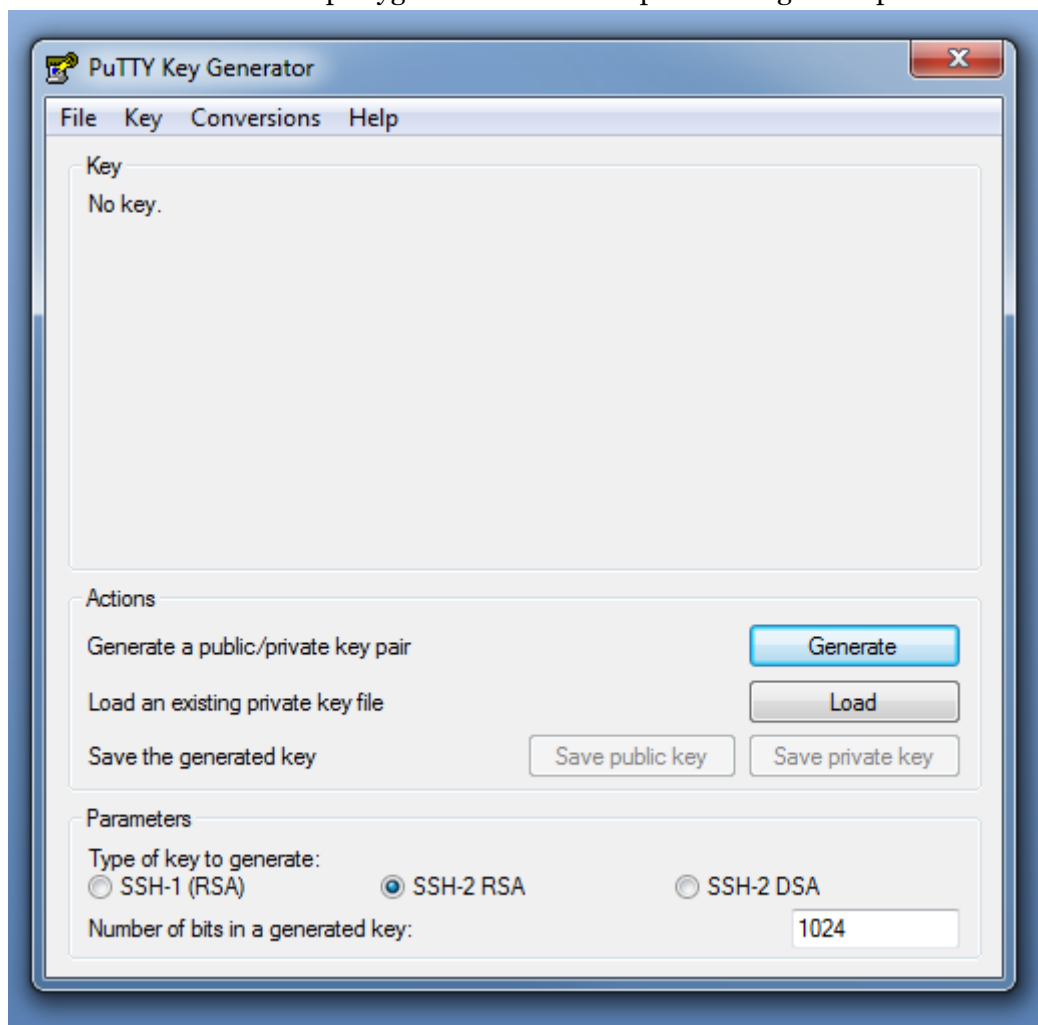
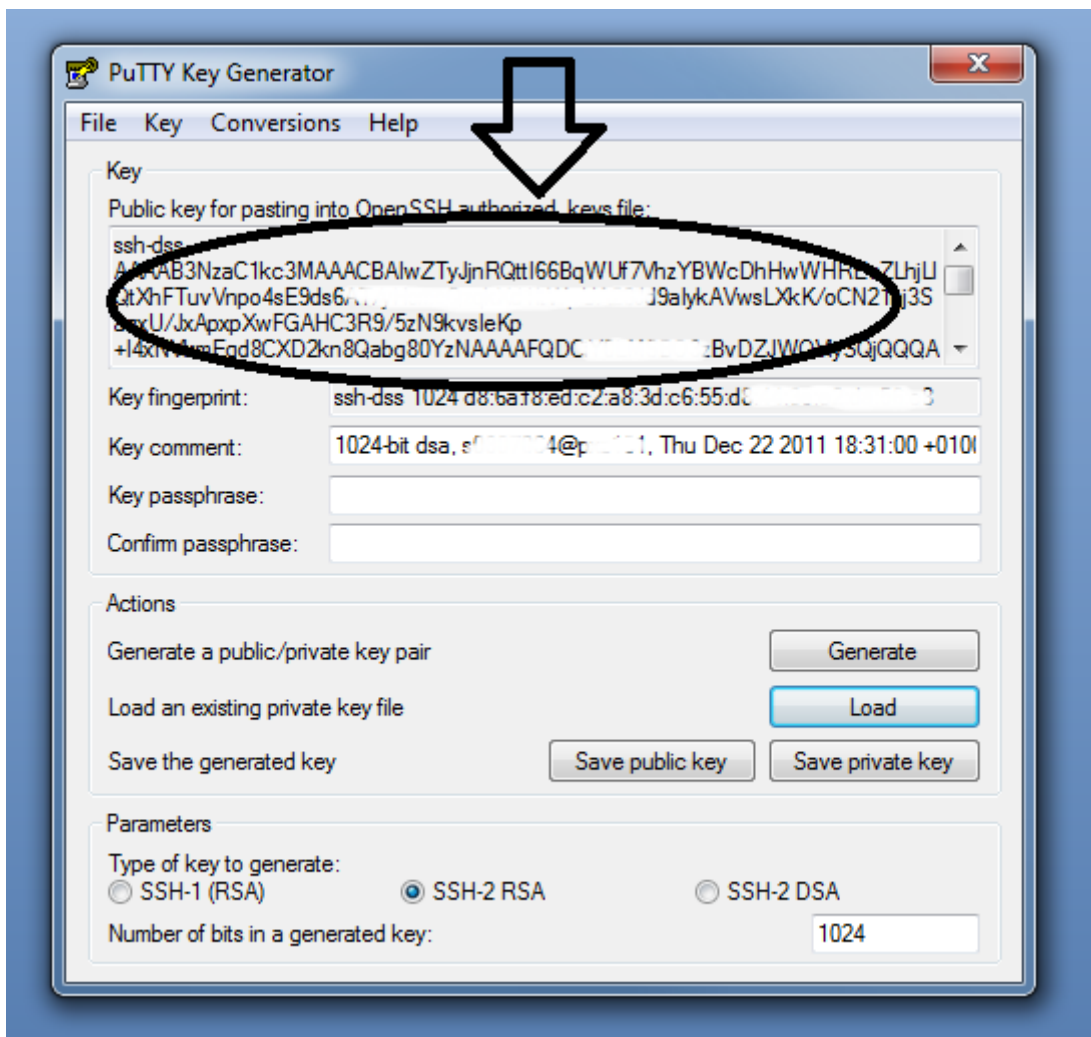


Conexión SFTP-SSH2 en PHP a través de un fichero .ppk

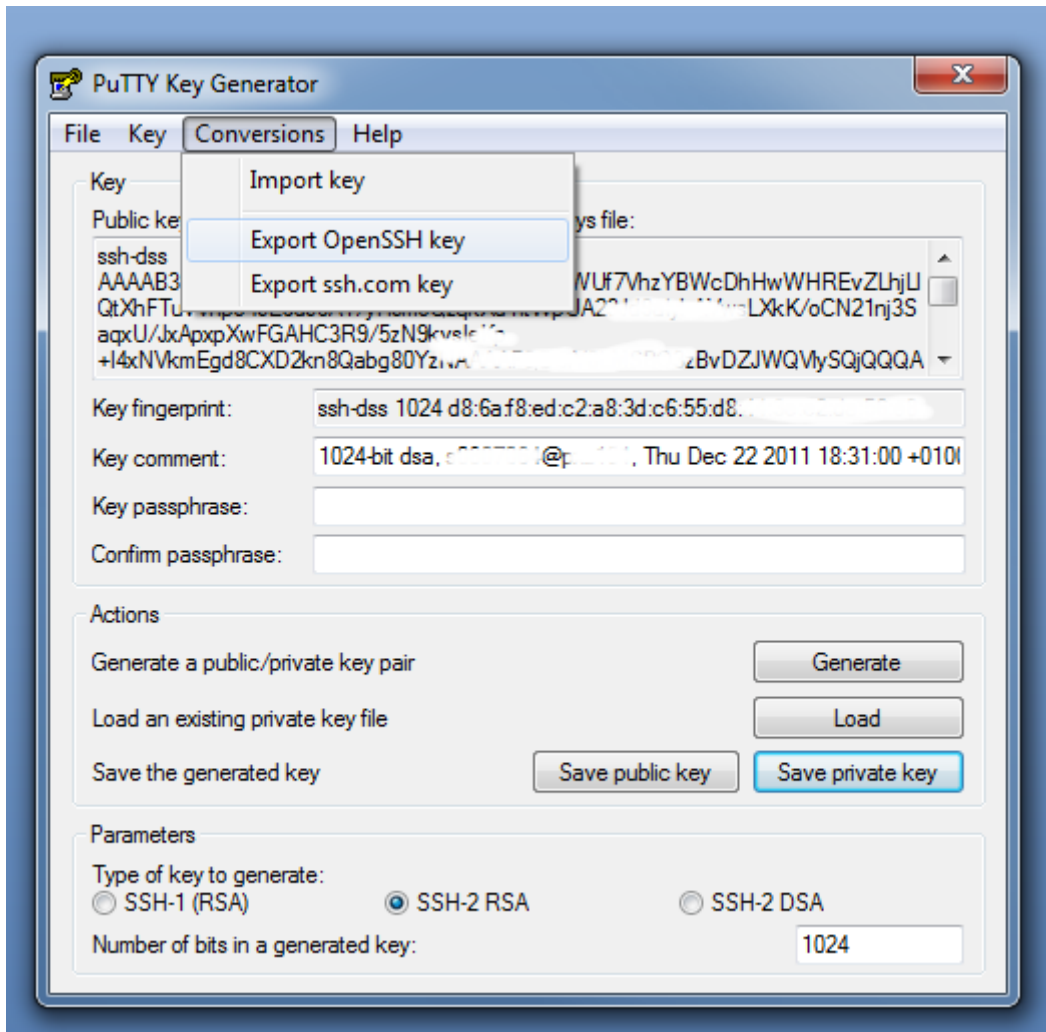
- Necesitamos nuestro fichero .PPK
- Necesitamos crear 2 ficheros de texto plano, uno sin extensión, donde añadiremos una clave privada que tendremos que generar a partir del .PPK y otro fichero que le pondremos extensión .pub, donde añadiremos una clave pública, generada también a partir del .ppk
- Hemos creado los 2 ficheros, por ejemplo con los siguientes nombres:
 - a. id_dsa
 - b. id_dsa.pub
- Para generar los contenidos públicos y privados a través del fichero .ppk necesitamos el programa puttygen.
- (<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>)
- Desde Linux podéis hacerlo de la siguiente manera, donde **fichero.ppk** es el fichero .ppk que nos han entregado:
 - a. `puttygen fichero.ppk -O public-openssh -o id_dsa.pub`
 - b. `puttygen fichero.ppk -O private-openssh -o id_dsa`
- Desde Windows abrimos `puttygen.exe` donde nos aparece la siguiente pantalla:



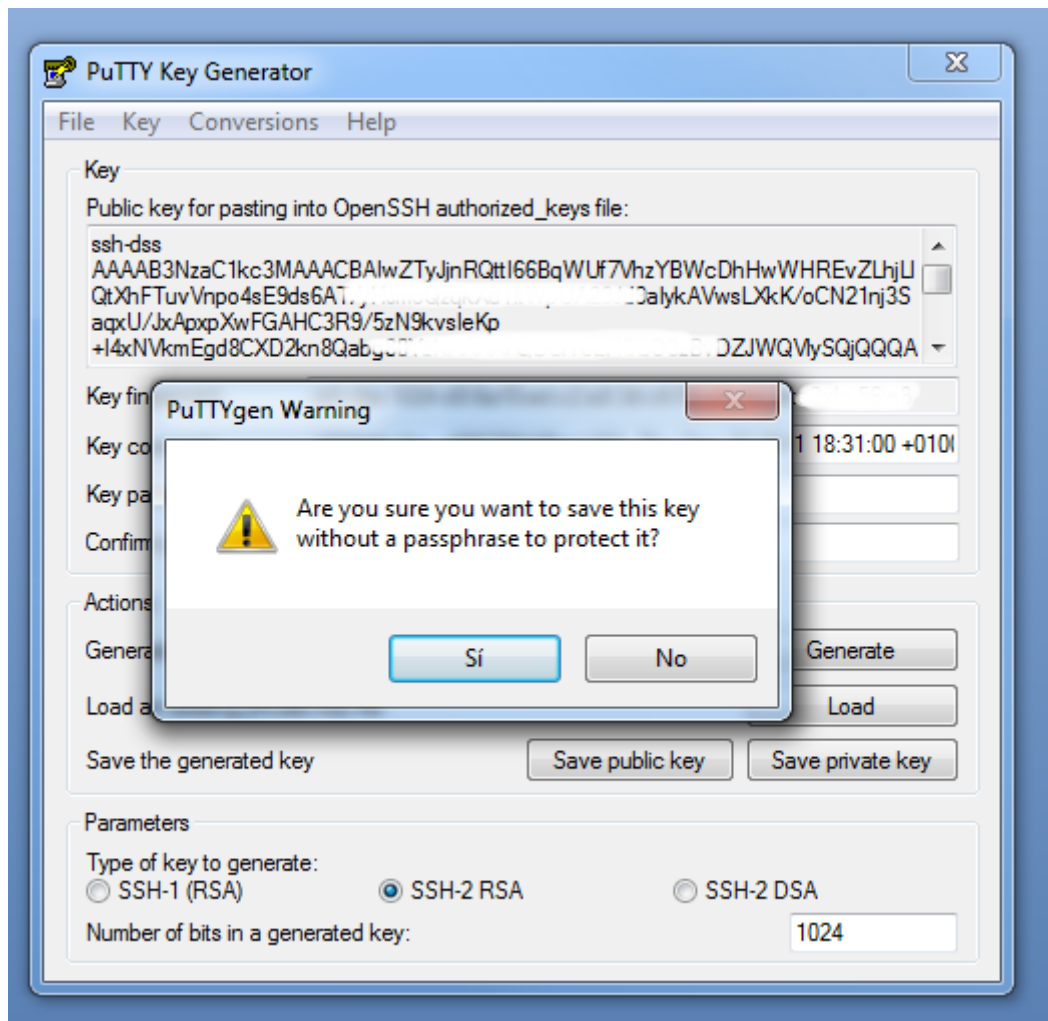
- Clicamos sobre el botón **Load** y seleccionamos nuestro fichero .ppk
- Una vez cargado ya disponemos de datos para añadir nuestra clave pública al fichero id_dsa.pub anteriormente creado. Será la siguiente que nos ha generado, para añadirla:



- Haremos copy/paste del contenido marcado en la imagen anterior y lo pegaremos en el fichero id_dsa.pub. Le damos a guardar y lo cerramos.
- Ahora nos queda rellenar el fichero id_dsa con una clave privada a partir del fichero .ppk. Vamos a ello.
- Iremos a la pestaña llamada Conversions->Export OpenSSH key



- Antes de realizar una exportación, debéis saber si el tipo de Key a generar será RSA o DSA, para ello podéis configurarlo en la zona de **Parameters** de la imagen anterior, justo en la parte del final. Mirando el fichero .ppk se puede ver qué tipo es, justo en la línea de Comment, por ejemplo en nuestro caso tenemos esto en el ppk que nos lo confirma → Comment: 1024-bit dsa, so807xxx@xxx000, Thu Dec 22 2011 18:31:00 +0100 donde vemos que el tipo de key es DSA.
- Clicamos sobre Export, ahora sí, y nos aparece la siguiente página:



- Clicamos sobre el botón “Sí” y guardamos el fichero que genera, con el nombre id_dsa en la misma ubicación que el creado antes con el nombre id_dsa.pub
- Ahora ya tenemos en total 3 ficheros:
 - El fichero .ppk
 - El fichero id_dsa
 - El fichero id_dsa.pub
- Solo nos interesan los 2 últimos.
- Estos 2 ficheros deberemos subirlos al servidor donde queremos realizar la llamada con nuestro fichero PHP, en la ruta siguiente:
 - [/home/pedroesc/.ssh/id_dsa](#)
 - [/home/pedroesc/.ssh/id_dsa.pub](#)
- Donde “pedroesc” será el nombre de usuario de tu path del servidor hasta la carpeta .ssh
- Ahora solo debemos testear con el siguiente código PHP:

```

<?php
// Code from escri2 (http://www.pedroescribano.com)
ini_set('display_errors', true);
error_reporting(E_ALL);

$url = "ifs.pedroescribano.com";
$user = "s08070XX";

// Creando conexión a servidor SSH, puerto 22
$conn = ssh2_connect($url, 22, array('hostkey'=>'ssh-dss'));
if(!$conn){
    throw new Exception("No se pudo conectar");
} else {
    echo "Se ha conectado<br />";
}

if (file_exists('/home/pedroesc/.ssh/id_dsa.pub')) {
    echo "file public OK<br />";
} else {
    echo "file public KO<br />";
}

if (file_exists('/home/pedroesc/.ssh/id_dsa')) {
    echo "file private OK<br />";
} else {
    echo "file private KO<br />";
}

if (ssh2_auth_pubkey_file($conn,
$user, '/home/pedroesc/.ssh/id_dsa.pub', '/home/pedroesc/.ssh/id_dsa')) {
    echo "Public Key Authentication Successful\n";
} else {
    die('Public Key Authentication Failed');
}

// Solicitando subsistema SFTP
$sftp = ssh2_sftp($conn);
echo '<pre>';
print_r($sftp);
echo '</pre>';
// Cargando un archivo al servidor, usando el stream ssh2.sftp
//$upload = copy("archivo.local", "ssh2.sftp://$sftp/archivo.remoto");
?>

```

Donde vamos a necesitar una URL del servidor al que queremos conectar, y un usuario. El password serán los 2 ficheros que hemos generado a partir del .PPK.

Para las funciones de ssh2_connect se requerirá tener instaladas las librerías siguientes si da error al ejecutar el script por ese motivo :

Primero debe de instalarse la extensión PECL/ssh2, la cual requiere las librerías OpenSSL y libssh2. La extensión puede ser instalada con PEAR: pear install ssh2, luego se agrega la extensión a php.ini y se reinicia el servidor web. Para confirmar que la instalación haya sido exitosa se puede revisar que la sección Registered PHP Streams incluya el conjunto de protocolos de SSH2, entre ellos ssh2.sftp:

Thread Safety	enabled
Zend Memory Manager	enabled
IPv6 Support	enabled
Registered PHP Streams	php, file, data, http, ftp, compress.zlib, ssh2.shell, ssh2.exec, ssh2.tunnel, ssh2.scp, ssh2.sftp
Registered Stream Socket Transports	tcp, udp
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags,

Las funciones principales para realizar la conexión son:

- **ssh2_connect**: Establece una conexión hacia un servidor SSH.
- **ssh2_auth_password**: Realiza la autenticación sobre SSH utilizando la dupla usuario/contraseña.
- **ssh2_sftp**: Solicita el subsistema SFTP del servidor SSH al que se está conectado.

Ahora veamos un breve ejemplo de cómo utilizar estas funciones para descargar y cargar archivos a un servidor SFTP con usuario y password, sin el fichero .ppk anteriormente comentado:

```
// Creando conexión a servidor SSH, puerto 22
$conexion = ssh2_connect("miservidor.com", 22);
// Autenticandose en el servidor
ssh2_auth_password($conexion, "usuario", "password");
// Solicitando subsistema SFTP
$sftp = ssh2_sftp($conexion);
// Cargando un archivo al servidor, usando el stream ssh2.sftp
$upload = copy("archivo.local", "ssh2.sftp://$sftp/archivo.remoto");
```

Copiar ficheros remotos de SFTP-SSH2

Una vez conectados a nuestro servidor sftp-ssh2 remoto, habremos llegado a la siguiente línea de código:

- `$sftp = ssh2_sftp($conn);`

La variable `$sftp` será nuestro puntero hacia la ruta raíz de nuestro servidor remoto. Por ejemplo queremos acceder a dos carpetas del servidor remoto, como son la carpeta “entrada” y “salida”, donde a la carpeta entrada subiré ficheros y la carpeta salida me descargaré otros ficheros por ejemplo; así que vamos a preparar cómo continuaría el código:

```
//Solicitando subsistema SFTP
$sftp = ssh2_sftp($conn);

//Nuestra carpeta local de ficheros a enviar al servidor remoto
$path_files = $_SERVER['DOCUMENT_ROOT'] . '/filescorreos/';
//Nuestra carpeta local para descargar ficheros del servidor remoto
$path_filessalida = $_SERVER['DOCUMENT_ROOT'] . '/filescorreossalida/';
//Subcarpeta remota a partir de la raíz donde subiremos nuestros ficheros
$subfolder_entrada = 'entrada';
//Subcarpeta remota a partir de la raíz donde descargaremos ficheros
$subfolder_salida = 'salida';

//Subir fichero local a servidor remoto a la carpeta /entrada/
$upload = copy($path_files . "FD10XX20121220153111.txt",
"ssh2.sftp://$sftp/$subfolder_entrada/FD10CD20121220153111.txt");

//Descargar fichero remoto a servidor local a la carpeta /filescorreossalida/
$upload =
copy("ssh2.sftp://$sftp/$subfolder_salida/FD10XX20121220153111.txt",$path_filessalida . "FD10XX20121220153111.txt",);
```

Listar directorio remoto SFTP-SSH2

Ya conectados al servidor remoto, estamos en la siguiente línea:

- `$sftp = ssh2_sftp($conn);`

Tan sencillo como añadir el siguiente código:

```
//Ficheros listados servidor Correos
echo "<br /><br />RUTA ENTRADA_____<br />";
listado_completo("ssh2.sftp://$sftp/$subfolder_entrada");
```

Donde la función listado_completo queda de la siguiente manera:

```
function listado_completo($ruta) {
    // abrir un directorio y listarlo recursivo
    if (is_dir($ruta)) {
        echo "<ul>";
        if ($dh = opendir($ruta)) {
            while (($file=readdir($dh))) {
                if ($file!="." && $file!="..") {
                    echo "<li>Nombre de archivo: $file";
                    if (is_dir($ruta.$file)) {
                        echo "Directorio: $ruta$file";
                        listado_completo($ruta.$file."/");
                    } else {
                        echo "</li>";
                    }
                }
            }
            closedir($dh);
            echo "</ul>";
        }
    } else echo "<br>No es ruta valida";
}
```

Y si queremos descargar el directorio remoto entero y descomprimir el contenido de éstos en nuestra carpeta llamada filescorreossalida , porque son .zips, ahí tenemos el código:

```
echo "<br /><br />RUTA SALIDA_____<br />";
listado_completo("ssh2.sftp://$sftp/$subfolder_salida");
descarga_listado_completo("ssh2.sftp://$sftp/$subfolder_salida",
$path_filessalida);
```


Donde la función `descarga_listado_completo` queda de la siguiente manera:

```
function descarga_listado_completo($ruta, $path_extract){
    $zip = new ZipArchive;

    if (is_dir($ruta)) {
        if ($dh = opendir($ruta)) {
            while (($file = readdir($dh)) {
                if ($file != "." && $file != "..") {
                    if (is_dir($ruta . '/' . $file)) {

                        } else {
                            copy($ruta . '/' . $file,
                                $path_extract . '/' . $file);

                            if ($zip->open($path_extract . '/' .
                                $file) === TRUE) {
                                    $zip->extractTo($path_extract);
                                    $zip->close();
                                    unlink($path_extract . '/' .
                                $file);
                            } else {
                                echo 'failed ' . $file . '<br
                            />';
                            }
                        }
                    }
                }
            }
            closedir($dh);
        }
    } else echo "<br>No es ruta valida";
}
```