1、客户端请求参数+生成的加密串

a）客户端请求url：xxx?a=1&b=2&sign=xxxxxxxx

b）验证规则：客户端和服务段拥有相同的密钥secret\_key，客户端在请求时请求参数+密钥生成加密串，一起发送给服务器，服务器根据获取到的参数进行加密串对比，相同返回数据

c）缺点：请求参数暴露在网络中，不安全，请求url可以永久请求，没有失效时间

2、请求参数+时间戳+加密串+（其他自定义限制）

a）客户端请求url：xxx?a=1&b=2&timestamp= 1557297225&sign=xxxxxxxx

b）验证规则：客户端在请求时请求参数+密钥+时间参数生成加密串，一起发送给服务器，服务端进行加密串对比和请求时间对比

c）缺点：请求参数暴露在网络中

d）优点：请求可控，每隔一段时间必须重新生成url，起到一定的安全性

3、异或加密

a）客户端请求url：xxx? sign=xxxxxxxx&其他参数

b）异或加密传递，解密对比数据

c）有点：传输中看不到数据

public static function xor\_encrypt\_bit(*$value*,*$encryt\_string*=*EncryString*){  
 for(*$i*=0,*$v*="";*$i*<*strlen*(*$value*);*$i*++){  
 *$v*.=*$value*{*$i*} ^ *$encryt\_string*{*$i*%*strlen*(*$encryt\_string*)};  
 }  
 return *$v*;  
}

对于url进行编码的

function base64url\_encode($data) {   
  return rtrim(strtr(base64\_encode($data), '+/', '-\_'), '=');   
}   
function base64url\_decode($data) {   
  return base64\_decode(str\_pad(strtr($data, '-\_', '+/'), strlen($data) % 4, '=', STR\_PAD\_RIGHT));   
}

例：

*$str* = "game\_name=作妖计&price\_order=-1";  
*$ency* = "yutjfghdsts545d1fg5ds21g5d1fdf12fg1df531a";  
  
for (*$i* = 0 ,*$v* = ''; *$i* < *strlen*(*$str*);*$i*++){  
 *$v* .= *$str*[*$i*] ^ *$ency*[*$i*%*strlen*(*$ency*)];  
}  
echo *$v*.'<br>';  
*$v* = base64url\_encode(*$v*)."<br>";  
*$v* = base64url\_decode(*$v*);  
for (*$i* = 0 ,*$v2* = ''; *$i* < *strlen*(*$v*);*$i*++){  
 *$v2* .= *$v*[*$i*] ^ *$ency*[*$i*%*strlen*(*$ency*)];  
}  
echo *$v2*.'<br>';  
function base64url\_encode(*$data*) {  
 return *rtrim*(*strtr*(*base64\_encode*(*$data*), '+/', '-\_'), '=');  
}  
  
function base64url\_decode(*$data*) {  
 return *base64\_decode*(*str\_pad*(*strtr*(*$data*, '-\_', '+/'), *strlen*(*$data*) % 4, '=', *STR\_PAD\_RIGHT*));  
 return *base64\_decode*(*str\_pad*(*strtr*(*$data*, '-\_', '+/'), (*strlen*(*$data*) % 4)+*strlen*(*$data*), '=', *STR\_PAD\_RIGHT*));  
}