

# 如何快速生成数字类型的兑换码，又能防止别人盲猜呢？没错带有验证真伪性的数字，比如身份证真伪验证

作者: [ieras](#)

原文链接: <https://ld246.com/article/1608124571070>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



工作中，经常会遇到兑换码功能的需求开发，这时候就会有各种问题

- 字母数字组合不好输入
- 纯数字兑换码容易被盲猜
- 其他问题

今天我们主要介绍一种带有真伪验证的数字兑换码生成方案。

## 真伪检测code创建代码

```
/**
 * 创建带有真伪性验证的数字
 * @param int $code 原始数字
 * @param int $int_key 加密key(必须大于0, 小于$code)
 * @return int 结果数字比原数字长2位
 */
function create_code($code,$int_key){
    $num = 0;$m = '5194268037';$n = '3061758429';
    foreach (range(0,strlen($code)-1) as $k=>$v){
        $num = $num + $code[$k];
    }
    $num = $num % intval($code/$int_key);
    $num = substr(sprintf('%02s', $num),-2);
    return $code.$m[$num[0]].$n[$num[1]];
}
```

## 真伪检测code检测代码

```
/**
```

```

* 验证带有真伪性验证的数字的真伪性
* @param int $code 原始数字
* @param int $int_key 加密key(必须大于0, 小于$code)
* @return boolean
*/
function check_code($code,$int_key){
    $tmp_code = substr($code,0,14);
    $num = 0;$m = '5194268037';$n = '3061758429';
    foreach (range(0,strlen($tmp_code)-1) as $k=>$v){
        $num = $num + $tmp_code[$k];
    }
    $num = $num % intval($tmp_code/$int_key);
    $num = substr(sprintf('%02s', $num),-2);
    $check_code = $tmp_code.$m[$num[0]].$n[$num[1]];
    if($check_code == $code){
        return true;
    }else{
        return false;
    }
}

```

## 测试使用

```

$str = '51650112989957';
//var_dump($str[0],$str[1]);
echo '<pre>'.$str.'<br />';
$int_key = 125637812;
$new_str = create_code($str,$int_key);
var_dump($new_str);
$check = check_code($new_str,$int_key);
var_dump($check);

```

## 一些说明

- 创建出来的code要比原来的数字多2位
- init\_key是会被除被取余的，所以不能是0，也不能大于code，因为取余后还要被除
- 方法里面的 m n其实就是0~9的排序，这个自己可以随意定义，目的就是让用户不好猜测，避免用户能简单算到的情况发生。

由于打到车了，今天的分享就草草结束吧！

其实算法类似身份证真伪验证那样，可以自己随意书写里面的真伪算法哦

再见！