(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 113761549 A
(43)申请公布日 2021.12.07

(21)申请号 202011221342.8

(22)申请日 2020.11.04

(71)申请人 北京沃东天骏信息技术有限公司
地址 100176 北京市北京经济技术开发区科创十一街18号院2号楼4层A402室
申请人 北京京东世纪贸易有限公司

(72)发明人 赵晓宇 韩成利

(74)专利代理机构 中原信达知识产权代理有限责任公司 11219
代理人 郝红玉 冯培培

(51)Int.Cl.
G06F 21/60(2013.01)
G06F 21/64(2013.01)

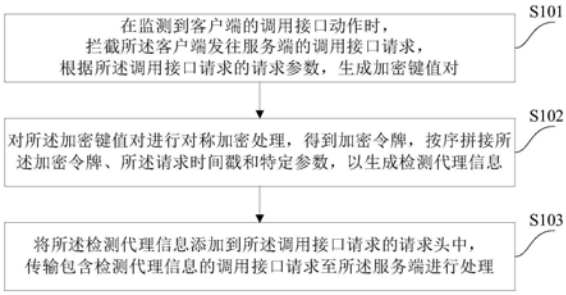
权利要求书2页 说明书12页 附图4页

(54)发明名称

一种接口安全控制、校验方法和装置

(57)摘要

本发明公开了一种接口安全控制、校验方法和装置，涉及计算机技术领域。该方法的一具体实施方式包括：在监测到客户端的调用接口动作时，拦截客户端发往服务端的调用接口请求，根据调用接口请求的请求参数，生成加密键值对；对加密键值对进行对称加密处理，得到加密令牌，按序拼接加密令牌、请求时间戳和特定参数，以生成检测代理信息；将检测代理信息添加到调用接口请求的请求头中，传输包含检测代理信息的调用接口请求至服务端进行处理。该实施方式通过在前后端约定一个加密解密的规则，设置http/https请求头中的M-Agent字段来进行校验，以实现接口安全监控。



CN 113761549 A

1.一种接口安全控制方法,其特征在于,包括:

在监测到客户端的调用接口动作时,拦截所述客户端发往服务端的调用接口请求,根据所述调用接口请求的请求参数,生成加密键值对;其中,所述请求参数包括请求时间戳;

对所述加密键值对进行对称加密处理,得到加密令牌,按序拼接所述加密令牌、所述请求时间戳和特定参数,以生成检测代理信息;

将所述检测代理信息添加到所述调用接口请求的请求头中,传输包含检测代理信息的调用接口请求至所述服务端进行处理。

2.根据权利要求1所述的方法,其特征在于,所述请求参数还包括接口名、参数字符串集和平台类型;

所述根据所述调用接口请求的请求参数,生成加密键值对,包括:

基于预设加密规则处理所述请求时间戳,得到加密键名;

按序拼接所述接口名、所述请求时间戳、所述参数字符串集和所述平台类型,采用信息摘要算法处理拼接后的信息,得到加密键值。

3.根据权利要求2所述的方法,其特征在于,所述基于预设加密规则处理所述请求时间戳,得到加密键名,包括:

将所述请求时间戳转制为二进制形式,截取预设数量的二进制数,对截取的二进制数进行取反并转制为十进制形式,得到加密键名。

4.一种接口安全校验方法,其特征在于,包括:

服务端接收客户端传输的调用接口请求,从所述调用接口请求的请求头中抽取出检测代理信息和请求参数;

根据所述请求参数生成第一加密键值对,以及对所述检测代理信息进行拆解,得到加密令牌并解密,得到第二加密键值对;

比对所述第一加密键值对和所述第二加密键值对是否相同,若比对结果相同,则触发对所述调用接口请求的处理,返回响应结果至所述客户端,否则返回令牌校验错误信息至所述客户端。

5.根据权利要求4所述的方法,其特征在于,所述请求参数包括第一请求时间戳、第一接口名、第一参数字符串集和第一平台类型;

所述根据所述请求参数生成第一加密键值对,包括:

基于预设加密规则处理所述第一请求时间戳,得到第一加密键名;

按序拼接所述第一接口名、所述第一请求时间戳、所述第一参数字符串集和所述第一平台类型,采用信息摘要算法处理拼接后的信息,得到第一加密键值。

6.根据权利要求5所述的方法,其特征在于,所述基于预设加密规则处理所述第一请求时间戳,得到第一加密键名,包括:

将所述第一请求时间戳转制为二进制形式,截取预设数量的二进制数,对截取的二进制数进行取反并转制为十进制形式,得到第一加密键名。

7.一种接口安全控制装置,其特征在于,包括:

监测模块,用于在监测到客户端的调用接口动作时,拦截所述客户端发往服务端的调用接口请求,根据所述调用接口请求的请求参数,生成加密键值对;其中,所述请求参数包括请求时间戳;

加密模块，用于对所述加密键值对进行对称加密处理，得到加密令牌，按序拼接所述加密令牌、所述请求时间戳和特定参数，以生成检测代理信息；

传输模块，用于将所述检测代理信息添加到所述调用接口请求的请求头中，传输包含检测代理信息的调用接口请求至所述服务端进行处理。

8.一种接口安全校验装置，其特征在于，包括：

抽取模块，用于服务端接收客户端传输的调用接口请求，从所述调用接口请求的请求头中抽取出检测代理信息和请求参数；

解密模块，用于根据所述请求参数生成第一加密键值对，以及对所述检测代理信息进行拆解，得到加密令牌并解密，得到第二加密键值对；

处理模块，用于比对所述第一加密键值对和所述第二加密键值对是否相同，若比对结果相同，则触发对所述调用接口请求的处理，返回响应结果至所述客户端，否则返回令牌校验错误信息至所述客户端。

9.一种电子设备，其特征在于，包括：

一个或多个处理器；

存储装置，用于存储一个或多个程序，

当所述一个或多个程序被所述一个或多个处理器执行，使得所述一个或多个处理器实现如权利要求1-6中任一所述的方法。

10.一种计算机可读介质，其上存储有计算机程序，其特征在于，所述程序被处理器执行时实现如权利要求1-6中任一所述的方法。

一种接口安全控制、校验方法和装置

技术领域

[0001] 本发明涉及计算机技术领域，尤其涉及一种接口安全控制、校验方法和装置。

背景技术

[0002] 随着电商网络的发展，网络购物越来越普及，电商平台通常会发放一些豆、大额优惠券或者红包来吸引用户流量，这些利益有时候会引来一些别有用心的人的觊觎，他们通过仿造接口请求在短时间内大规模调用接口，将这些豆、券或者是红包刷走，以谋取暴利。

[0003] 目前主要采用限制单IP的最大发送量、图形验证码、短信验证码、设备指纹判断以及token方式监控接口安全，但在实现本发明的过程中，发明人发现现有方式实行的安全性不高，已有现成的破解方案；兼容性不高，无法同时适用于小程序、pc、移动端。

发明内容

[0004] 有鉴于此，本发明实施例提供一种接口安全控制、校验方法和装置，至少能够解决现有技术安全性低、兼容性低和不具备普适性的现象。

[0005] 为实现上述目的，根据本发明实施例的一个方面，提供了一种接口安全控制方法，包括：

[0006] 在监测到客户端的调用接口动作时，拦截所述客户端发往服务端的调用接口请求，根据所述调用接口请求的请求参数，生成加密键值对；其中，所述请求参数包括请求时间戳；

[0007] 对所述加密键值对进行对称加密处理，得到加密令牌，按序拼接所述加密令牌、所述请求时间戳和特定参数，以生成检测代理信息；

[0008] 将所述检测代理信息添加到所述调用接口请求的请求头中，传输包含检测代理信息的调用接口请求至所述服务端进行处理。

[0009] 可选的，所述请求参数还包括接口名、参数字符串集和平台类型；

[0010] 所述根据所述调用接口请求的请求参数，生成加密键值对，包括：

[0011] 基于预设加密规则处理所述请求时间戳，得到加密键名；

[0012] 按序拼接所述接口名、所述请求时间戳、所述参数字符串集和所述平台类型，采用信息摘要算法处理拼接后的信息，得到加密键值。

[0013] 可选的，所述基于预设加密规则处理所述请求时间戳，得到加密键名，包括：

[0014] 将所述请求时间戳转制为二进制形式，截取预设数量的二进制数，对截取的二进制数进行取反并转制为十进制形式，得到加密键名。

[0015] 为实现上述目的，根据本发明实施例的一个方面，提供了一种接口安全校验方法，包括：

[0016] 服务端接收客户端传输的调用接口请求，从所述调用接口请求的请求头中抽取出检测代理信息和请求参数；

[0017] 根据所述请求参数生成第一加密键值对，以及对所述检测代理信息进行拆解，得

到加密令牌并解密，得到第二加密键值对；

[0018] 比对所述第一加密键值对和所述第二加密键值对是否相同，若比对结果相同，则触发对所述调用接口请求的处理，返回响应结果至所述客户端，否则返回令牌校验错误信息至所述客户端。

[0019] 可选的，所述请求参数包括第一请求时间戳、第一接口名、第一参数字符串集和第一平台类型；

[0020] 所述根据所述请求参数生成第一加密键值对，包括：

[0021] 基于预设加密规则处理所述第一请求时间戳，得到第一加密键名；

[0022] 按序拼接所述第一接口名、所述第一请求时间戳、所述第一参数字符串集和所述第一平台类型，采用信息摘要算法处理拼接后的信息，得到第一加密键值。

[0023] 可选的，所述基于预设加密规则处理所述第一请求时间戳，得到第一加密键名，包括：

[0024] 将所述第一请求时间戳转制为二进制形式，截取预设数量的二进制数，对截取的二进制数进行取反并转制为十进制形式，得到第一加密键名。

[0025] 为实现上述目的，根据本发明实施例的另一方面，提供了一种接口安全控制装置，包括：

[0026] 监测模块，用于在监测到客户端的调用接口动作时，拦截所述客户端发往服务端的调用接口请求，根据所述调用接口请求的请求参数，生成加密键值对；其中，所述请求参数包括请求时间戳；

[0027] 加密模块，用于对所述加密键值对进行对称加密处理，得到加密令牌，按序拼接所述加密令牌、所述请求时间戳和特定参数，以生成检测代理信息；

[0028] 传输模块，用于将所述检测代理信息添加到所述调用接口请求的请求头中，传输包含检测代理信息的调用接口请求至所述服务端进行处理。

[0029] 可选的，所述请求参数还包括接口名、参数字符串集和平台类型；

[0030] 所述监测模块，用于：

[0031] 基于预设加密规则处理所述请求时间戳，得到加密键名；

[0032] 按序拼接所述接口名、所述请求时间戳、所述参数字符串集和所述平台类型，采用信息摘要算法处理拼接后的信息，得到加密键值。

[0033] 可选的，所述监测模块，用于：将所述请求时间戳转制为二进制形式，截取预设数量的二进制数，对截取的二进制数进行取反并转制为十进制形式，得到加密键名。

[0034] 为实现上述目的，根据本发明实施例的另一方面，提供了一种接口安全校验装置，包括：

[0035] 抽取模块，用于服务端接收客户端传输的调用接口请求，从所述调用接口请求的请求头中抽取出检测代理信息和请求参数；

[0036] 解密模块，用于根据所述请求参数生成第一加密键值对，以及对所述检测代理信息进行拆解，得到加密令牌并解密，得到第二加密键值对；

[0037] 处理模块，用于比对所述第一加密键值对和所述第二加密键值对是否相同，若比对结果相同，则触发对所述调用接口请求的处理，返回响应结果至所述客户端，否则返回令牌校验错误信息至所述客户端。

[0038] 可选的，所述请求参数包括第一请求时间戳、第一接口名、第一参数字符串集和第一平台类型；

[0039] 所述解密模块，用于：基于预设加密规则处理所述第一请求时间戳，得到第一加密键名；

[0040] 按序拼接所述第一接口名、所述第一请求时间戳、所述第一参数字符串集和所述第一平台类型，采用信息摘要算法处理拼接后的信息，得到第一加密键值。

[0041] 可选的，所述解密模块，用于：将所述第一请求时间戳转制为二进制形式，截取预设数量的二进制数，对截取的二进制数进行取反并转制为十进制形式，得到第一加密键名。

[0042] 为实现上述目的，根据本发明实施例的再一方面，提供了一种接口安全控制、校验电子设备。

[0043] 本发明实施例的电子设备包括：一个或多个处理器；存储装置，用于存储一个或多个程序，当所述一个或多个程序被所述一个或多个处理器执行，使得所述一个或多个处理器实现上述任一所述的接口安全控制、校验方法。

[0044] 为实现上述目的，根据本发明实施例的再一方面，提供了一种计算机可读介质，其上存储有计算机程序，所述程序被处理器执行时实现上述任一所述的接口安全控制、校验方法。

[0045] 根据本发明所述提供的方案，上述发明中的一个实施例具有如下优点或有益效果：前后端通过采用一定的规则，对调用接口请求中的时间戳等请求参数进行加密处理，以约定一个M-Agent字段并放到http/https请求头中，服务端收到后判断解密得到的信息与客户端根据参数生成的信息是否一致，以此实现接口安全监控。

[0046] 上述的非惯用的可选方式所具有的进一步效果将在下文中结合具体实施方式加以说明。

附图说明

[0047] 附图用于更好地理解本发明，不构成对本发明的不当限定。其中：

[0048] 图1是根据本发明实施例的一种接口安全控制方法的主要流程示意图；

[0049] 图2是根据本发明实施例的一种接口安全校验方法的主要流程示意图；

[0050] 图3是根据本发明实施例的一种接口安全控制装置的主要模块示意图；

[0051] 图4是根据本发明实施例的一种接口安全校验装置的主要模块示意图；

[0052] 图5是本发明实施例可以应用于其中的示例性系统架构图；

[0053] 图6是适于用来实现本发明实施例的移动设备或服务器的计算机系统的结构示意图。

具体实施方式

[0054] 以下结合附图对本发明的示范性实施例做出说明，其中包括本发明实施例的各种细节以助于理解，应当将它们认为仅仅是示范性的。因此，本领域普通技术人员应当认识到，可以对这里描述的实施例做出各种改变和修改，而不会背离本发明的范围和精神。同样，为了清楚和简明，以下的描述中省略了对公知功能和结构的描述。

[0055] 目前主要采用限制单IP的最大发送量、图形验证码、短信验证码、设备指纹判断以

及token方式监控接口安全,以下对各方式及其缺点进行说明:

[0056] 1、限制单IP的最大发送量,可以防止单一IP下多手机号被刷的问题,但对大区域网络公共IP不友好,且灰色产业能轻易绕过。

[0057] 2、图形验证码,可以机器学习自动识别,还能接入打码平台来验证是否是真实用户以达到拦截的目的。但是存在一些第三方的验证码打码平台,该平台可以自动识别出图形验证码的答案,以绕过这道拦截,所以拦截力度不大。

[0058] 3、短信验证码,也有接码平台,而且容易被不法分子进行短信轰炸,恶意刷掉大量短信费用,给公司或个人造成大量的金钱损失。

[0059] 4、设备指纹识别方式,通过对每个设备(手机/PC)生成唯一、不可伪造的设备ID。但不同生态的平台对用户隐私数据的开放权限不同,难以生成唯一识别码,且无法实现Web和App的跨域统一。主动式设备指纹的另一个局限性,由于强依赖客户端代码,导致生成的指纹在反欺诈的场景中对抗性较弱。

[0060] 5、token,对于重要的API接口,生成token值做验证。原理是用户登录后会向服务器提供用户认证信息(如账户和密码),服务器认证完后返回给客户端一个token,用户再次访问站点其他接口获取信息时,需要带上此令牌。服务端接收到请求后进行token验证,如果token不存在或过期,说明请求无效,并拒绝服务。但该种方式要求用户必须是登录状态,而有些接口,比如小程序里根据用户的openid来助力的情况,实际上是没有登录的。

[0061] 参见图1,示出的是本发明实施例提供的一种接口安全控制方法的主要流程图,包括如下步骤:

[0062] S101:在监测到客户端的调用接口动作时,拦截所述客户端发往服务端的调用接口请求,根据所述调用接口请求的请求参数,生成加密键值对;其中,所述请求参数包括请求时间戳;

[0063] S102:对所述加密键值对进行对称加密处理,得到加密令牌,按序拼接所述加密令牌、所述请求时间戳和特定参数,以生成检测代理信息;

[0064] S103:将所述检测代理信息添加到所述调用接口请求的请求头中,传输包含检测代理信息的调用接口请求至所述服务端进行处理。

[0065] 上述实施方式中,对于步骤S101,在监测到客户端主动调用某一接口的动作时,拦截其向服务端发出的调用接口请求并进行处理,以实现接口安全加密。具体地:

[0066] 1、根据调用接口请求的请求时间戳,生成加密key。加密规则:将请求时间戳转制为二进制数,截取预定数量的二进制数(如从后往前或从前往后截取31位),对截取出的二进制数取反并转制为十进制,得到十进制形式的加密key。实际操作中,也可以根据js或者java生成的随机数,然后通过某种加密规则生成加密key。

[0067] 由于后续对称加密算法要求加密key的密钥长度为16位、24位、32位,所以此处加密key的长度最小为16位,不足末尾补1。

[0068] 整体执行代码如下:

```
* getBinaryKey 获取密钥的方法
* @param time - 时间戳
* @returns string
*/
  function getBinaryKey(time){
```

[0069]

```
    let key=time.toString(2)//time 换算成二进制取 31 位并取反，结果保持 16 位，不够的话末尾补 1
    let length = key.length;
    key = key.substr(0, length)
    key = ~(parseInt(key,2))+''
    if (key.length < 16) { // 不足 16 位末尾补 1
    key = key + '1'.repeat(16 - key.length)
```

[0070]

```
    }
    return key;
  }
```

[0071] 2、从调用接口请求中抽取接口名、参数字符串集、平台类型，结合请求时间戳，使用字符“-”(仅为示例，还可以是其他字符)对其进行按序拼接，经过MD5加密以生成加密value，具体形式如加密value＝MD5(接口名-请求时间戳-参数字符串集-平台类型)。需要说明的是，对加密value中各参数的数量不做限制，如请求时间戳可以被拼接两次。

[0072] 参数说明如下：

[0073] ①接口名：接口请求名称，如https://wxapp.m.xx.com/shopwechat/getShopData，最后一个“/”之后的getShopData即为接口请求名称；

[0074] ②请求时间：接口请求时间戳，根据javascript的内置函数new Date生成，生成方式为timestamp＝new Date().getTime()。时间戳是指格林威治时间1970年01月01日00时00分00秒(北京时间1970年01月01日08时00分00秒)起至现在的总秒数，是一份能够表示一份数据在一个特定时间点已经存在的完整的可验证的数据。

[0075] ③参数字符串集：所有get和post方式的参数(get是指直接跟在请求地址后边的参数，post参数是指放在body里传的参数)，参数以key1＝value1&key2＝value2的形式进行拼接，拼接起来的字符串按照字典顺序进行排列。比如{"b":1,"a":2}格式化成a＝2&b＝1参与加密。

[0076] ④平台类型：微信小程序是wx、pc端是web、h5则是h5，这个字段是不同平台在调用接口时可以自行修改的。比如运行小程序，字段上传wx、Pc端传web、h5同理。运行在哪个平台是开发的时候就预先设定的，所以可以在代码里可以直接固定完毕。

[0077] 整体执行代码如下：

将 params 按 key 的字典排序组成类似 key1=value1&key2=value2... 的字符串

```
 * @param {object} params - 参数
 * @returns json 字符串
 */
function orderJsonParam(param){
```

[0078]

```
  let unordered=param;
  let ordered={}
  Object.keys(unordered).sort().forEach(function(key) {
        ordered[key] = unordered[key];
  });
  return jsonSerialize(ordered)
}
```

[0079] 对于步骤S102和S103,在得到加密键值对后,采用对称加密算法对其进行再加密处理,如AES(Advanced Encryption Standard,高级加密标准)、DES(Data Encryption Standard,数据加密标准)、TripleDES、IDEA(Triple Data Encryption Algorithm,三重数据加密算法)、PBE(Public Beta Environment,基于口令加密)等。

[0080] 采用对称加密算法,对上述两步骤生成的加密key和加密value进行再加密,得到token＝encrypt(value,key),本方案优选AES加密。

[0081] 具体加密执行代码如下:

```
/
* aes 加密方法
* @param ciphertext - 要加密的字符串
* @param key - 密钥字符串
```

[0082]

```
* @returns string
*/
function encryptByAes(val,keyStr){
  let key = CryptoJS.enc.Utf8.parse(keyStr);//字符串类型的 key 用之
```

前需要用 uft8 先 parse 一下才能用

//CryptoJS 生成的密文是一个对象，

//如果直接将其转为字符串是一个 Base64 编码过的，

//在 encryptedData.ciphertext 上的属性转为字符串才是后端需要的格式

[0083]

```
let encryptedData=CryptoJS.AES.encrypt(val, key);
return encryptedData.ciphertext.toString();
}
```

[0084] M-Agent检测代理信息,包括渠道/包名/平台类型/系统平台/系统版本/分辨率/客户端唯一标识/请求时间戳/token,如“kepler.xx.com/com.xx.kepler/wx/Android/4.1.2/320x765/0/100000000/e7f7892c21e8f8276781b2688f5ff4c”,其中,

[0085] ①kepler.xx.com是渠道名称,固定值;

[0086] ②com.xx.kepler是包名,固定值;

[0087] ③wx是平台名称,此处代表小程序,如果是pc端可以传web,也是开发时就写好的固定值;

[0088] ④Android是系统平台,安卓机是Android,苹果机则是ios,这个值是根据宿主机型获取到的;

[0089] ⑤4.1.2系统版本,根据宿主机型获取;

[0090] ⑥320x765分辨率,单位px,宿主手机的屏幕分辨率;

[0091] ⑦0uuid,此处取固定值0;

[0092] ⑧100000000,请求的时间戳,即请求的时间。该时间是是指格林威治时间1970年01月01日00时00分00秒(北京时间1970年01月01日08时00分00秒)起至现在的总秒数。所以是个唯一值。

[0093] ⑨e7f7892c21e8f8276781b2688f5ff4c,token值(即令牌,用于加解密的字符串),是接口安全实现最重要的标示,是用来加解密的主要字段。

[0094] 因而,渠道/包名/平台名称这三个参数是写在代码里的固定值,系统版本/分辨率/客户端唯一标识这三个参数,在微信小程序中是通过wx.getSystemInfo这个api可以获取到。在浏览器中则可以通过navigator.userAgent来获取。因而,渠道/包名/平台类型/系统平台/系统版本/分辨率/客户端唯一标识/等信息不用做接口校验,只用做信息上报,所以不做过多解释。

[0095] 将M-Agent信息放在请求头header中,得到加入M-Agent信息的调用接口请求发送给服务端进行处理。

[0096] 上述实施例所提供的方法,前后端通过采用一定的规则,对调用接口请求中的时间戳等请求参数进行加密处理,以约定一个M-Agent字段并放到http/https请求头中,且M-agent信息用过一次之后就不能使用了,可以有效防止被刷。

[0097] 参见图2,示出了根据本发明实施例的一种接口安全校验方法流程示意图,包括如下步骤:

[0098] S201:服务端接收客户端传输的调用接口请求,从所述调用接口请求的请求头中抽取出检测代理信息和请求参数;

[0099] S202:根据所述请求参数生成第一加密键值对,以及对所述检测代理信息进行拆解,得到加密令牌并解密,得到第二加密键值对;

[0100] S203:比对所述第一加密键值对和所述第二加密键值对是否相同;

[0101] S204:若比对结果相同,则触发对所述调用接口请求的处理,返回响应结果至所述客户端;

[0102] S205:否则,返回令牌校验错误信息至所述客户端。

[0103] 上述实施方式中,对于步骤S201,服务端接收客户端传输的调用接口请求,该请求的请求头中包含M-Agent信息和请求参数,如第一请求时间戳、第一接口名、第一参数字符串集和第一平台类型,这些信息均可以从请求中抽取出来。

[0104] 对于步骤S202,同客户端生成键值对的方式一致,此处生成第一加密key的加密规则为:将第一请求时间戳转制为二进制数,截取预设数量的二进制数(从后向前或从前向后截取31位),对截取出的二进制数取反并转制为十进制,得到十进制形式的加密key。实际操作中,也可以根据js或者java生成的随机数,然后通过某种加密规则生成加密key。加密key的长度与客户端的要求一致。

[0105] 对请求参数中的第一请求时间戳、第一接口名、第一参数字符串集和第一平台类型,使用字符“-”(仅为示例,还可以是其他字符)对其进行按序拼接,经过MD5加密以生成加密value,具体形式如MD5(第一接口名-第一请求时间戳-第一参数字符串集-第一平台类型)。需要说明的是,对加密value中各参数的数量不做限制,如第一接口名可以被拼接两次,且各参数的含义和执行代码同客户端一致。

[0106] 对请求头中的M-Agent信息进行拆解,得到加密令牌、第一请求时间戳和第一特定参数(如渠道/包名/平台类型/系统平台/系统版本/分辨率/客户端唯一标识)。采用与客户端加密方式(如AES)对应的解密方式,对该加密令牌进行对称解密,得到第二加密键值对。

[0107] 另外,M-Agent信息中也包含第一请求时间戳,因此也可以直接对M-Agent信息中的第一请求时间戳,基于加密规则,生成第一加密key。

[0108] 对于步骤S203～S205,比对第一键值对和第二键值对,若值相等则处理调用接口请求,返回响应数据给客户端,否则返回“token校验错误”给客户端。

[0109] 上述实施例所提供的方法,通过在前后端约定一个加密解密的规则,设置http/https请求头中的M-Agent字段来进行校验,判断解密得到的信息与客户端根据参数生成的信息是否一致,以此实现接口安全监控。

[0110] 本发明实施例提供一种防御方案,通过在前后端约定一个加密解密的规则,设置http/https请求头中的M-Agent字段来进行校验,以解决灰产伪造用户来恶意调用接口进行谋利的问题。其还具备有益效果:

[0111] 1)可以在服务端处理调用接口请求之前进行校验,而非像IP限制类,等接口被调用后再进行的补救;

[0112] 2)可以实现Web和App跨端的统一,不涉及对手机底层的权限获取,通用性比较强。

[0113] 3)不需要额外的支出,像短信验证码类是需要一定资金支持的,且M-agent信息用过一次之后就不能使用了,可以有效防止被刷。

[0114] 参见图3，示出了本发明实施例提供的一种接口安全控制装置300的主要模块示意图，包括：

[0115] 监测模块301，用于在监测到客户端的调用接口动作时，拦截所述客户端发往服务端的调用接口请求，根据所述调用接口请求的请求参数，生成加密键值对；其中，所述请求参数包括请求时间戳；

[0116] 加密模块302，用于对所述加密键值对进行对称加密处理，得到加密令牌，按序拼接所述加密令牌、所述请求时间戳和特定参数，以生成检测代理信息；

[0117] 传输模块303，用于将所述检测代理信息添加到所述调用接口请求的请求头中，传输包含检测代理信息的调用接口请求至所述服务端进行处理。

[0118] 本发明实施装置中，所述请求参数还包括接口名、参数字符串集和平台类型；

[0119] 所述监测模块301，用于：基于预设加密规则处理所述请求时间戳，得到加密键名；

[0120] 按序拼接所述接口名、所述请求时间戳、所述参数字符串集和所述平台类型，采用信息摘要算法处理拼接后的信息，得到加密键值。

[0121] 本发明实施装置中，所述监测模块301，用于：

[0122] 将所述请求时间戳转制为二进制形式，截取预设数量的二进制数，对截取的二进制数进行取反并转制为十进制形式，得到加密键名。

[0123] 参见图4，示出了本发明实施例提供的一种接口安全校验装置400的主要模块示意图，包括：

[0124] 抽取模块401，用于服务端接收客户端传输的调用接口请求，从所述调用接口请求的请求头中抽取出检测代理信息和请求参数；

[0125] 解密模块402，用于根据所述请求参数生成第一加密键值对，以及对所述检测代理信息进行拆解，得到加密令牌并解密，得到第二加密键值对；

[0126] 处理模块403，用于比对所述第一加密键值对和所述第二加密键值对是否相同，若比对结果相同，则触发对所述调用接口请求的处理，返回响应结果至所述客户端，否则返回令牌校验错误信息至所述客户端。

[0127] 本发明实施装置中，所述请求参数包括第一请求时间戳、第一接口名、第一参数字符串集和第一平台类型；

[0128] 所述解密模块402，用于：基于预设加密规则处理所述第一请求时间戳，得到第一加密键名；

[0129] 按序拼接所述第一接口名、所述第一请求时间戳、所述第一参数字符串集和所述第一平台类型，采用信息摘要算法处理拼接后的信息，得到第一加密键值。

[0130] 本发明实施装置中，所述解密模块402，用于：将所述第一请求时间戳转制为二进制形式，截取预设数量的二进制数，对截取的二进制数进行取反并转制为十进制形式，得到第一加密键名。

[0131] 另外，在本发明实施例中所述装置的具体实施内容，在上面所述方法中已经详细说明了，故在此重复内容不再说明。

[0132] 图5示出了可以应用本发明实施例的示例性系统架构500。

[0133] 如图5所示，系统架构500可以包括终端设备501、502、503，网络504和服务器505(仅仅是示例)。网络504用以在终端设备501、502、503和服务器505之间提供通信链路的介

质。网络504可以包括各种连接类型，例如有线、无线通信链路或者光纤电缆等等。

[0134] 用户可以使用终端设备501、502、503通过网络504与服务器505交互，以接收或发送消息等。终端设备501、502、503上可以安装有各种通讯客户端应用。

[0135] 终端设备501、502、503可以是具有显示屏并且支持网页浏览的各种电子设备，服务器505可以是提供各种服务的服务器。

[0136] 需要说明的是，本发明实施例所提供的方法一般由服务器505执行，相应地，装置一般设置于服务器505中。

[0137] 应该理解，图5中的终端设备、网络和服务器的数目仅仅是示意性的。根据实现需要，可以具有任意数目的终端设备、网络和服务器。

[0138] 下面参考图6，其示出了适于用来实现本发明实施例的终端设备的计算机系统600的结构示意图。图6示出的终端设备仅仅是一个示例，不应对本发明实施例的功能和使用范围带来任何限制。

[0139] 如图6所示，计算机系统600包括中央处理单元(CPU)601，其可以根据存储在只读存储器(ROM)602中的程序或者从存储部分608加载到随机访问存储器(RAM)603中的程序而执行各种适当的动作和处理。在RAM 603中，还存储有系统600操作所需的各种程序和数据。CPU 601、ROM 602以及RAM 603通过总线604彼此相连。输入/输出(I/O)接口605也连接至总线604。

[0140] 以下部件连接至I/O接口605：包括键盘、鼠标等的输入部分606；包括诸如阴极射线管(CRT)、液晶显示器(LCD)等以及扬声器等的输出部分607；包括硬盘等的存储部分608；以及包括诸如LAN卡、调制解调器等的网络接口卡的通信部分609。通信部分609经由诸如因特网的网络执行通信处理。驱动器610也根据需要连接至I/O接口605。可拆卸介质611，诸如磁盘、光盘、磁光盘、半导体存储器等等，根据需要安装在驱动器610上，以便于从其上读出的计算机程序根据需要被安装入存储部分608。

[0141] 特别地，根据本发明公开的实施例，上文参考流程图描述的过程可以被实现为计算机软件程序。例如，本发明公开的实施例包括一种计算机程序产品，其包括承载在计算机可读介质上的计算机程序，该计算机程序包含用于执行流程图所示的方法的程序代码。在这样的实施例中，该计算机程序可以通过通信部分609从网络上被下载和安装，和/或从可拆卸介质611被安装。在该计算机程序被中央处理单元(CPU)601执行时，执行本发明的系统中限定的上述功能。

[0142] 需要说明的是，本发明所示的计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质或者是上述两者的任意组合。计算机可读存储介质例如可以是——但不限于——电、磁、光、电磁、红外线、或半导体的系统、装置或器件，或者任意以上的组合。计算机可读存储介质的更具体的例子可以包括但不限于：具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机访问存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、光纤、便携式紧凑磁盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本发明中，计算机可读存储介质可以是任何包含或存储程序的有形介质，该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。而在本发明中，计算机可读的信号介质可以包括在基带中或者作为载波一部分传播的数据信号，其中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式，包括但不限

于电磁信号、光信号或上述的任意合适的组合。计算机可读的信号介质还可以是计算机可读存储介质以外的任何计算机可读介质，该计算机可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。计算机可读介质上包含的程序代码可以用任何适当的介质传输，包括但不限于：无线、电线、光缆、RF等等，或者上述的任意合适的组合。

[0143] 附图中的流程图和框图，图示了按照本发明各种实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上，流程图或框图中的每个方框可以代表一个模块、程序段、或代码的一部分，上述模块、程序段、或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意，在有些作为替换的实现中，方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如，两个接连地表示的方框实际上可以基本并行地执行，它们有时也可以按相反的顺序执行，这依所涉及的功能而定。也要注意的是，框图或流程图中的每个方框、以及框图或流程图中的方框的组合，可以用执行规定的功能或操作的专用的基于硬件的系统来实现，或者可以用专用硬件与计算机指令的组合来实现。

[0144] 描述于本发明实施例中所涉及到的模块可以通过软件的方式实现，也可以通过硬件的方式来实现。所描述的模块也可以设置在处理器中，例如，可以描述为：一种处理器包括监测模块、加密模块、传输模块。其中，这些模块的名称在某种情况下并不构成对该模块本身的限定，例如，加密模块还可以被描述为“令牌加密模块”。

[0145] 作为另一方面，本发明还提供了一种计算机可读介质，该计算机可读介质可以是上述实施例中描述的设备中所包含的；也可以是单独存在，而未装配入该设备中。上述计算机可读介质承载有一个或者多个程序，当上述一个或者多个程序被一个该设备执行时，使得该设备包括：

[0146] 在监测到客户端的调用接口动作时，拦截所述客户端发往服务端的调用接口请求，根据所述调用接口请求的请求参数，生成加密键值对；其中，所述请求参数包括请求时间戳；

[0147] 对所述加密键值对进行对称加密处理，得到加密令牌，按序拼接所述加密令牌、所述请求时间戳和特定参数，以生成检测代理信息；

[0148] 将所述检测代理信息添加到所述调用接口请求的请求头中，传输包含检测代理信息的调用接口请求至所述服务端进行处理。

[0149] 本发明实施例提供一种防御方案，通过在前后端约定一个加密解密的规则，设置http/https请求头中的M-Agent字段来进行校验，以解决灰产伪造用户来恶意调用接口进行谋利的问题。其还具备有益效果：

[0150] 1)可以在服务端处理调用接口请求之前进行校验，而非像IP限制类，等接口被调用后再进行的补救；

[0151] 2)可以实现Web和App跨端的统一，不涉及对手机底层的权限获取，通用性比较强。

[0152] 3)不需要额外的支出，像短信验证码类是需要一定资金支持的，且M-agent信息用过一次之后就不能使用了，可以有效防止被刷。

[0153] 上述具体实施方式，并不构成对本发明保护范围的限制。本领域技术人员应该明白的是，取决于设计要求和其他因素，可以发生各种各样的修改、组合、子组合和替代。任何

在本发明的精神和原则之内所作的修改、等同替换和改进等，均应包含在本发明保护范围之内。

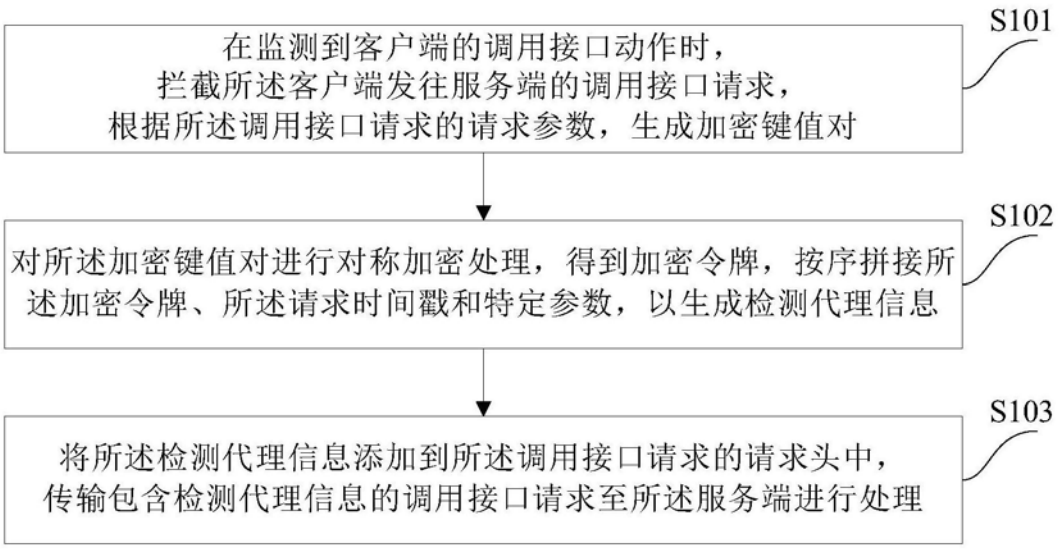

S101
在监测到客户端的调用接口动作时，
拦截所述客户端发往服务端的调用接口请求，
根据所述调用接口请求的请求参数，生成加密键值对
S102
对所述加密键值对进行对称加密处理，得到加密令牌，按序拼接所述加密令牌、所述请求时间戳和特定参数，以生成检测代理信息
S103
将所述检测代理信息添加到所述调用接口请求的请求头中，
传输包含检测代理信息的调用接口请求至所述服务端进行处理

图1

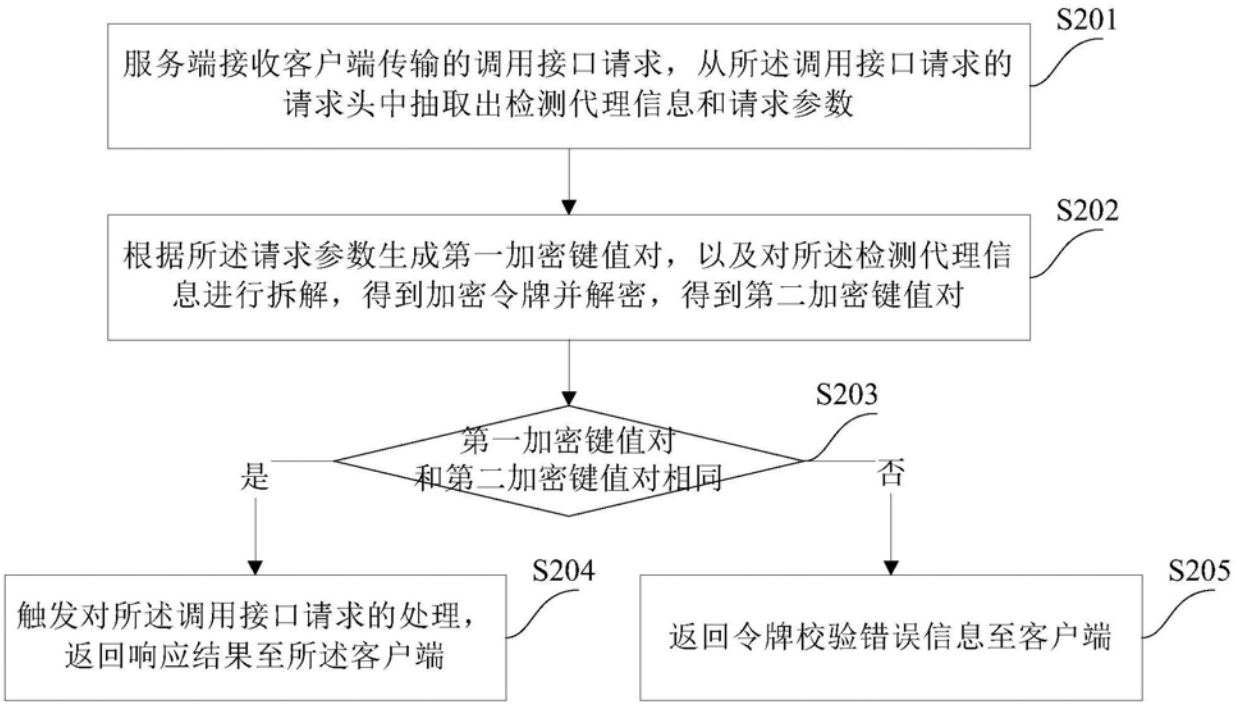

S201
服务端接收客户端传输的调用接口请求，从所述调用接口请求的请求头中抽取出检测代理信息和请求参数
S202
根据所述请求参数生成第一加密键值对，以及对所述检测代理信息进行拆解，得到加密令牌并解密，得到第二加密键值对
S203
第一加密键值对
和第二加密键值对相同
是
否
S204
触发对所述调用接口请求的处理，
返回响应结果至所述客户端
S205
返回令牌校验错误信息至客户端

图2

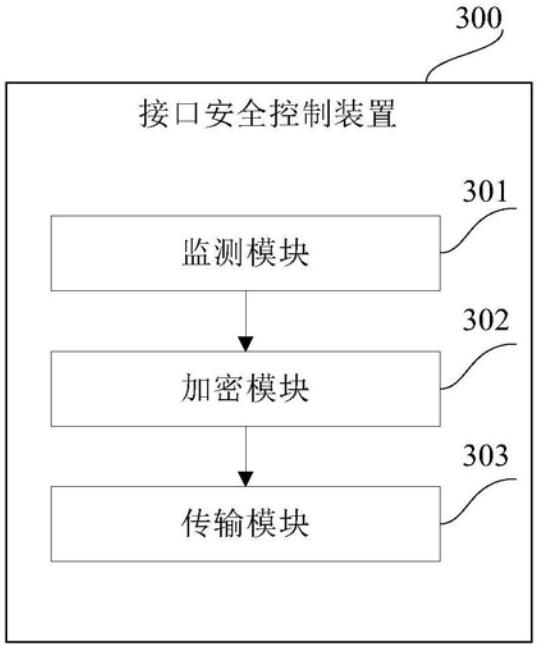

300
接口安全控制装置
301
监测模块
302
加密模块
303
传输模块

图3

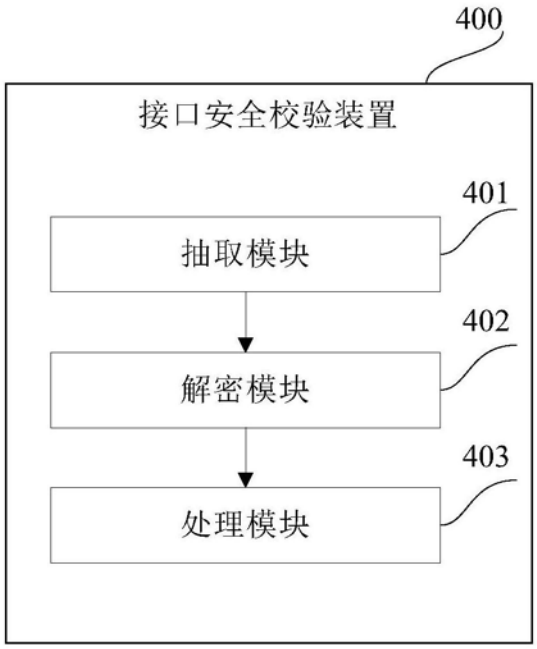

400
接口安全校验装置
401
抽取模块
402
解密模块
403
处理模块

图4

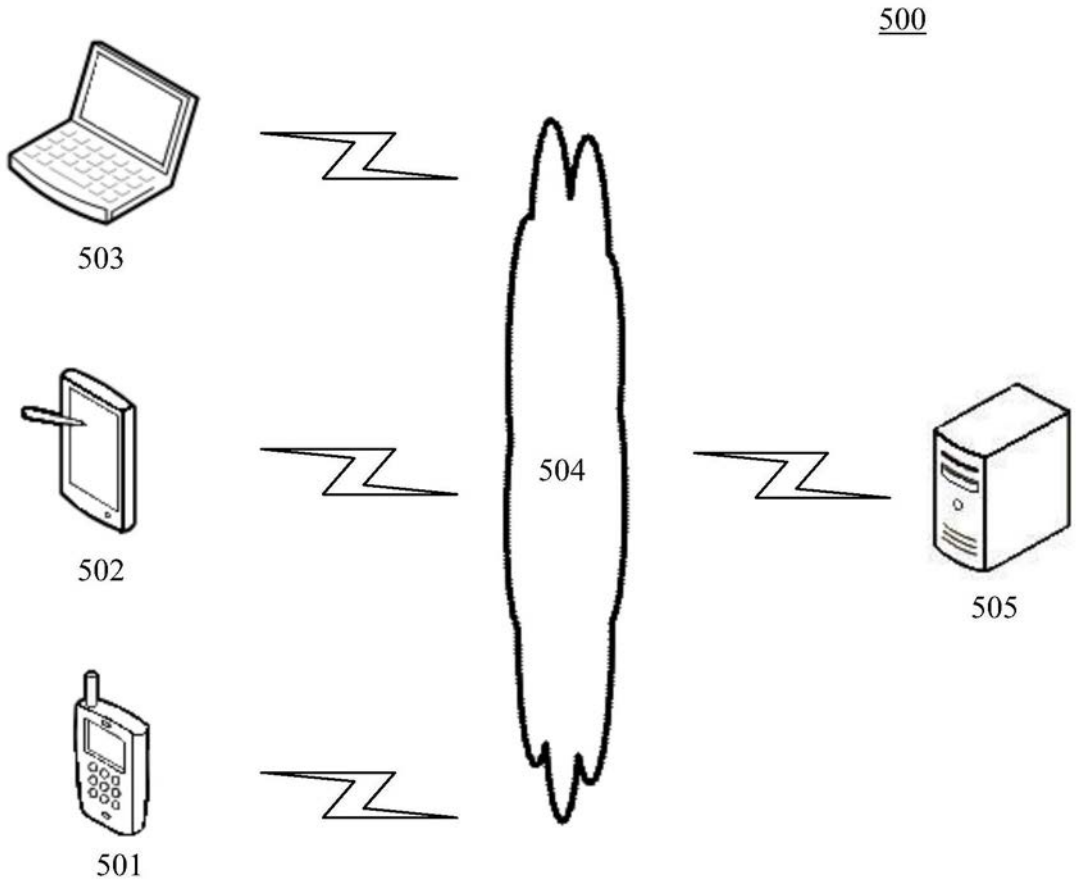


图5

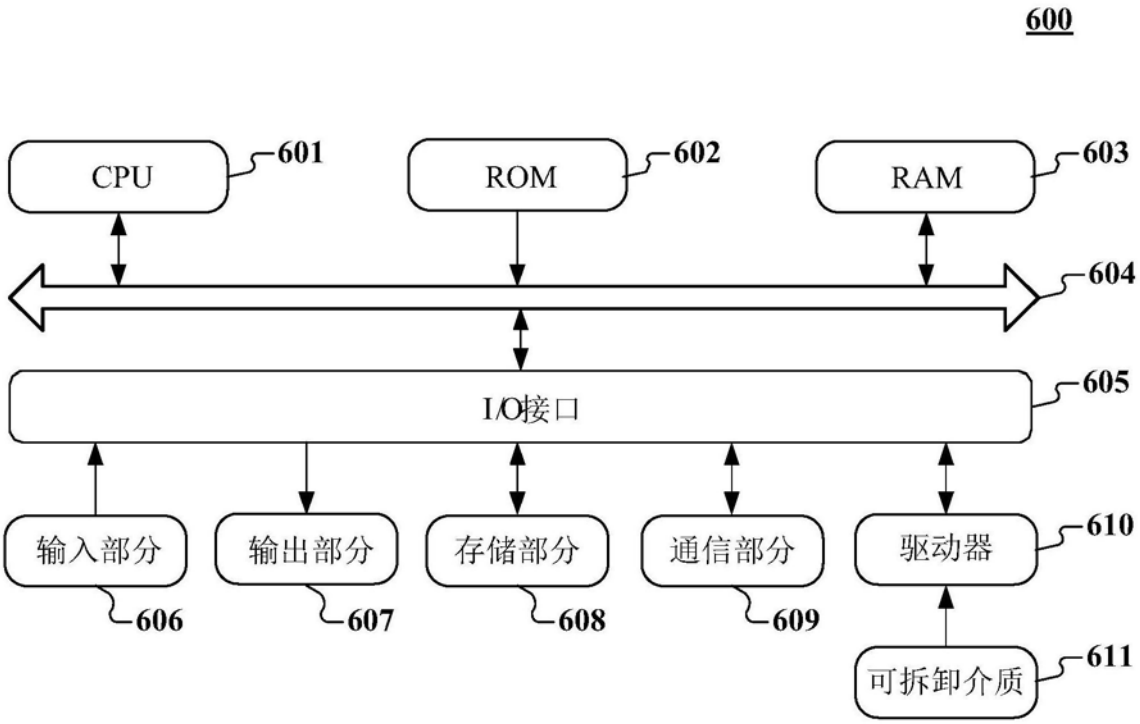

600
CPU
601
ROM
602
RAM
603
604
I/O接口
605
输入部分
606
输出部分
607
存储部分
608
通信部分
609
驱动器
610
可拆卸介质
611

图6