A MORE SECURE LOGIN FORM

with Jcryption HTML-Form encryption plugin

Without https our precious data travel unsicure across the net.
Jcryption HTML-Form encryption plugin
send our form data encrypted

TABLE OF CONTENTS

INTRODUCTION

The login secure form is a solution using two script language, one on user side (browser) and the other on server side (PHP) that works togheter to obscure (crypt) the post/get data; and decrypt them at server side.

All security is about levels though - nothing is secure. This library obviously fills a specific level, and provides some protection. It'd certainly make some guy in the IT dept packet sniffing likely give up.
You're not going to use it for a banking website.

The following description is excerpted form Jcryption site (http://www.jcryption.org/info/).

Normally if you submit a form and you don't use SSL, your data will be sent in plain text. But SSL is neither supported by every webhost nor it's easy to install/apply sometimes. So Daniel Griesser created this plug-in in order that you are able to encrypt your data fast and simple. jCryption uses the public-key algorithm of RSA for the encryption.

jCryption at it's current state is no replacement for SSL, because there is no authentication, but the main goal of jCryption should be a very easy and fast to install plugin which offers a base level of security.

The way jCryption is, that the data is encrypted on the client (javascript) and decrypted your virtual private server with PHP.

jCryption was tested with Internet Explorer 6 +, Mozilla Firefox 3+, Safari 3, Opera 9+, Google Chrome.

Some of the features are:

- RSA form data encryption up to 2048 bit
- AjaxSubmit supported
- no SSL required
- easy to install, use and extend
- doesn't block the browser on calculations

Requirement :

jquery.jcryption.js require **jQuery 1.4.2+ (already tested with jQuery 1.6.1)**
jCryption PHP class require **bcmath extension**

CHAPTER 1: INSTALLATION

1.1. PREREQUISITES

INTENDED AUDIENCE

This guide is intended for developers that are familiar that wish to desingn a login script and cannot use a server with Https capability.

We expect the reader to have some basic knowledge of system administration. Of course, as this is a PHP –Javascript solution, so knowing PHP 5 and Javascript is a huge advantage.

WEB SERVER

The first thing you want to do is ensure that you have a standard, working installation of a webserver (e.g. Apache, IIS, etc.). In example

PHP 5

The scripts are developed for PHP5, and framework are not tested with version 4 of PHP. The main reason is that PHP 5 has a completely redesigned, mature object model.

DATABASE

No database is required to crypt/decrypt, but login process can use, if present, a Ldap to validate user/password and and a user table to select administrative right (based on user type).

1.2. INSTALLATION IN TESTED ENVIRONMENT.

WINDOWS

Installation instructions and scripts presented in this tutorial are tested on Operating System Windows XP S.P.3 (installed on virtual vmware machine)

Note: the instructions in this tutorial are for a **WAMP5** installation. You may need to modify them slightly for a different configuration on your machine.

Details of installed environment are:
  WAMP5 Version 1.7.2, http://sourceforge.net/projects/wampserver/files/WAMP5/

Host name of my local apache installation is `grossinixv`, so my installed apache navigation pages start with http://grossinixv.

Browser used in example panels is SeaMonkey/2.4.1.-english version.

Tested browsers:

- Internet Explorer 6,7, 8
- Firefox 3.x
- Seamonkey 1.1.17italian

Step 1: Downloading Jcryption

Jcription can be retrieved from the download page: http://jcryption.googlecode.com/files/jCryption-1.2.zip .
This link download last version (1.2) of library.
Documentation informs users of previous version, that Cryption 1.1 brought some major changes and simplified alot.
Most of the old options and callback functions are gone. Author thought they make to code alot more confusing so he just removed them. But we should be able to do the same things like before. Neverless previous version and examples can be found here http://jcryption.googlecode.com/svn/trunk/jCryption-1.0.1.package.zip .

Step 2: Unpacking Jcryption

The file we retrieved is a compressed zip file. Unzip it in temporary folder (c:\temp).

Step 3: Moving files to their proper location

Now that we retrieved and unpacked the files, we need to put the files on their proper location. The default location of the webroot on Apache in WAMP5 is ***c:\wamp\www***. Verify this with your installation of Apache. The location is specified in the httpd.conf under the following setting:

```
DocumentRoot "C:/wamp/www"
```

To create ou login solution, we need to copy the files from the temporary unzipped directory to this directory. As most of you will already have something in the default directory, we will do this in a subdirectory ***login_crypted*** (name used in this tutorial), by issuing the following command from the location where we unpacked the downloade-zip package

```
> move  c:\temp\  C:\wamp\www\login_crypted\
```

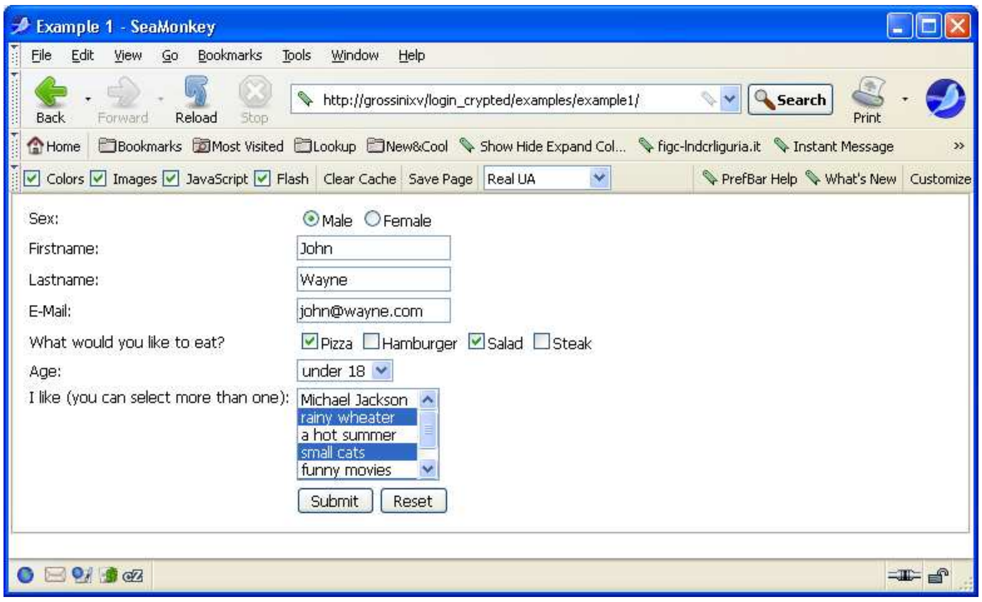
Step 4: Setting permissions

This step is not necessary in a Windows environment.

Step 4: Test JCryption installed library

The library come with some example so we can try them to test our installation and familiarize with library functions.
(*pay attention that examples need navigation to net , you can avoid this downloading jquery and calling it locally* "http://ajax.googleapis.com/ajax/libs/jquery/1.6.1/jquery.min.js”).

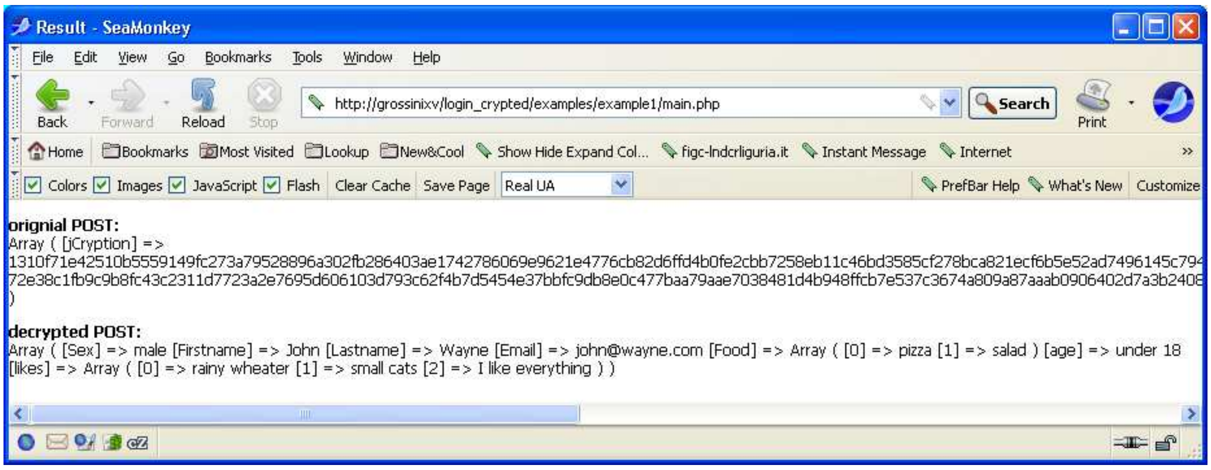
Let try example 1 navigating to http://grossinixv/login_crypted/examples/example1/ (grossinixv is my local host name) :


Example 1 - SeaMonkey
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://grossinixv/login_crypted/examples/example1/
Search
Print
Home Bookmarks Most Visited Lookup New&Cool Show Hide Expand Col... figc-lndcrliguria.it Instant Message
Colors Images JavaScript Flash Clear Cache Save Page Real UA PrefBar Help What's New Customize

Sex:	(● Male) ○ Female
Firstname:	John
Lastname:	Wayne
E-Mail:	john@wayne.com
What would you like to eat?	☑ Pizza ☐ Hamburger ☑ Salad ☐ Steak
Age:	under 18
I like (you can select more than one):	Michael Jackson, [✓ rainy wheater], a hot summer, [✓ small cats], funny movies

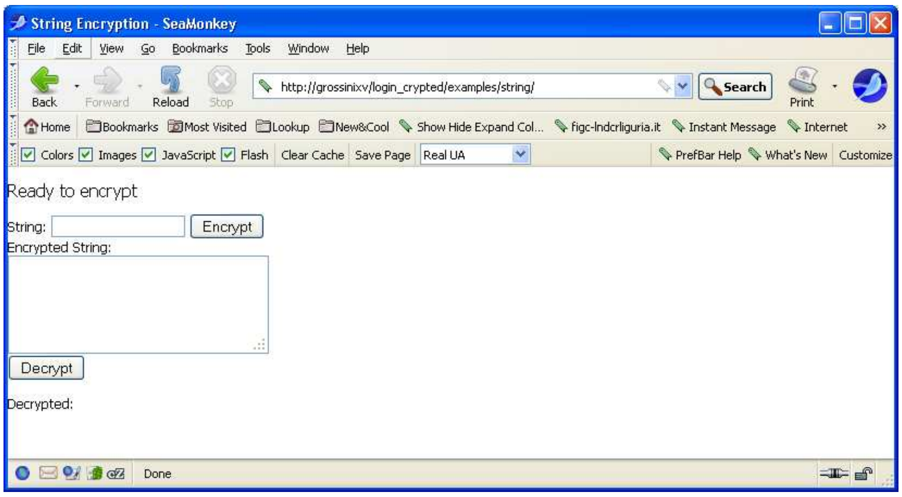
Submit Reset

Here is result of submit if installation is OK:



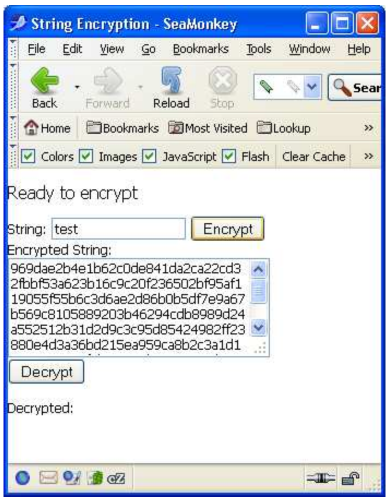
```
orignial POST:
Array ( [jCryption] =>
1310f71e42510b5559149fc273a79528896a302fb286403ae1742786069e9621e4776cb82d6ffd4b0fe2cbb7258eb11c46bd3585cf278bca821ecf6b5e52ad7496145c794
72e38c1fb9c9b8fc43c2311d7723a2e7695d606103d793c62f4b7d5454e37bbfc9db8e0c477baa79aae7038481d4b948ffcb7e537c3674a809a87aaab0906402d7a3b2408
)

decrypted POST:
Array ( [Sex] => male [Firstname] => John [Lastname] => Wayne [Email] => john@wayne.com [Food] => Array ( [0] => pizza [1] => salad ) [age] => under 18
[likes] => Array ( [0] => rainy wheater [1] => small cats [2] => I like everything ) )
```

Follow example string:


String Encryption - SeaMonkey
http://grossinixv/login_crypted/examples/string/
Encrypt
Decrypt
Done

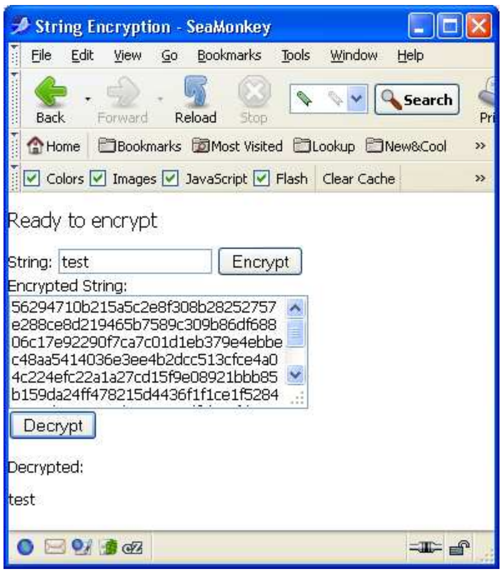
```
Ready to encrypt
String:
Encrypted String:

Decrypted:
```

Push Encrypt



```
Ready to encrypt
String: test
Encrypted String:
969dae2b4e1b62c0de841da2ca22cd3
2fbbf53a623b16c9c20f236502bf95af1
19055f55b6c3d6ae2d86b0b5df7e9a67
b569c8105889203b46294cdb8989d24
a552512b31d2d9c3c95d85424982ff23
880e4d3a36bd215ea959ca8b2c3a1d1
Decrypted:
```

Push Decrypt



```
Ready to encrypt
String: test
Encrypted String:
56294710b215a5c2e8f308b28252757
e288ce8d219465b7589c309b86df688
06c17e92290f7ca7c01d1eb379e4ebbe
c48aa5414036e3ee4b2dcc513cfce4a0
4c224efc22a1a27cd15f9e08921bbb85
b159da24ff478215d4436f1f1ce1f5284
Decrypted:
test
```

CHAPTER 2: OUR SECURE LOGIN APPLICATION

1.1. PREPARING APPLICATION DIRECTORY

For our application we prefer use our standard organization that have library modules in a lib subdir so we rework our directory moving `jquery.jcryption.js` and `jcryption.php` to a new `lib` directory. We create also a `user` empty dir.

```
Directory di C:\wamp\www\login_crypted
   30/10/2011  22.16    <DIR>          .
   30/10/2011  22.16    <DIR>          ..
   30/10/2011  12.26    <DIR>          user
   31/10/2011  08.37    <DIR>          lib
```

To avoid internetnet navigation to find jquery library we downloaded Jquery lib so our lib directory become:

```
Directory di C:\wamp\www\login_crypted\lib
   30/10/2011  13.19   <DIR>        examples
   17/05/2011  22.23         16.039 jcryption.php
   30/10/2011  18.31         91.342 jquery-1.6.1.min.js
   17/05/2011  23.01         17.239 jquery.jcryption.js
```

Let's create a sample login application: a page with a button and two field.

Our Login application consists of 2 files. These will be :

- ***index.php***, contains the **user side** of our application
- ***main.php***, contains the **server side logic** of our application

Complete zipped application is avaible here: http://digidownload.libero.it/magiainformatica/login_crypted.zip

1.2. INDEX.PHP

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Login </title>
<style type="text/css">
html,body {
 margin:0;
 padding:0;
 font-family:Tahoma;
 font-size:12px;
}
input,textarea,select {
 font-family:Tahoma;
 font-size:12px;
}
</style>
<script type="text/javascript" src="./lib/jquery-1.6.1.min.js"></script>
<script type="text/javascript" src="./lib/jquery.jcryption.js" ></script>
<script type="text/javascript">
$(document).ready(function() {
 var $status = $('<div id="status" style="margin-top:15px;"><img src="loading.gif"
alt="Loading..." title="Loading..." style=" margin-right:15px;" /><span></span></div>').hide();
 var $statusText = $status.find("span");
 $("#status_container").append($status);

 $("#fullForm").jCryption({
    collectionSpeed:250,
    encryptionSpeed:10,
    beforeEncryption:function() {
       $('#error').remove();
       $statusText.text("Loading Keys ...").parent().show();
       return true;
    }
 });

});
</script>
</head>

<body>
&nbsp;<br>
<form action="main.php" method="post" id="fullForm" class="general">

<table style="width: 60%; height: 142px;" cellpadding="0" cellspacing="0" bgcolor="#cccccc">
<!--<table border="0" cellspacing="5" cellpadding="0"> -->
<tbody>
<tr><td colspan=3 align=center><b>Login &nbsp;&nbsp;</b><font color=#E8E8E8><small> crypted form
    by ma.gi.a</small></font></td></tr>
<tr>
<td width="8%" ><img src="lock_open.png" /></td>
<td width="10%">User:</td>
<td><input class="text" name="User" type="text" /></td>
</tr>
<tr>
<td width="8%"><img src="lock_open.png" /></td>
<td width="10%">Password:</td>
<td><input class="text" name="Passw" id="Passw" type="password" /></td>
</tr>
<tr>
<td>&nbsp;</td>
<td></td>
<td><input title="Submit" alt="Submit" name="submitButton" type="submit" value="Submit"
class="submit" /> <input title="Reset" alt="Reset" name="reset" type="reset" value="Reset"
/></td>
</tr>
</tbody></table>

</form>
<div id="status_container" style="text-align:center;"></div>

</body>
</html>
```

Example breakdown

Let's take a closer look at login.php line by line:

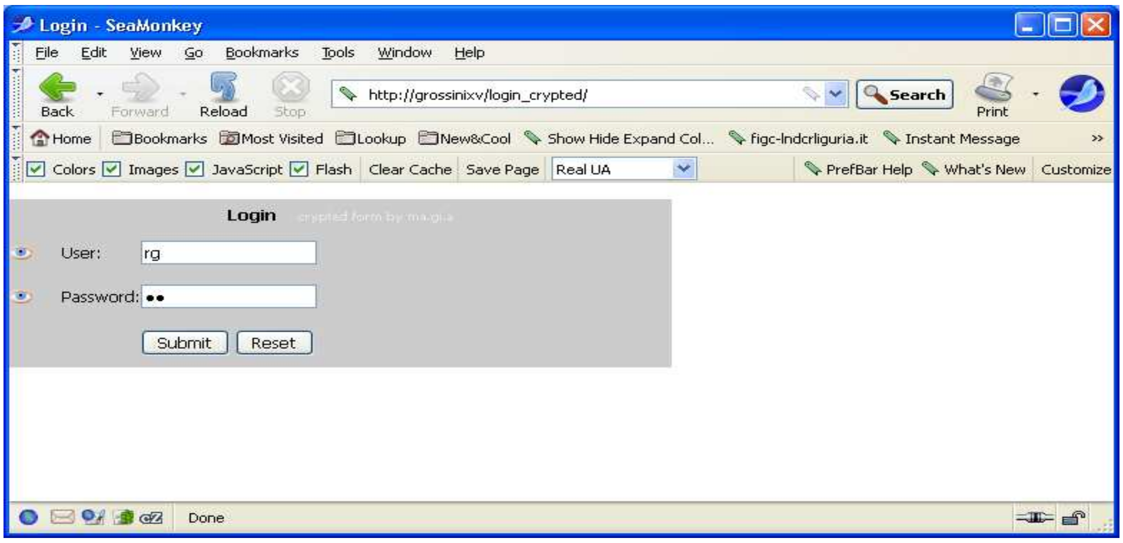
This two lines load javascript modules from lib directory:

```
<script type="text/javascript" src="./lib/jquery-1.6.1.min.js"></script>
<script type="text/javascript" src="./lib/jquery.jcryption.js" ></script>
```

Javascript jCryption function informs about Aiax activity with server who create rsa asymmetric key and send public key to browser who use it to crypt login data.
If you need a loading.gif you can search from many example in google or you can download it, as example, here http://www.oppenheim.com.au/wp-content/uploads/2007/08/ajax-loader-3.gif -. Remember to change its name and put it in login_crypted dir . It's up to you solve lock_open.png similar problem.

```
$(document).ready(function() {

 var $status = $('<div id="status" style="margin-top:15px;"><img src="loading.gif" alt="Loading..."
title="Loading..." style=" margin-right:15px;" /><span></span></div>'  ).hide();
 var $statusText = $status.find("span");
 $("#status_container").append($status);

 $("#fullForm").jCryption({
    collectionSpeed:250,
    encryptionSpeed:10,
    beforeEncryption:function() {
       $('#error').remove();
       $statusText.text("Loading Keys ...").parent().show();
       return true;
    }
 });

});
```

Now, if we would fire up our browser, and load the ***index.php*** page:


Login - SeaMonkey
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
Search
Print
Home Bookmarks Most Visited Lookup New&Cool Show Hide Expand Col... figc-lndcrliguria.it Instant Message
Colors Images JavaScript Flash Clear Cache Save Page Real UA PrefBar Help What's New Customize
Done

http://grossinixv/login_crypted/

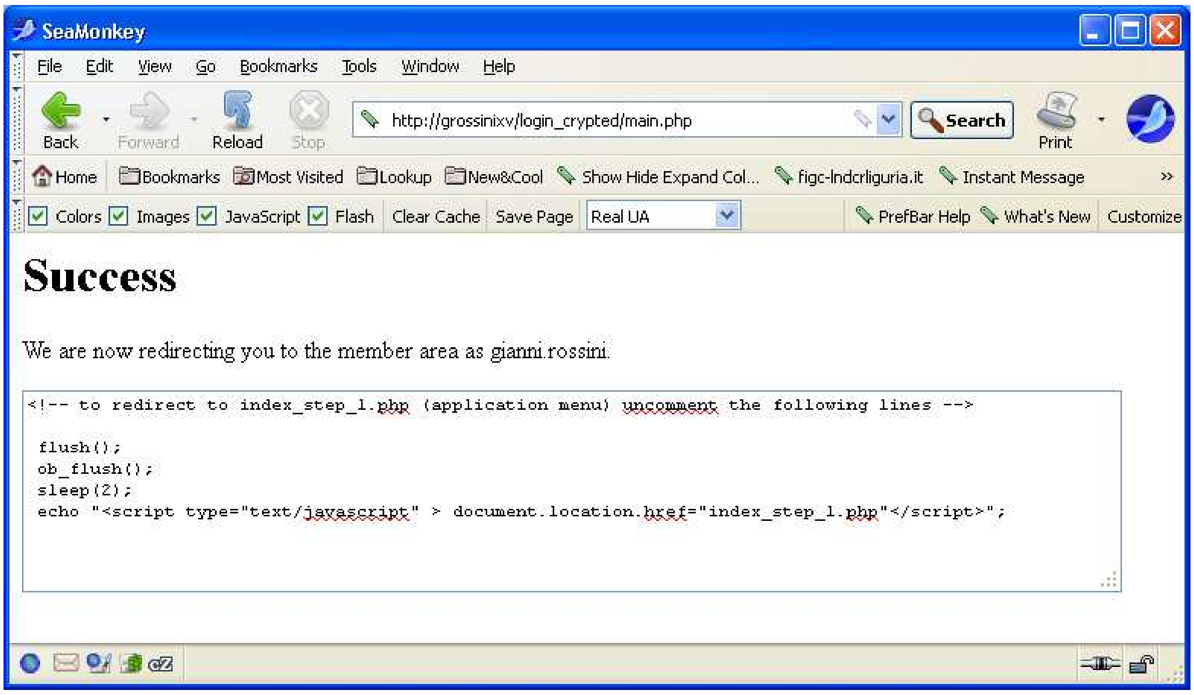
Login

User: rg

Password: ••

Submit Reset

And the expected result :is

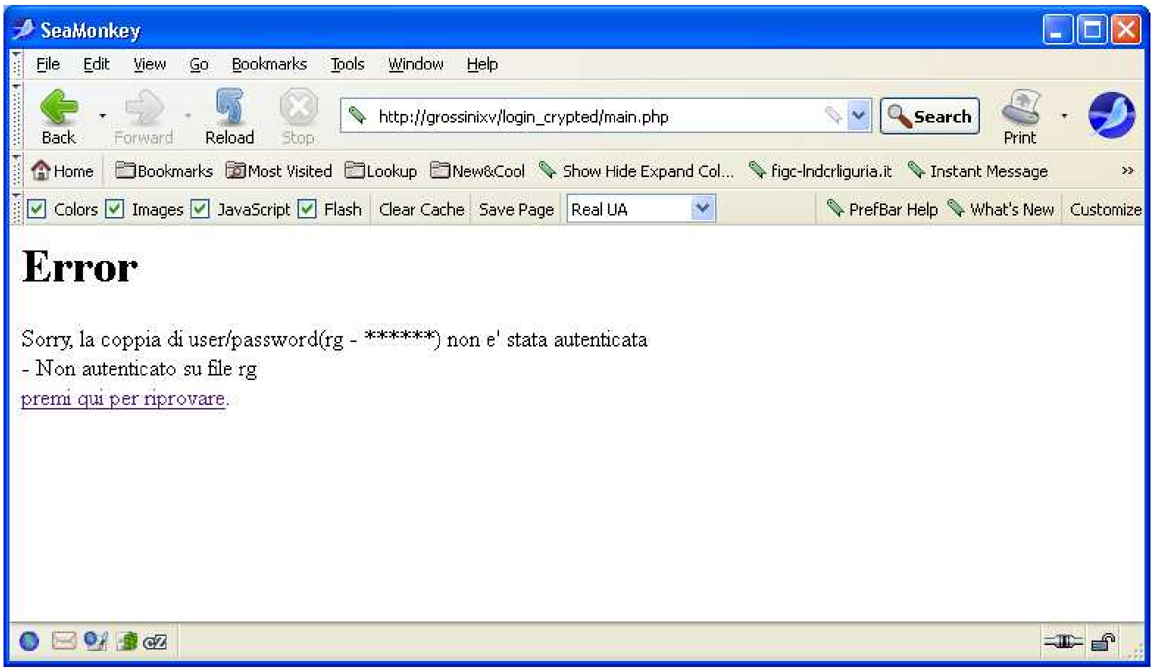

SeaMonkey
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://grossinixv/login_crypted/main.php
Search
Print
Home Bookmarks Most Visited Lookup New&Cool Show Hide Expand Col... figc-lndcrliguria.it Instant Message
Colors Images JavaScript Flash Clear Cache Save Page Real UA PrefBar Help What's New Customize

Success

We are now redirecting you to the member area as gianni.rossini.

```
<!-- to redirect to index_step_1.php (application menu) uncomment the following lines -->

 flush();
 ob_flush();
 sleep(2);
 echo "<script type="text/javascript" > document.location.href="index_step_1.php"</script>";
```

And if invalid user name password sent (allow firewall if you want go imap – user not present in user dir):


SeaMonkey
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://grossinixv/login_crypted/main.php
Search
Print
Home Bookmarks Most Visited Lookup New&Cool Show Hide Expand Col... figc-lndcrliguria.it Instant Message
Colors Images JavaScript Flash Clear Cache Save Page Real UA PrefBar Help What's New Customize

Error

Sorry, la coppia di user/password(rg - ******) non e' stata autenticata
- Non autenticato su file rg
premi qui per riprovare.

1.3. MAIN.PHP

Now server side of login, our clever main.php who take care of key generation and form data decryptation:

```
<?php
session_start();
require_once("./lib/jcryption.php");

$keyLength = 256;
$jCryption = new jCryption();

if(isset($_GET["generateKeypair"]))
{
    if (empty($_SESSION["e"])){

      // logic added by ma.gi.a. di rossini to avoid the key generation
      // every time we use this module
      // asymmetric keys are once generated and saved in a file using  array2string() -
      // To get the keys from file we use string2array()
      if ((!file_exists("lib/keypair.txt")))
      {
       $keys = $jCryption->generateKeypair($keyLength);
       // write file with already generated keys
            array2string($keys,$output,$parent);
            // Store the string in a file
            $f1 = fopen("lib/keypair.txt","w+");
            fwrite($f1,$output);
            fclose($f1);
       }
      else
      {
          // Read the file back from the disk
            $f1 = fopen("lib/keypair.txt","r");
            $newString = fread($f1,filesize('lib/keypair.txt'));
            fclose($f1);
         // Convert the content back to an array
          string2array($newString, $keys);
      }

      // sets the keys in the session to have them ready when you submit the form
      // please keep this structure.
      // You'll need the hex value of the key for javascript
      // and the int value of the key for PHP. e = public key, d = private key, n = modulo

      $_SESSION["e"] = array("int" => $keys["e"], "hex" => $jCryption->dec2string($keys["e"],16));
      $_SESSION["d"] = array("int" => $keys["d"], "hex" => $jCryption->dec2string($keys["d"],16));
      $_SESSION["n"] = array("int" => $keys["n"], "hex" => $jCryption->dec2string($keys["n"],16));
     }

     //returns the needed keys for the javascript part in a JSON string
   //maxdigits is need for the javascript and calculated like ($keyLength * 2 / 16 + 3)

     /* debug
     $f1 = fopen("keyparhex.txt","w+");
     fwrite($f1,
'{"e":"'.$_SESSION["e"]["hex"].'","n":"'.$_SESSION["n"]["hex"].'","maxdigits":"'.intval($keyLength*2
/16+3).'"}');
     fclose($f1);
     */
     echo
'{"e":"'.$_SESSION["e"]["hex"].'","n":"'.$_SESSION["n"]["hex"].'","maxdigits":"'.intval($keyLength*2
/16+3).'"}';

  } else {
```

```
//
//here the decrypt function is called. The first parameter is the encrypted POST.
//Second parameter is the private key d in it's int form and third the modulo n.
//This function will return your orginal decrypted POST.
$var = $jCryption->decrypt($_POST['jCryption'], $_SESSION["d"]["int"], $_SESSION["n"]["int"]);

//unset($_SESSION["e"]);
//unset($_SESSION["d"]);
//unset($_SESSION["n"]);

  // parse the query string from post -  into variables
parse_str($var,$result);
$posted = $_POST['jCryption'];

 //debug
/*
$f1 = fopen("received.txt","w+");
        fwrite($f1,"\r\n-posted-\r\n");
        fwrite($f1,$posted);
        fwrite($f1,"\r\n- var decrypted -\r\n");
        fwrite($f1,$var);
        fclose($f1);

 */
//echo "<br> decrypted data are:" .$var;
//$result["User"] = "rg";
//$result["Passw"] = "rr";

 /* original code commented - with password we do IMAP authentication
if (preg_match('/\b[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}\b/i', $result["Passw"])) {
   echo "true";
} else {
   echo "false";
}
 */
     $username = $result["User"];
     $password = $result["Passw"];
     $where_we_are = pathinfo($_SERVER['SCRIPT_FILENAME']) ;
     $root= $where_we_are['dirname']."/";

     // if not exists username file try with LDAP – not used in this example along with includes
     if ((!file_exists($root."user/".$username.".txt")))
     {
    // try LDAP autentication
      $imap_host[] = "10.103.8.21";  // mailpp01.postel.it
      $imap_port = "143";
      $ldap_ok = 0;
      require("lib/auth_imap.inc");
      if(authValidateUser($username, $password))
      {
            // echo " Ldap ". $imap_host[0] . " AUTENTICATO - ";
            $ldap_ok = 1;
            $_SESSION['UUser'] = $username;
            $_SESSION['LoggedIn'] = 2;
      }
      else
      {
            $msg_Ldap = " Ldap 10.103.8.21 NON autenticato ";
            //echo $msg_Ldap;
      }
     }
      // if exists username file
   if ((file_exists($root."user/".$username.".txt")))
   {
         $msglocalfile = "Non autenticato su file $username";
```

```
            $fp=fopen($root."user/".$username.".txt","r");
            $contents=fread($fp,filesize($root."user/".$username.".txt"));
            fclose($fp);
            $info=explode("|",$contents);
            $username1=$info[0];
            $password1=$info[1];
            //if($password1==$_SESSION['password'])
            if($password1 == $password){
                  //$firstuser = $temp_username;
                  $username = $username1;
                  $msglocalfile = " Autenticato su file $username ";
                  $_SESSION['UUser'] = $username;
                  $_SESSION['LoggedIn'] = 2;
            }else {
                 session_unregister('LoggedIn') ;
                 session_unregister('UUser');
            }
            // remove comment in next line to have this info at top
            //echo $msglocalfile;
         }
         // return from authentication
         //echo " -- $username1 -- $password1";
         if(isset($_SESSION['UUser']))
           {

            //$_SESSION['LoggedIn'] = 1;

             echo "<h1>Success</h1>";
             echo "<p>We are now redirecting you to the member area.</p>";
            echo ' <TEXTAREA ROWS=8 COLS=100 WRAP="off" name=textcode>';
            echo '&lt;!-- to redirect to index_step_1.php (application menu) uncomment the following
lines --&gt;'."\n\r flush();\n ob_flush();\n sleep(2);\n echo ".
'"&lt;script type="text/javascript" &gt;
document.location.href="index_step_1.php"&lt;/script&gt;";';
         echo '</TEXTAREA>';

         // display message ...
         // flush();
         // ob_flush();
          sleep(2);
       // redirect
       // - this not works sometime in IE - see Security tab, then choose Custom Level and the Meta
Tag Refresh
       //   echo "<meta http-equiv='refresh' content='=2;index.php' />";
       // - this give warning
       //   header("refresh: 10; index.php");
       // echo "<script type='text/javascript'>document.location.href='index_step_1.php'</script> ";
            //echo "true";

        }
      else
        {

               echo "<h1>Error</h1>";
               //echo " root is $root" . $msglocalfile."<br>";
               echo "<p>Sorry, la coppia di user/password($username - ******) non e' stata
autenticata<br>";
               echo " $msg_Ldap - $msglocalfile <br>";
               echo "<a href=\"index.php\">premi qui per riprovare</a>.</p>";

         // per test il contenuto della password appare nella schermata
         // echo $password;
         //echo "false";
        }
  }
```

```
   /**
    * This function converts an array into a separated string
    *
    * @param Array $myarray The array to convert to string
    * @param String $output The reference to the output string
    * @param String $parentkey It is a helper variable
    */
   function array2string($myarray,&$output,&$parentkey){
      foreach($myarray as $key=>$value){
         if (is_array($value)) {
            $parentkey .= $key."·";
            array2string($value,$output,$parentkey);
            $parentkey = "";
         }
         else {
            $output .= $parentkey.$key."·".$value."\n";
         }
      }
   }

   /**
    * This function converts a separated string into an array
    *
    * @param String $string The string to convert into an Array
    * @param Array $myarray The array to store the output
    */
   function string2array($string,&$myarray){
      $lines = explode("\n",$string);
      foreach ($lines as $value){
         $items = explode("·",$value);
         if (sizeof($items) == 2){
            $myarray[$items[0]] = $items[1];
         }
         else if (sizeof($items) == 3){
            $myarray[$items[0]][$items[1]] = $items[2];
         }
      }
   }

?>
```

Let's take a closer look at main.php:
The script is extensively commented so we recall you here only on the fact that there is part of code related to Ldap authentication and normaly, in local installation, we don't have this feature. So, to let us design and debug our script, I added code that uses to compare user/password with simple text files located in user dir named from USER field:

Directory di c:\wamp\www\login_crypted\user

```
30/10/2011  12.26    <DIR>          .
30/10/2011  12.26    <DIR>          ..
29/11/2009  18.02                17 rg.txt
29/11/2009  18.19                16 rs.txt
```

In my installation rg.txt has two fields separated by | (user for access in mysql user table and password to compare with entered field):

```
gianni.rossini|rr
```

The following line redirect to application menu (not present) if valid logon data entered and can be commented out .

```
echo "<script type='text/javascript'>document.location.href='index_step_1.php'</script> ";
```