PHP的几种加密解密算法

# 开发环境

1、环境搭建：Windows 7+Apache 2.4.18+MySQL 5.7.11+PHP 7.1.0。

2、文本编辑器：Sublime 3。

# 二、主要技术

本案例主要使用PHP 7中几种加密解密技术：主要有md5、crypt、sha1、base64\_encode(base64\_decode)、urlencode(urldecode)来展示不同加密技术的加密效果。

# 三、实验内容与结果

## 3.1、md5加密技术

md5是一种单向的加密技术，不可逆，不能进行解密。md5() 函数使用 RSA 数据安全，包括 MD5 报文摘要算法。MD5 报文摘要算法将任意长度的信息作为输入值，并将其换算成一个 128 位长度的“指纹信息”或“报文摘要”值来代表这个输入值，并以换算后的值作为结果。

### （1）实现过程

<?php

/\*\*

\*md5加密算法

\*/

echo "<form action='' method='post'>";

echo "<input type='text' name='text'>";

echo "<input type='submit' name='sub' value='md5加密'>";

echo "</form>";

// md5加密

if(isset($\_POST['sub'])){

$str=$\_POST['text'];

$md5str=md5($str);

echo "加密结果：".$md5str;

}

?>

### （2）结果测试

比如在文本框内输入“123$#%d&3e\*wsc”，点击【md5加密】按钮，可以看到程序运行结果如图1所示。

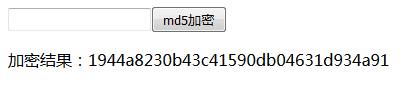


图1 案例结果图（md5加密结果测试）

## 3.2、crypt加密技术

crypt是一种单向的加密技术，不可逆，不能进行解密。具体的算法依赖于 salt 参数的格式和长度。通过增加由使用特定加密方法的特定字符串所生成的字符串数量，salt 可以使加密更安全。

### （1）实现过程

<?php

/\*\*

\*crypt加密算法

\*其单向加密，不可逆，不可解密

\*/

echo "<form action='' method='post'>";

echo "<input type='text' name='text'>";

echo "<input type='submit' name='sub' value='crypt

加密'>";

echo "</form>";

// crypt加密

if(isset($\_POST['sub'])){

$str=$\_POST['text'];

$cryptstr=crypt($str,'r2'); // 'r2'为盐值，

默认是随机生成的两位字串

echo "加密结果：".$cryptstr;

}

?>

### （2）结果测试

比如在文本框内输入“123$#%d&3e\*wsc”，点击【crypt加密】按钮，可以看到程序运行结果如图2所示。

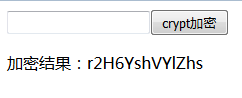


图2 案例结果图（cryt加密结果测试）

## 3.3、sha1加密技术

Sha1是一种单向的加密技术，不可逆，不能进行解密。

### （1）实现过程

<?php

/\*\*

\*sha1加密算法

\*其单向加密，不可逆，不可解密

\*/

echo "<form action='' method='post'>";

echo "<input type='text' name='text'>";

echo "<input type='submit' name='sub' value='sha1加

密'>";

echo "</form>";

// sha1加密

if(isset($\_POST['sub'])){

$str=$\_POST['text'];

$sha1str=sha1($str);

echo "加密结果：".$sha1str;

}

?>

### （2）结果测试

比如在文本框内输入“123$#%d&3e\*wsc”，点击【sha1加密】按钮，可以看到程序运行结果如图3所示。

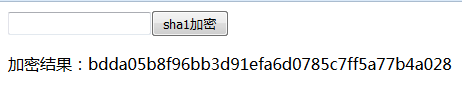


图3 案例结果图（Sha1加密结果测试）

## 3.4、base64编码加密、解密技术

设计此种编码是为了使二进制数据可以通过非纯 8-bit 的传输层传输，例如电子邮件的主体。

### （1）实现过程

<?php

/\*\*

\*Base64编码加密技术

\*base64\_encode — 使用 MIME base64 对数据进行编码

\*string base64\_encode ( string $data ) ,使用 base64

\*对 data 进行编码。设计此种编码是为了使二进制数据可以通过非

\*纯 8-bit \*的传输层传输，例如电子邮件的主体。

\*/

echo "<form action='' method='post'>";

echo "<input type='text' name='text'>";

echo "<input type='submit' name='sub' value='加

密'>";

echo "</form>";

// 加密

if(isset($\_POST['sub'])){

$str=$\_POST['text'];

$enstr=base64\_encode($str);

echo "加密结果：".$enstr."<br/>";

@session\_start();

$\_SESSION['entsr']=$enstr;

echo "<a href='base64.php?pwd=1'>解密</a>";

}

// 解密

if($pwd=isset($\_GET['pwd'])){

@session\_start();

$str=@$\_SESSION['entsr'];

$destr=base64\_decode($str);

echo "解密结果：".$destr."<br/>";

}

?>

### （2）结果测试

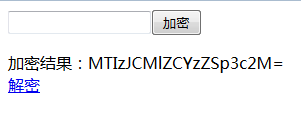
①比如在文本框内输入“123$#%d&3e\*wsc”， 点击【加密】按钮，可以看到程序运行结果如图4所示。

图4 案例结果图（Base64\_encode加密结果测试）

②然后点击【解密】按钮，可以看到程序运行结果如图5所示。

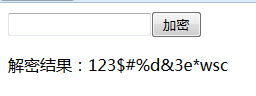


图5 案例结果图（Base64\_decode解密结果测试）

## 3.5、url编码加密、解密技术

url编码加密技术使用urlencode/urldecode实现加密解密,urlencode()返回字符串，此字符串中除了 -\_. 之外的所有非字母数字字符都将被替换成百分号（%）后跟两位十六进制数，空格则编码为加号（+）。注意：只对非字母数字进行处理。

### （1）实现过程

<?php

/\*\*

\*url编码加密技术

\*/

echo "<form action='' method='post'>";

echo "<input type='text' name='text'>";

echo "<input type='submit' name='sub' value='加

密'>";

echo "</form>";

// 加密

if(isset($\_POST['sub'])){

$str=$\_POST['text'];

$enstr=urlencode($str);

echo "加密结果：".$enstr."<br/>";

@session\_start();

$\_SESSION['entsr']=$enstr;

echo "<a href='urlcode.php?pwd=1'>解

密</a>";

}

// 解密

if($pwd=isset($\_GET['pwd'])==1){

@session\_start();

$str=@$\_SESSION['entsr'];

$destr=urldecode($str);

echo "解密结果：".$destr."<br/>";

}

?>

### （2）结果测试

①比如在文本框内输入“123$#%d&3e\*wsc”， 点击【加密】按钮，可以看到程序运行结果如图6所示。

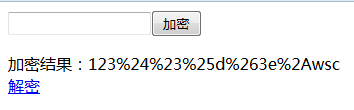


图6 案例结果图（urlcode加密结果测试）

②然后点击【解密】按钮，可以看到程序运行结果如图7所示。

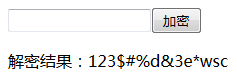


图7 案例结果图（Urldecode解密结果测试）

## 3.6、自定义数字加密、解密算法

此种数字加密、解密算法主要使用位运算符对数字进行加密、解密。

### （1）实现过程

<?php

/\*\*

\*自定义数字加密算法

\*/

echo "<form action='' method='post'>";

echo "数字口令：<input type='text' name='text'>";

echo "<input type='submit' name='sub' value='确

定'>";

echo "</form>";

define("PI",3.1415926);

// 自定义加密算法

function Encrypt($str){

return $str=$str<<PI;

}

// 自定义解密算法

function Decrypt($str){

return $str=$str>>PI;

}

if(isset($\_POST['sub'])){

echo "加密口令

&nbsp;&nbsp;".Encrypt($\_POST['text']).

"</br>"; // 输出口令

@session\_start();

$\_SESSION['pwd']=Encrypt($\_POST['text']);

echo "<a href='zdy.php?pwd=1'>解密口令</a>";

}

@session\_start();

if(isset($\_GET['pwd'])){

echo "解密口令

&nbsp;&nbsp;".Decrypt($\_SESSION['pwd']);

}

?>

### （2）结果测试

①比如在文本框内输入“123456789”， 点击【确定】按钮，可以看到程序运行结果如图8所示。

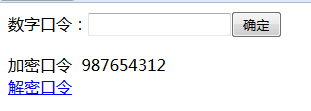


图8 案例结果图（自定义算法加密结果测试）

②然后点击“解密口令”超链接，可以看到程序运行结果如图9所示。

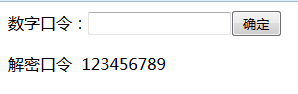


图9 案例结果图（自定义算法解密结果测试）