Division of communication and network
Xi'an Jiaotong-Liverpool University

Lab 8 – Hand-on RSA algorithm

Aim

Try to use RSA algorithm to improve the security during login process.

Resources

1. You should continuously use the all settings of lab6.

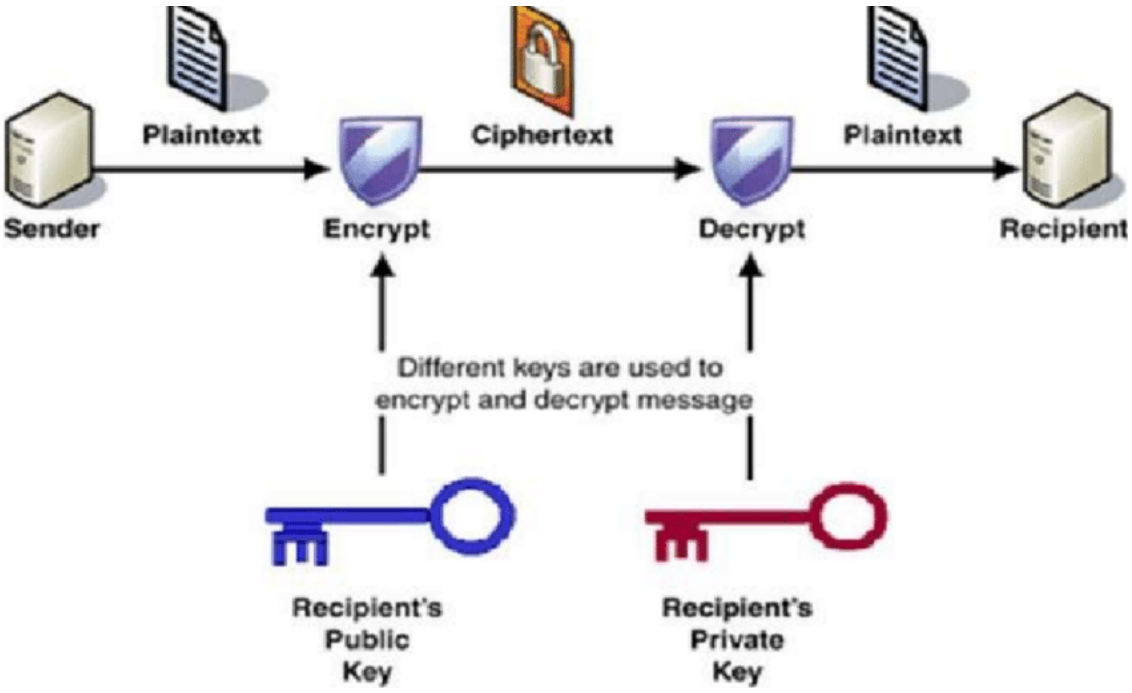
Tips:

1. If you are not sure why you are doing something, ask a TA. This is what they are here for.

2. The forums are available for questions and discussions.

3. These labs are expected take more than the 2 allocated hours. You should complete them in your own time before the next lab. Practice makes perfect!

RSA algorithm:



Remember the RSA algorithm? If not, you should try to recall it before coding.

Improve the login function:

Let's use RSA algorithm to against EVE in the following way:
We will use "openssl" to generate the RSA key pairs.
Normally, Mac and Linux systems have "openssl" already. For a Windows system, you need to install "openssl" in your system.

Let's use "openssl" to generate the RSA key pairs:

Using the following command, you can get a PEM file with the RSA private key.

```
[ping@PingdeMacBook-Pro Desktop % openssl genrsa -out rsa_private_key.pem 1024
Generating RSA private key, 1024 bit long modulus
..............++++++
...................++++++
e is 65537 (0x10001)
```

It looks like:

```
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQC8eVT+pVl31LIqDc5x2FH+QljWEG8c7LrPNBzX+urYS/K2jaUX
iUMKNU9PEDq3ZETSWNTGfqITROyuC8vawiy3q/JcDdHNSOljM841zv7l816oNxEQ
pwyCVluOAAneIy9iVwmtbMpePqKDEoSrn7QE1s+W/6I5HhEFAZnQ8DZs6wIDAQAB
AoGBALcXIyNRG63WONGzodZkb5qRd11Uj6xIqF07Yb3KqjM+7GS9CyDnHfIfwZCr
0m5vgI/a7bB6OhaAAXA+U2WK9gY1artBlcHkek63fshY3iaRfoHySMtEHygpINtr
zBfN2Nw4TtmTBt3wrOwPbqGTqnyhTLD+sT36a4yVLQH4LV7BAkEA+PbgzpODzSWS
lUiJSRQ+7mql2OE/IUs8vtxI7P+BU9/I/qtlUsL6O1vovc4Z0YPFWiNxUkqoWbrQ
KycEp7+YSwJBAMHM1q4tCOqi7hUNdyAAS8RiA+q3JQQO8wW/BG3sEMezUAt6P54l
i23PWxD7301IDjCXypxkc0QQLgoJPanu2eECQAV4ZzgixaKcULw2+80/RKK4dSxu
xpRUsuD+tht/Abh2ElSGL5PB9P2Y52REQwz3eD6iyLqmKUzPbgOEt/V3oEECQBNi
RJq5QGoPj9alOSQHQ4zJ7PBeDyK/yAjsGSpRcUA4LCppuNE9mhuKoOYq+yPEsD6m
AArydSD6qVAxqmxDyqECQAjtPCkUT5tiUCih13AgxZCFtQkURb1I48TsJtRd6I3n
jN1YXwsVMfWZaYbHwXRroKRg8x4L+3qAzeIWW6XErZo=
-----END RSA PRIVATE KEY-----
```

Then, deducing an PEM file with the RSA public key.

```
% openssl pkcs8 -topk8 -inform PEM -in rsa_private_key.pem -outform PEM -nocrypt -out private_key.pem
%
```

It looks like:

```
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC8eVT+pVl31LIqDc5x2FH+QljW
EG8c7LrPNBzX+urYS/K2jaUXiUMKNU9PEDq3ZETSWNTGfqITROyuC8vawiy3q/Jc
DdHNSOljM841zv7l816oNxEQpwyCVluOAAneIy9iVwmtbMpePqKDEoSrn7QE1s+W
/6I5HhEFAZnQ8DZs6wIDAQAB
-----END PUBLIC KEY-----
```

Then we can use them to upgrade the "login.php".

In login.php part, we need to modify it as:

```
    <script src="https://cdn.bootcdn.net/ajax/libs/jsencrypt/3.2.1/jsencrypt.min.js"></script>
    <script>
        function security(){
            var pass = document.getElementById("pass").value;
            pubKey = '-----BEGIN PUBLIC KEY-----
            MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC8eVT+pVl31LIqDc5x2FH+QljWEG8c7LrPNBzX+urYS/K
            2jaUXiUMKNU9PEDq3ZETSWNTGfqITROyuC8vawiy3q/JcDdHNSOljM841zv7l816oNxEQpwyCVluOAAneIy
            9iVwmtbMpePqKDEoSrn7QE1s+W/6I5HhEFAZnQ8DZs6wIDAQAB-----END PUBLIC KEY-----';
            var encryptor = new JSEncrypt();
            encryptor.setPublicKey(pubKey);
            var encryptData = encryptor.encrypt(pass);
            document.getElementById("pass").value = encryptData;
        }
    </script>
```

"jsencrypt" is a js library that support RSA encryption and decryption. With the above js function security, we can encryption the password by the public key.

On the receiving part, we need to modify it accordingly:

```
<?php
$private_key = '-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQC8eVT+pVl31LIqDc5x2FH+QljWEG8c7LrPNBzX+urYS/K2jaUX
iUMKNU9PEDq3ZETSWNTGfqITROyuC8vawiy3q/JcDdHNSOljM841zv7l816oNxEQ
pwyCVluOAAneIy9iVwmtbMpePqKDEoSrn7QE1s+W/6I5HhEFAZnQ8DZs6wIDAQAB
AoGBALcXIyNRG63WONGzodZkb5qRd11Uj6xIqF07Yb3KqjM+7GS9CyDnHfIfwZCr
0m5vgI/a7bB6OhaAAXA+U2WK9gY1artBlcHkek63fshY3iaRfoHySMtEHygpINtr
zBfN2Nw4TtmTBt3wrOwPbqGTqnyhTLD+sT36a4yVLQH4LV7BAkEA+PbgzpODzSWS
lUiJSRQ+7mql2OE/IUs8vtxI7P+BU9/I/qtlUsL6O1vovc4Z0YPFWiNxUkqoWbrQ
KycEp7+YSwJBAMHM1q4tCOqi7hUNdyAAS8RiA+q3JQQO8wW/BG3sEMezUAt6P54l
i23PWxD7301IDjCXypxkc0QQLgoJPanu2eECQAV4ZzgixaKcULw2+80/RKK4dSxu
xpRUsuD+tht/Abh2ElSGL5PB9P2Y52REQwz3eD6iyLqmKUzPbgOEt/V3oEECQBNi
RJq5QGoPj9alOSQHQ4zJ7PBeDyK/yAjsGSpRcUA4LCppuNE9mhuKoOYq+yPEsD6m
AArydSD6qVAxqmxDyqECQAjtPCkUT5tiUCih13AgxZCFtQkURb1I48TsJtRd6I3n
jN1YXwsVMfWZaYbHwXRroKRg8x4L+3qAzeIWW6XErZo=
-----END RSA PRIVATE KEY-----';

$pri_key = openssl_pkey_get_private($private_key);
$decrypted = '';
session_start(); // magic word to start a session.
include("includes/db.php");
if (isset($_POST['login'])) {
    $email = mysqli_real_escape_string($con,$_POST['email']);
    $pass = mysqli_real_escape_string($con,$_POST['pass']);
    openssl_private_decrypt(base64_decode($pass), $decrypted, $pri_key);
    $pass = $decrypted;
```

Php has the openssl extension and support the encryption and decryption functions. Here, we use the private key to decrypt the password to plain text before the database query.

Please try to make it work and you can use echo in php or alert in js to output the variables during the debug process.