Practical Lab 8

Student ID: ____________________   Student Name: _________________________

Q1. Test the following codes to **encrypt/decrypt** a message using **javascript RSA functions**.

Test demo1 here: [Demo 1]

1. Download **rsa.js** and put it under your student folder. Set permission as 755.
2. Enter a message, see the first highlight part
3. Call **RSA encryption and decryption functions**, see the second and third highlight part
4. Link **rsa.js** in **test_des.html**, see the fourth highlight part

Please note that RSA key pair are provided in the fifth highlight part and the sixth highlight part.

test_rsa.html code as below:

```
<html>
  <head>
    <title>JavaScript RSA Encryption</title>
  </head>
  <body>
            <h1>Lab 8 Demo 1: JavaScript RSA test</h1>

            Enter a plaintext: <input id="plaintext" name="plaintext" type="text">
            <br/><br/>
            RSA encryption key is defined in the Code!
            <br/><br/>
            <button type="button" onclick="RSA_encryption()">Encrypt the Message</button>

            <h3>Encrypted value</h3>
            <p id="encrypted"></p>
            <br/><br/>
            RSA decryption key is defined in the Code!
            <br/><br/>
            <button type="button" onclick="RSA_decryption()">Decrypt the "Encrypted value"</button>
            <h3>Decrypted value</h3>
            <p id="decrypted"></p>

    <script src="rsa.js"></script>
    <script type="text/javascript">
                    function RSA_encryption(){
                            var plaintext = document.getElementById("plaintext").value;
                            var pubilc_key = "-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzdxaei6bt/xIAhYsdFdW62CGTpRX+GXoZkzqvbf5oOxw4wKENjFX7LsqZ
XxdFfoRxEwH90zZHLHgsNFzXe3JqiRablDcNZmKS2F0A7+Mwrx6K2fZ5b7E2fSLFbC7FsvL22mN0KNAp35tdADpl4lKqNFuF7NT22Z
Bp/X3ncod8cDvMb9tl0hiQ1hJv0H8My/31w+F+Cdat/9Ja5d1ztOOYIx1mZ2FD2m2M33/BgGY/BusUKqSk9W91Eh99+tHS5oTvE8CI8g7
pvhQteqmVgBbJOa73eQhZfOQJ0aWQ5m2i0NUPcmwvGDzURXTKW+72UKDz671bE7YAch2H+U7UQeawwIDAQAB-----END PUBLIC
KEY-----";
                            // Encrypt with the public key...
                            var encrypt = new JSEncrypt();
                            encrypt.setPublicKey(pubilc_key);
                            var encrypted = encrypt.encrypt(plaintext);

                            document.getElementById("encrypted").innerHTML = encrypted;
                    }

            function RSA_decryption(){

                    var encrypted = document.getElementById("encrypted").innerHTML;

                    var private_key = "-----BEGIN PRIVATE KEY-----
MIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQDN3Fp6Lpu3/EgCFix0V1brYIZOIFf4ZehmTOq9t/mg7HDjAoQ2M
VfsuyplfF0V+hHETAf3TNkcseCw0XNd7cmqJFpsgNw1mYpLYXQDv4zCvHorZ9nlvsTZ9lsVsLsWy8vbaY3Qo0Cnfm10AOmXiUqo0W
4Xs1PbZkGn9fedyh3xwO8xv22XSGJDWEm/QfwzL/fXD4X4J1q3/0Irl3XO045gjHWZnYUPabYzff8GAZj8G6xQqpKT1b3USH3360dLmh
O8TwIjyDum+FC16qZWAFsk5rvd5CFI85AnRpZDmbaLQ1Q9ybC8YPNRFdMpb7vZQoPPrvVsTtgByHYf5TtRB5rDAgMBAAECggEAU
DPieCnCd1rtvwpehXElpwxzJxg6ccdaVMjwx7tuoRidHoRzeB2fUNbWvLVIGvDTjTPGAr5l9BoFHT5tARJMeGlzbISDxsosDBRKu88cC
x6dRI3ukcjSLsxMh8XUDhyWLsSgAMIpxVfHUuOsHmLZ2I3Ho6o1KIxdVg/JSgtdwTqjz3w8jmGQ/NXgc7Ym/ys1fLG9L2nYdMzK/mRJf
/BnXiCNE6/SYlZYO716oC688UJBWS3BqB9jaJyNpigX//ynJvU6xw8FhHt4fRStUmCCYAYhCQu3XgbtmxKisDGhdBVASG+DM+vVTh+
sSvxkNrjJjF+m2tSg578A8C8Ls0r3uQKBgQDpO9e178NR0HHmvWbZR9+uPugf4UT9+U2/dEfJBHAOp2GRslvXkFwbPHuSHkc0iEPw
z+U8gPC8jInSslKOUDtaGtUaVzzWrxxh7DggWx4pYs3I0Ki8C+CRTTdOY9GAFa9jhlyRmf6v9QoAH/loGNV2qYFbb+HweD0PnxIWha1t
xQKBgQDh9lBBItW7T96foUmHOn+x6xIF5MNDHxLBY6bngxKvMTZoi5C6wmmCmasF45LWbkvUiMAsovYN5z4cJnKXWmRmCS8NX
UucmUgdvsmCbiB62BmZvHaOffmnldhcAjBebT/Bn5qMvKCNy3fQFSfuEw1eRRO2lofB4o7z7m794vo25wKBgEPowrQcrZhCwwdWG
n4laUGl23l80+PHFRYru0MSYbZCkiwjZXRMeiUMBUbUPhNTocSal7rsKCweF3sbpOH/BmkD6wySXgp8Th1M9EKnhS6zsAtKhfbK1oY
4H2RZuAQ9TCYD0BIM7pU5GcJTjQD8ShsU269N8lFcERtdTbldjtOpAoGAF4YkADAa6lhjXg0loY2Gk9hdFji913QZuMaOLtYnkNO3zW
SSWc85ut4Svxc1R1vOSz89eqgwo7vqbHXYQken4jOckXCgGZqftnERe6HJgeCTsby8PxOAdVUBuHqF3J7VH2xlY7eTo4+GVsSNFq0
nHCRm6/RmW9ohdeXh6k7CLAsCgYBZe3RLWuffKxg+lZmv9tJDOO813QPLFeixrBYhKjGDcwjVYcCugGNDmyStM0/++uWddgMKav
NALjpamu8KolDNivrjL1qaFHX9Bpi108T+dDn2WpX+vUP6hjA/U2wtTvUbJle1SsbZxRrV9gf5PAJqTrQY4u28ezjR3PCV+R4kdw==-----
END PRIVATE KEY-----";
                                // Decrypt with the private key...
        var decrypt = new JSEncrypt();
        decrypt.setPrivateKey(private_key);
        var decrypted = decrypt.decrypt(encrypted);

                                document.getElementById("decrypted").innerHTML = decrypted;
                }
                </script>
  </body>
</html>
```

Q2. Test the following codes to **encrypt/decrypt** a message using **php RSA functions**.

test Demo2 here: Demo 2

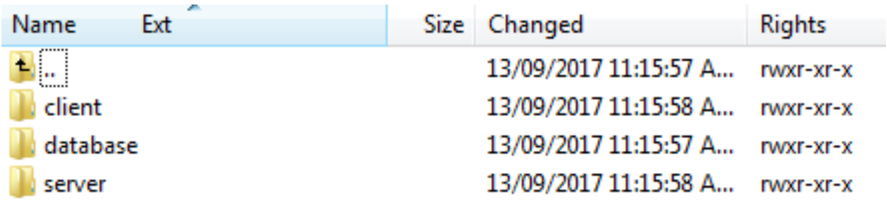
1. Download **rsa.php** file and put it under your student folder. Set permission as 755.
2. Download **public.key** and **private.key** to your folder. Set permission as 777.
3. Link **rsa.php** in the **test_rsa.php,** see first highlight part
4. Enter a plaintext message for encryption, the second highlight part
5. Retrieve RSA public (encryption) key from the public.key file, see the third highlight part
6. Call php RSA encryption API to encrypt the plaintext using the RSA public key, see the fourth highlight part
7. Retrieve RSA private (decryption) key from the private.key file, see the fifth highlight part
8. Call php RSA decryption API to decrypt the ciphertext using the RSA private key, see the sixth highlight part

test_des.php code as below:

```
<?php
include('rsa.php');
?>
<html>
<body>

<h1>Lab 8 Demo 2: PHP RSA test</h1>
<?php

// set the plaintext
$plaintext = 'hello world';
echo 'Plain text: ' . $plaintext."<br/><br/><br/>";

// Get the public Key
$publicKey = get_rsa_publickey('public.key');

// compute the ciphertext
$encrypted = rsa_encryption($plaintext, $publicKey);
echo $encrypted."<br/><br/><br/>";

// Get the private Key
$privateKey = get_rsa_privatekey('private.key');

// compute the decrypted value
$decrypted = rsa_decryption($encrypted, $privateKey);

echo 'Unencrypted Data: ' . $decrypted;
?>
</body>
</html>
```

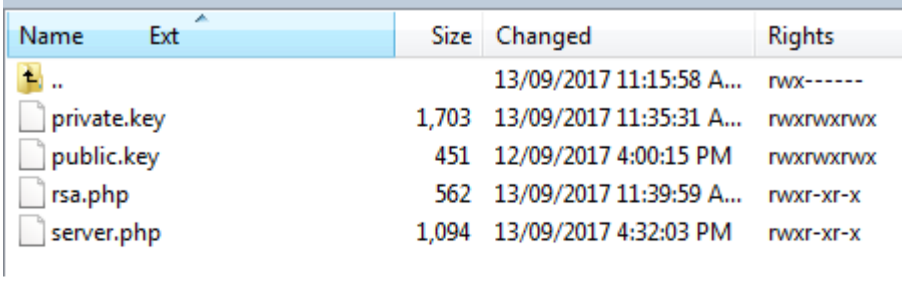
Q3. Based on **Q1, Q2**, write a client.html and server.php to achieve the following:

Create the folders/files as below:



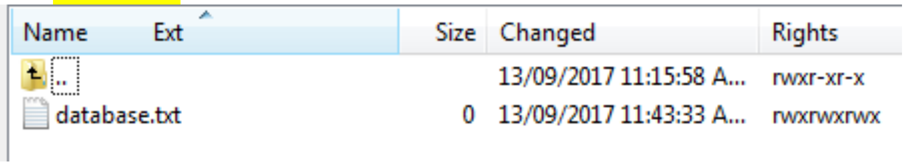
Name	Ext	Size	Changed	Rights
..			13/09/2017 11:15:57 A...	rwxr-xr-x
client			13/09/2017 11:15:58 A...	rwxr-xr-x
database			13/09/2017 11:15:57 A...	rwxr-xr-x
server			13/09/2017 11:15:58 A...	rwxr-xr-x

in the folder server:



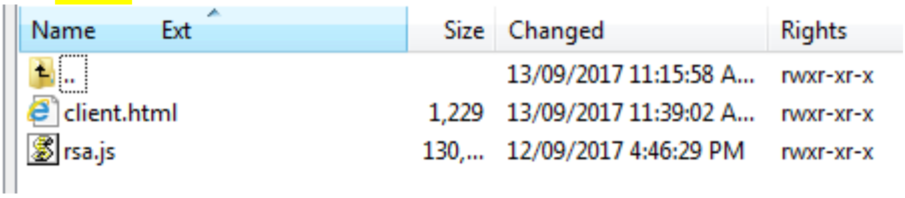
Name	Ext	Size	Changed	Rights
..			13/09/2017 11:15:58 A...	rwx------
private.key		1,703	13/09/2017 11:35:31 A...	rwxrwxrwx
public.key		451	12/09/2017 4:00:15 PM	rwxrwxrwx
rsa.php		562	13/09/2017 11:39:59 A...	rwxr-xr-x
server.php		1,094	13/09/2017 4:32:03 PM	rwxr-xr-x

in the folder database:



Name	Ext	Size	Changed	Rights
..			13/09/2017 11:15:58 A...	rwxr-xr-x
database.txt		0	13/09/2017 11:43:33 A...	rwxrwxrwx

in the folder client:



Name	Ext	Size	Changed	Rights
..			13/09/2017 11:15:58 A...	rwxr-xr-x
client.html		1,229	13/09/2017 11:39:02 A...	rwxr-xr-x
rsa.js		130,...	12/09/2017 4:46:29 PM	rwxr-xr-x

Please download **rsa.js**, **rsa.php**, **public.key** and **private.key** from blackboard, and do not change anything from them.

IMPORTANT: You can directly use the RSA public (encryption) key and private (decryption) key in javascript (refer to the fifth highlight part and the sixth highlight part in **Q1**) and in PHP (refer to the third highlight part the the fifth highlight part in **Q2**).

Expected outcome: users' input will be encrypted (RSA encryption using public key) before submitting to server, and will be decrypted (RSA decryption using private key) on the server side, and store plaintext in the database.

Client-side:

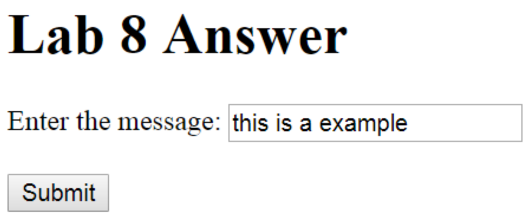
1. Enter a message
2. Encrypt the message using javascript RSA encryption API
3. Submit the ciphertext

Server-side:

1. Retrieve the ciphertext from client-side
2. Retrieve the RSA private (decryption) key
3. Decrypt the ciphertext using php RSA decryption API
4. Save the decrypted value to database.

For example:

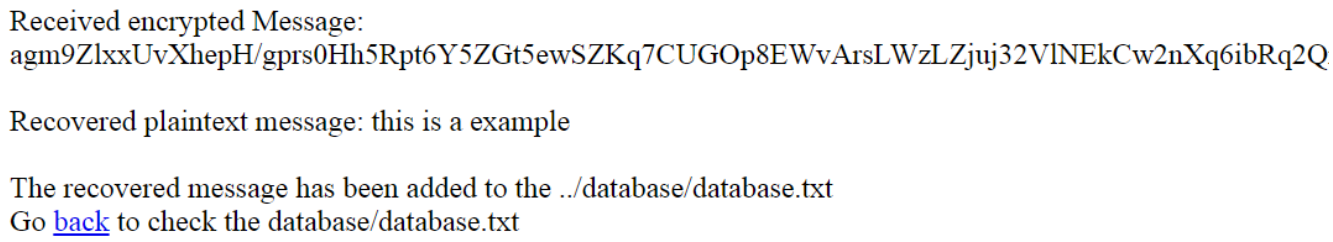
client-side



Lab 8 Answer

Enter the message: this is a example

server-side

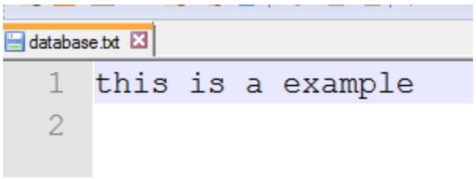


Received encrypted Message:
agm9ZlxxUvXhepH/gprs0Hh5Rpt6Y5ZGt5ewSZKq7CUGOp8EWvArsLWzLZjuj32VlNEkCw2nXq6ibRq2Q

Recovered plaintext message: this is a example

The recovered message has been added to the ../database/database.txt
Go back to check the database/database.txt

Database



```
this is a example

```

Answer of Q3:

client.html

```
------------------------------------------------------------------------------------------------------------------
<html>

<body>

<h1>Lab 8 Answer</h1>

<FORM ACTION="../server/server.php" method="POST">
        Enter the message: <input type="text" id="message" name="message" />
        <br/><br/>

        <button type="submit" onclick="RSA_encryption()">Submit</button>
</table>

<br/><br/>


</FORM>

<script type="text/javascript" src="rsa.js"></script>
<script type="text/javascript">

                function RSA_encryption(){

                        var message = document.getElementById("message").value;

                        var pubilc_key = "-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzdxaei6bt/xIAhYsdFdW62CGTpRX+
GXoZkzqvbf5oOxw4wKENjFX7LsqZXxdFfoRxEwH90zZHLHgsNFzXe3JqiRabIDcNZmKS2F0A7+
Mwrx6K2fZ5b7E2fSLFbC7FsvL22mN0KNAp35tdADpI4IKqNFuF7NT22ZBp/X3ncod8cDvMb9tl0h
iQ1hJv0H8My/31w+F+Cdat/9Ja5d1ztOOYIx1mZ2FD2m2M33/BgGY/BusUKqSk9W91Eh99+tHS5
oTvE8CI8g7pvhQteqmVgBbJOa73eQhZfOQJ0aWQ5m2i0NUPcmwvGDzURXTKW+72UKDz671b
E7YAch2H+U7UQeawwIDAQAB-----END PUBLIC KEY-----";

                        // Encrypt with the public key...
                        var encrypt = new JSEncrypt();
                        encrypt.setPublicKey(pubilc_key);
                        var encrypted = encrypt.encrypt(message);

                        document.getElementById("message").innerHTML = encrypted;
                        document.getElementById("message").value = encrypted;
                }
</script>


</body>
</html>

------------------------------------------------------------------------------------------------------------------
```

server.php

```
------------------------------------------------------------------------------------------------------------------------

<?php

include('rsa.php');

?>

<html>
<body>

<?php

        //Receive user input from clint side
        $message = $_POST['message'];


        //retrieve private (decryption) key
        $privateKey = get_rsa_privatekey('private.key');

        //decrypt the received message using the private (decryption)key and PHP RSA
decrytion API
        $recovered_message = rsa_decryption($message, $privateKey);

        echo "Received encrypted Message: " . $message . "<br/><br/>";

        echo "Recovered plaintext message: " . $recovered_message . "<br/><br/>";

        //access to database/database.txt
        $file = fopen("../database/database.txt","a");
        //insert this user into the users.txt file
        fwrite($file,$recovered_message."\n");
        //close the "$file"
        fclose($file);

        echo "The recovered message has been added to the ../database/database.txt";
        echo "<br/>Go <a href='http://titan.csit.rmit.edu.au/~e23700/Lab8/answer/'>back</a> to
check the database/database.txt";

?>

</body>
</html>
------------------------------------------------------------------------------------------------------------------------
```