQIBADANBgkqhkiG9w0BAQEFAASCAl8wg
gJbAgEAAoGBAJOeCJn1YFhvK7g77PiwjjrTt1
jFWJHhB5QR7S1+4qWcFe1puVbjvQSL5nHE9vD
DPDbB3Y04jcjqMCfKR3GvrK0Bvc1RW976yh0a
Ta31Jjwz7UGa2fjWYlTi2O8A2MxP+ABqHuLGj
```

Step 2

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

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



RSA Encryption

Enter Plain Text to Encrypt - This is a confidential sentence that needs to be encrypted

The String which is to be encrypted using RSA

Enter Public/Private key

```
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCTngiZ9WBY
byu4O+z4sI4607dYxViR4QeUEe0tfuKlnBXtablW470Ei+ZxxPbwwzw
2wd2NOI3I6jAnykdxr6ytAb3NUVve+sodGk2t9SY8M+1Bmtn41mJU4
tjvANjMT/gAah7ixo+f6MKnchXXEcAwr8qcHCQgMoxixHyXzy/U6QID
AQAB
```

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.

Cipher Type: RSA

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

Key Type: Public Key

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.

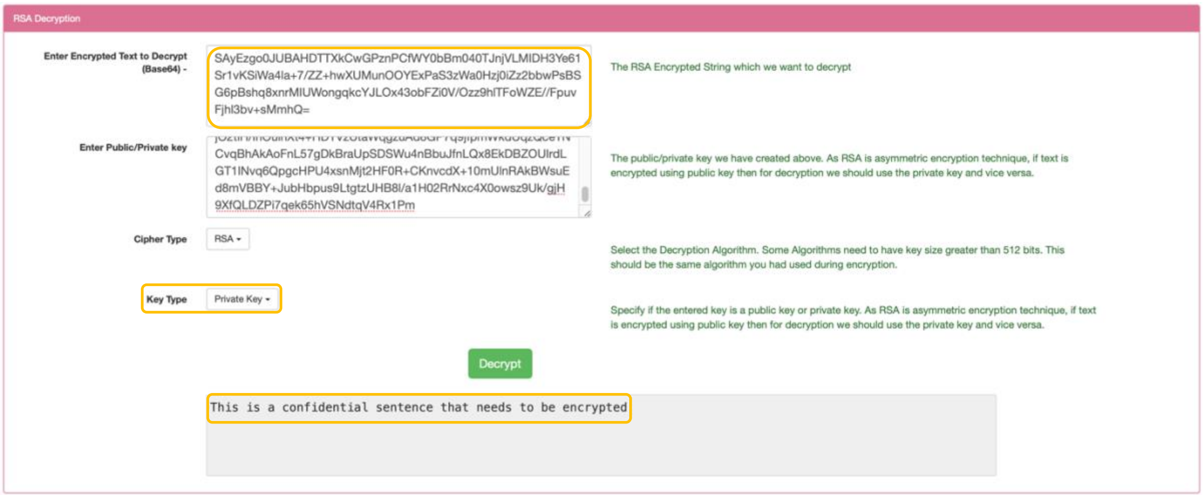
```
SAyEzgo0JUBAHDTTXkCwGPznPCfWY0bBm040TJnjVLMIDH3Ye61Sr1vKSiWa4la+7/ZZ+hwXUMunOOYExPaS3zWa0Hzj0iZz2bbwPsBSG6pB
shq8xnrMIUWongqkcYJLOx43obFZi0V/Ozz9hlTFoWZE//FpuvFjhl3bv+sMmhQ=
```

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*



RSA Decryption

Enter Encrypted Text to Decrypt (Base64) -

SAyEzgo0JUBAHDTTXkCwGPznPCfWY0bBm040TJnjVLMIDH3Ye61
Sr1vKSiWa4la+7/ZZ+hwXUMunOOYExPaS3zWa0Hzj0iZz2bbwPsBS
G6pBshq8xnrMIUWongqkcYJLOx43obFZi0V/Ozz9hlTFoWZE//Fpuv
Fjhl3bv+sMmhQ=

The RSA Encrypted String which we want to decrypt

Enter Public/Private key

CvqBhAkAoFnL57gDkBraUpSDSWu4nBbuJfnLQx8EkDBZOUlrdL
GT1INvq6QpgcHPU4xsnMjt2HF0R+CKnvcdX+10mUlnRAkBWsuE
d8mVBBY+JubHbpus9LtgtzUHB8l/a1H02RrNxc4X0owsz9Uk/gjH
9XfQLDZPi7qek65hVSNdtqV4Rx1Pm

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.

Cipher Type: RSA

Select the Decryption Algorithm. Some Algorithms need to have key size greater than 512 bits. This should be the same algorithm you had used during encryption.

Key Type: Private Key

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.

```
This is a confidential sentence that needs to be encrypted
```

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.