Practical Lab 9

Student ID: ____________________ **Student Name: _________________________**

Q1. Test the following javascrpt code:

test Demo2 here: [Demo 1]()

1. Download **rsa.js** and **sha256.js** and put them under your student folder. Set permission as 755.
2. Enter the message, see Step 1: the first highlight part
3. Get the current timestamp, see Step 2: the second highlight part
4. Encrypt the hashed value of message and the timestamp together, see Step 3: the third highlight part
5. Decrypt the encrypted value, see Step 4: the fourth highlight part
6. Split the decrypted value, see Step 5: the fifth highlight part

click_this.html code as below:

```
------------------------------------------------------------------------------------------------------------
<html>
<body>
        <h1>Lab 9 Demo 1</h1>

        <h3>Step 1: message</h3>
        <input name="message" id="message" type="text" >
        <br/><br/>
        <button type="button" onclick="hash()">Hash</button>
        <br/><br/>
        <p name="hash_value" id="hash_value"></p>

        <h3>Step 2: timestamp</h3>
        <h3>Current Time: The Current Unix Timestamp since Jan 01 1970. (UTC)</h3>
        <button type="button" onclick="getTime()">Get current time in second</button>
        <br/><br/>
        <p name="currentTime" id="currentTime"></p>

        <h3>Step 3: RSA encryption</h3>
        <h3>Encrypted (the message + "&" the timestamp) together</h3>
        <button type="button" onclick="encryption()">encrypt</button>
        <br/><br/>
        <p name="encrypted" id="encrypted"></p>


        <h3>Step 4: RSA decryption</h3>
        <h3>Decrypt the ciphertext</h3>
        <button type="button" onclick="decryption()">decrypt</button>
        <br/><br/>
        <p name="decrypted" id="decrypted"></p>

        <h3>Step 5: Split decrypted value based on "&"</h3>
        <h3>Decrypt the ciphertext</h3>
        <button type="button" onclick="split_string()">split</button>
        <br/><br/>
        <p name="split1" id="split1"></p>
        <p name="split2" id="split2"></p>




        <script src="sha256.js"></script>
        <script src="rsa.js"></script>
        <script type="text/javascript">
                function hash() {
                    var input = document.getElementById('message').value;

                    var hash = SHA256.hash(input);

                    document.getElementById('hash_value').innerHTML = hash;
                    document.getElementById('hash_value').value = hash;
          }

          function getTime() {

                    //get current time in Second since Jan 01 1970. (UTC)
                    var currentTime = Math.floor(new Date().getTime() / 1000);

                    document.getElementById('currentTime').innerHTML = currentTime;
                    document.getElementById('currentTime').value = currentTime;
          }

          function encryption(){

                    var hashed_value = document.getElementById('hash_value').value;
                    var timestamp = document.getElementById('currentTime').value;

                    var ciphertext = RSA_encryption(hashed_value+"&"+timestamp, pubilc_key);

                    document.getElementById('encrypted').innerHTML = ciphertext;
                    document.getElementById('encrypted').value = ciphertext;
          }

          function decryption(){

                    var encrypted = document.getElementById('encrypted').value;

                    var decrypted = RSA_decryption(encrypted, private_key);

                    document.getElementById('decrypted').innerHTML = decrypted;
                    document.getElementById('decrypted').value = decrypted;
          }


          function split_string(){

                    var string = document.getElementById('decrypted').value;

                    var split = string.split("&");

                    document.getElementById('split1').innerHTML = split[0];
                    document.getElementById('split1').value = split[0];

                    document.getElementById('split2').innerHTML = split[1];
                    document.getElementById('split2').value = split[1];
          }


     </script>

     <script type="text/javascript">

          var pubilc_key = "-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzdxaei6bt/xIAhYsdFdW62CGTpRX+GXoZkzqvbf5oOxw4wKENjFX7LsqZXxdFfoRxEwH90zZHLH
gsNFzXe3JqiRabIDcNZmKS2F0A7+Mwrx6K2fZ5b7E2fSLFbC7FsvL22mN0KNAp35tdADpl4lKqNFuF7NT22ZBp/X3ncod8cDvMb9tl0hiQ1hJv0H8My/31w+F
+Cdat/9Ja5d1ztOOYIx1mZ2FD2m2M33/BgGY/BusUKqSk9W91Eh99+tHS5oTvE8CI8g7pvhQteqmVgBbJOa73eQhZfOQJ0aWQ5m2i0NUPcmwvGDzURXT
KW+72UKDz671bE7YAch2H+U7UQeawwIDAQAB-----END PUBLIC KEY-----";
          var private_key = "-----BEGIN PRIVATE KEY-----
MIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQDN3Fp6Lpu3/EgCFix0V1brYIZOlFf4ZehmTOq9t/mg7HDjAoQ2MVfsuyplfF0V+hHETAf3
TNkcseCw0XNd7cmqJFpsgNw1mYpLYXQDv4zCvHorZ9nlvsTZ9IsVsLsWy8vbaY3Qo0Cnfm10AOmXiUqo0W4Xs1PbZkGn9fedyh3xwO8xv22XSGJDWEm/
QfwzL/fXD4X4J1q3/0lrl3XO045gjHWZnYUPabYzff8GAZj8G6xQqpKT1b3USH3360dLmhO8TwIjyDum+FC16qZWAFsk5rvd5CFl85AnRpZDmbaLQ1Q9ybC
8YPNRFdMpb7vZQoPPrvVsTtgByHYf5TtRB5rDAgMBAAECggEAUDPieCnCd1rtvwpehXElpwxzJxg6ccdaVMjwx7tuoRidHoRzeB2fUNbWvLVIGvDTjTPGAr5
I9BoFHT5tARJMeGIzbISDxsosDBRKu88cCx6dRl3ukcjSLsxMh8XUDhyWLsSgAMIpxVfHUuOsHmLZ2I3Ho6o1KIxdVg/JSgtdwTqjz3w8jmGQ/NXgc7Ym/ys1
fLG9L2nYdMzK/mRJf/BnXiCNE6/SYlZYO716oC688UJBWS3BqB9jaJyNpigX//ynJvU6xw8FhHt4fRStUmCCYAYhCQu3XgbtmxKisDGhdBVASG+DM+vVTh+
sSvxkNrjJjF+m2tSg578A8C8Ls0r3uQKBgQDpO9e178NR0HHmvWbZR9+uPugf4UT9+U2/dEfJBHAOp2GRsIvXkFwbPHuSHkc0iEPwz+U8gPC8jInSslKOUDta
GtUaVzzWrxxh7DggWx4pYs3I0Ki8C+CRTTdOY9GAFa9jhIyRmf6v9QoAH/loGNV2qYFbb+HweD0PnxlWha1txQKBgQDh9IBBltW7T96foUmHOn+x6xlF5M
NDHxLBY6bngxKvMTZoi5C6wmmCmasF45LWbkvUiMAsovYN5z4cJnKXWmRmCS8NXUucmUgdvsmCbiB62BmZvHaOffmnIdhcAjBebT/Bn5qMvKCNy3f
QFSfuEw1eRRO2IofB4o7z7m794vo25wKBgEPowrQcrZhCwwdWGn4laUGI23l80+PHFRYru0MSYbZCkiwjZXRMeiUMBUbUPhNTocSaI7rsKCweF3sbpOH/
BmkD6wySXgp8Th1M9EKnhS6zsAtKhfbK1oY4H2RZuAQ9TCYD0BIM7pU5GcJTjQD8ShsU269N8lFcERtdTbldjtOpAoGAF4YkADAa6lhjXg0loY2Gk9hdFji91
3QZuMaOLtYnkNO3zWSSWc85ut4Svxc1R1vOSz89eqgwo7vqbHXYQken4jOckXCgGZqftnERe6HJgeCTsby8PxOAdVUBuHqF3J7VH2xlY7eTo4+GVsSNFq0
```

```
nHCRm6/RmW9ohdeXh6k7CLAsCgYBZe3RLWuffKxg+lZmv9tJDOO813QPLFeixrBYhKjGDcwjVYcCugGNDmyStM0/++uWddgMKavNALjpamu8KolDNivrj
L1qaFHX9Bpi108T+dDn2WpX+vUP6hjA/U2wtTvUbJle1SsbZxRrV9gf5PAJqTrQY4u28ezjR3PCV+R4kdw==-----END PRIVATE KEY-----";

                function RSA_encryption(message, publicKey){

                        var encrypt = new JSEncrypt();
                        encrypt.setPublicKey(publicKey);
                        var encrypted = encrypt.encrypt(message);

                        return encrypted;
                }

                function RSA_decryption(ciphertext, privateKey){

                        var decrypt = new JSEncrypt();
                        decrypt.setPrivateKey(privateKey);
                        var decrypted = decrypt.decrypt(ciphertext);

                        return decrypted;
                }

        </script>

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

Q2. Test the following PHP codes:

Test demo1 here: Demo 2

1. Download **rsa.php** and put it under your student folder. Set permission as 755.
2. Link rsa.php to the code, see the first highlight part.
3. Enter a message, see the second highlight part
4. Compute the hash value of the message, see the third highlight part.
5. Get the current timestamp, see the fourth highlight part.
6. Call **PHP RSA encryption API** to encrypt the "hashed_value + & + timestamp", see the fifth highlight part
7. Call **PHP RSA decryption API** to decrypt the ciphertext, see the sixth highlight part.
8. Split the decrypted value based on the separator "&", see the seventh highlight part.

click_this.php code as below:

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

<h1>Lab 9 Demo 2:</h1>
<?php

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

// compute the hash value of the message
$hash = hash("sha256", $message);
echo 'Hashed value: ' . $hash."<br/><br/>";

//get current timestamp in second sice Jan 01 1970. (UTC)
$timestamp = time();
echo "Timestamp: ".$timestamp."<br/><br/>";
```

```
// Get the public Key
$publicKey = get_rsa_publickey('public.key');
// compute the ciphertext
$encrypted = rsa_encryption($hash."&".$timestamp, $publicKey);
echo "encrypted 'hash value + & + timestamp': ".$encrypted."<br/><br/>";

// Get the private Key
$privateKey = get_rsa_privatekey('private.key');
// compute the decrypted value
$decrypted = rsa_decryption($encrypted, $privateKey);
echo 'decrypted: ' . $decrypted."<br/><br/>";

// Split decrypted value based on "&"
echo "<br/>Split the decrypted value based on '&': <br/>";
$split_value = explode("&", $decrypted);
echo "split 1 (hashed message): " . $split_value[0]."<br/>";
echo "split 2 (timestamp): " . $split_value[1]."<br/>";
?>
</body>
</html>
```

Q3. Based on **Q1, Q2**, please update the login.html and login.php in [Lab6 answer]() to upgrade the security level of the login processing procedure.

Expected outcome: When the user logs in, the adversary may eavesdrop the hashed_password and use it to log in as the user next time, without need to know the user's plaintext password. Thus, the transmitted data during the login processing must be different every time.

Client-side:

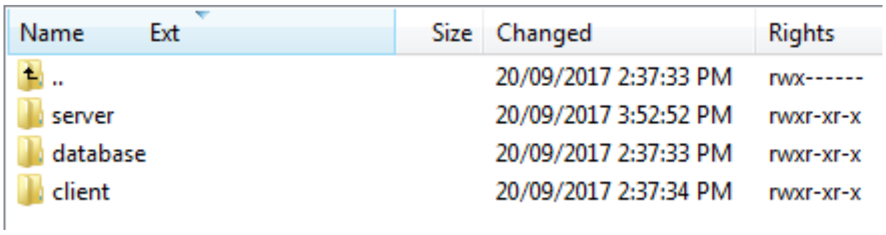
1. Enter username
2. Enter password
3. Get timestamp
4. Compute hashed_password
5. Encrypt "hashed_password+&+timestamp" with RSA encryption using Server's RSA public key
6. Submit the plaintext username and the ciphertext

Server-side:

1. Retrieve username and the ciphertext from client-side
2. Find if the username exists in the database
3. If exists, decrypt the ciphertext using Server's private key
4. Split the decrypted_value as 2 parts: client-side hashed_password and the client-side timestamp
5. Get the server-side timestamp, and compare with the client-side timestamp, the difference must be less than 150 seconds
6. If so, compare the client-side hashed_password and Server-side stored hashed_password
7. Print result on the page, such as "Cannot find the Username in the database", "Wrong password" or "Login Successful".

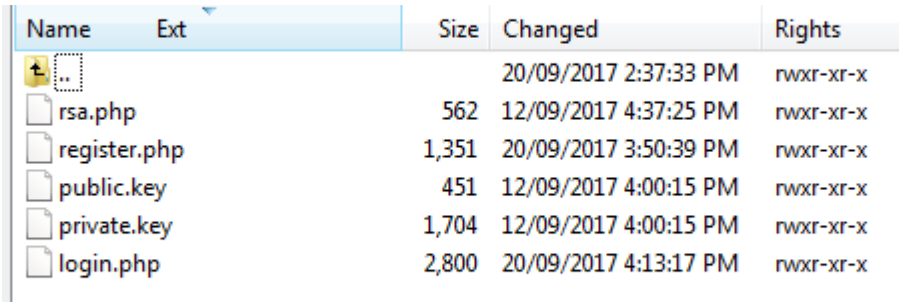
See example [here]()

Create the folders/files as below:



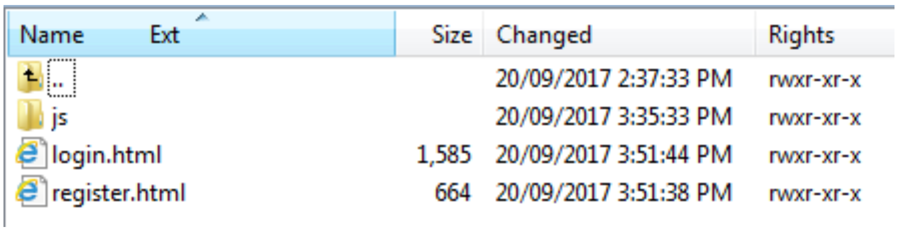
Name	Ext	Size	Changed	Rights
..			20/09/2017 2:37:33 PM	rwx------
server			20/09/2017 3:52:52 PM	rwxr-xr-x
database			20/09/2017 2:37:33 PM	rwxr-xr-x
client			20/09/2017 2:37:34 PM	rwxr-xr-x

in the folder server:



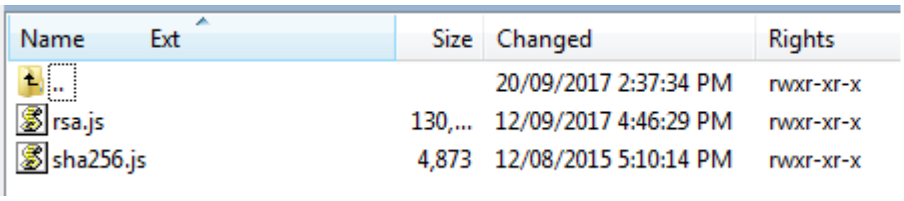
Name	Ext	Size	Changed	Rights
..			20/09/2017 2:37:33 PM	rwxr-xr-x
rsa.php		562	12/09/2017 4:37:25 PM	rwxr-xr-x
register.php		1,351	20/09/2017 3:50:39 PM	rwxr-xr-x
public.key		451	12/09/2017 4:00:15 PM	rwxr-xr-x
private.key		1,704	12/09/2017 4:00:15 PM	rwxr-xr-x
login.php		2,800	20/09/2017 4:13:17 PM	rwxr-xr-x

in the folder client:



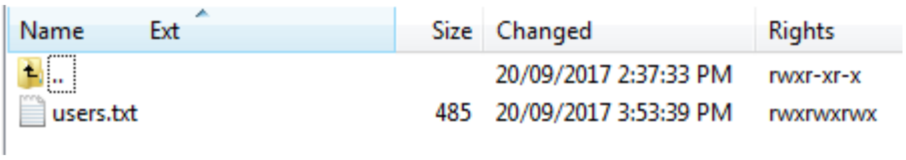
Name	Ext	Size	Changed	Rights
..			20/09/2017 2:37:33 PM	rwxr-xr-x
js			20/09/2017 3:35:33 PM	rwxr-xr-x
login.html		1,585	20/09/2017 3:51:44 PM	rwxr-xr-x
register.html		664	20/09/2017 3:51:38 PM	rwxr-xr-x

in the folder js:



Name	Ext	Size	Changed	Rights
..			20/09/2017 2:37:34 PM	rwxr-xr-x
rsa.js		130,...	12/09/2017 4:46:29 PM	rwxr-xr-x
sha256.js		4,873	12/08/2015 5:10:14 PM	rwxr-xr-x

in the folder database:



Name	Ext	Size	Changed	Rights
..			20/09/2017 2:37:33 PM	rwxr-xr-x
users.txt		485	20/09/2017 3:53:39 PM	rwxrwxrwx

Please download sha256.js, rsa.js, rsa.php, public.key and private.key from blackboard, and do not change anything from them

Answer of Q3:

login.html

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

<h2>Lab 9 Login</h2>

<form action="../server/login.php" method="POST">

Username: <input type="text" name="username" id="username">
<br/><br/>
Password: <input type="password" name="password" id="password">
<br/><br/>

<button type="submit" onclick="encrypt_password()">Submit to Login</button>
</form>

<script src="js/sha256.js"></script>
<script src="js/rsa.js"></script>
<script type="text/javascript">


                    function encrypt_password(){
                              var password = document.getElementById('password').value;
                              var hashed_password = SHA256.hash(password);
                              var timestamp = Math.floor(new Date().getTime() / 1000);

                              var encrypted = RSA_encryption(hashed_password+"&"+timestamp, pubilc_key);

                              document.getElementById('password').innerHTML = encrypted;
                              document.getElementById('password').value = encrypted;
                    }

</script>

<script type="text/javascript">

                    var pubilc_key = "-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzdxaei6bt/xIAhYsdFdW62CGTpRX+GXoZkzqvbf5oOxw4wKENjFX7LsqZ
XxdFfoRxEwH90zZHLHgsNFzXe3JqiRabIDcNZmKS2F0A7+Mwrx6K2fZ5b7E2fSLFbC7FsvL22mN0KNAp35tdADpl4lKqNFuF7NT22Z
Bp/X3ncod8cDvMb9tI0hiQ1hJv0H8My/31w+F+Cdat/9Ja5d1ztOOYIx1mZ2FD2m2M33/BgGY/BusUKqSk9W91Eh99+tHS5oTvE8Cl8g7
pvhQteqmVgBbJOa73eQhZfOQJ0aWQ5m2i0NUPcmwvGDzURXTKW+72UKDz671bE7YAch2H+U7UQeawwIDAQAB-----END PUBLIC
KEY-----";

                    function RSA_encryption(message, publicKey){

                              var encrypt = new JSEncrypt();
                              encrypt.setPublicKey(publicKey);
                              var encrypted = encrypt.encrypt(message);

                              return encrypted;
                    }

          </script>

</body>
</html>
```

login.php

```
-----------------------------------------------------------------------------------------------------------------------------
<?PHP
          include("rsa.php");
?>
<html>
<body>
<?php
          //Receive username from clint side
          $received_username = $_POST['username'];
```

```
		//Receive password from client side
		$received_password = $_POST['password'];

		if($received_username!="" & $received_password != ""){
			$find = 0;
			//read users.txt line by line
			foreach(file('../database/users.txt') as $line) {
				//split each line as two parts
				list($username, $hashed_password) = explode(",",$line);
				//verify if an exist user with the same username
				if($username == $received_username){
					$find = 1;
					echo "Find the user: ".$username."<br/><br/>";
					//verify the password
					$privateKey = get_rsa_privatekey('private.key');
					$decrypted = rsa_decryption($received_password, $privateKey);

					echo "Decrypted received_password".$decrypted."<br/><br/>";
					// $decrypted should be 'hashed_password + & + timestamp
					$split_value = explode("&", $decrypted);
					// $split_value[0] should be hashed_password
					// $split_value[1] should be timestamp

					echo "Split decrypted value as 2 parts: ";
					echo "<br/>".$split_value[0];
					echo "<br/>".$split_value[1]."<br/><br/>";
					//get current timestamp on server-side
					$timestamp = time();
					echo "Get current timestamp: ".$timestamp."<br/><br/>";

					echo "if the timestamp on server side is not greater 150 seconds than the client
submitted timestamp, <br/>";
					echo "we can compare the client-side submitted hashed_password and the
Server-side stored hashed_password <br/>";
					echo "otherwise, the server should treat the login request as invalid.<br/><br/>";

					// campare the timestamp, to make sure the submitted information did not be
copied and used later
					if($timestamp - (int)$split_value[1] <= 150 ){
						// compare the submitted hashed password and the server stored hashed
password
						if($hashed_password == $split_value[0]."\n"){

							echo "if they are same, login successful.";
							$login
= 1;

							break;
						}else{
							echo "Wrong Password;";
						}
					}else{
						echo "<br/><br/>The difference between the client-side submitted
timestamp and the current server-side timestamp is greater than 150 seconds. Invalid login request!<br/><br/>";
					}
					break;
				}
			}
			if($find==0){
				echo "<br/>Cannot find the user ->".$received_username."<- in the database<br/>";
			}

			echo "<br/>Go <a href='http://titan.csit.rmit.edu.au/~e23700/Lab9/answer'>back</a> to register,
login or check the users.txt";
		}else{
			echo "Username and Password cannot be empty!";
			echo "<br/>Go <a href='http://titan.csit.rmit.edu.au/~e23700/Lab9/answer'>back</a> to register,
login or check the users.txt";
		}
?>
</body>
</html>
```