(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 113449330 A
(43)申请公布日 2021.09.28

(21)申请号 202111015995.5

(22)申请日 2021.08.31

(71)申请人 北京华云安信息技术有限公司
地址 100094 北京市海淀区丰豪东路9号2号楼10层4单元1001

(72)发明人 韩旭 马维士 吴璇

(74)专利代理机构 北京华专卓海知识产权代理事务所(普通合伙) 11664
代理人 孙振韬 王一

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

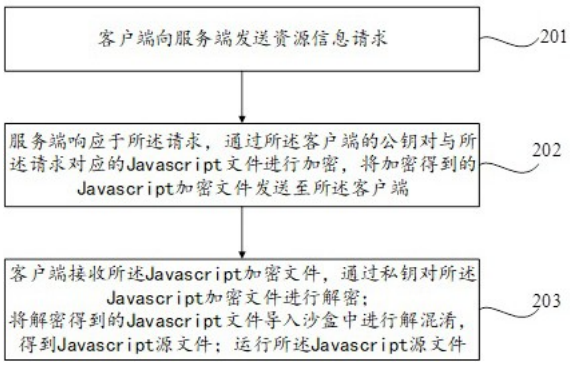
权利要求书1页 说明书8页 附图2页

(54)发明名称

对Javascript加密文件进行传输的方法

(57)摘要

本公开的实施例提供了对Javascript加密文件进行传输的方法、设备和计算机可读存储介质。所述方法包括客户端向服务端发送资源信息请求;所述请求包括客户端的公钥;服务端响应于所述请求,通过所述客户端的公钥对与所述请求对应的Javascript文件进行加密,将加密得到的Javascript加密文件发送至所述客户端;所述Javascript文件是通过对Javascript源文件进行混淆得到的;客户端接收所述Javascript加密文件,通过私钥对所述Javascript加密文件进行解密;将解密得到的Javascript文件导入沙盒中进行解混淆,得到Javascript源文件;运行所述Javascript源文件。实现了对Javascript文件的保护。



CN 113449330 A

1.一种对Javascript加密文件进行传输的方法,其特征在于,包括:

客户端向服务端发送资源信息请求;所述请求包括客户端的公钥;

服务端响应于所述请求,通过所述客户端的公钥对与所述请求对应的Javascript文件进行加密,将加密得到的Javascript加密文件发送至所述客户端;所述Javascript文件是通过对Javascript源文件进行混淆得到的;

客户端接收所述Javascript加密文件,通过私钥对所述Javascript加密文件进行解密;将解密得到的Javascript文件导入沙盒中进行解混淆,得到Javascript源文件;运行所述Javascript源文件。

2.根据权利要求1所述的方法,其特征在于,所述对Javascript源文件进行混淆包括:

对所述Javascript源文件进行无损压缩,并将压缩后的Javascript源文件转换成抽象语法树;

通过预设的加密策略,对所述抽象语法树中的节点进行加密;

将完成加密的抽象语法树重新转换成Javascript文件,完成对所述Javascript源文件的混淆。

3.根据权利要求2所述的方法,其特征在于,所述预设的加密策略包括:

对所述Javascript源文件中的变量进行混淆处理;

加入代码防转换操作;

禁止控制台的输出和调试;和/或

设置安全域名。

4.根据权利要求3所述的方法,其特征在于,所述加入代码防转换操作包括:

加入防转换代码,防止浏览器对所述Javascript源文件进行转换。

5.根据权利要求4所述的方法,其特征在于,所述禁止控制台的输出和调试包括:

通过闭包或设置隐藏域的方式,禁止控制台输出和调试所述Javascript源文件。

6.根据权利要求5所述的方法,其特征在于,所述将解密得到的Javascript文件导入沙盒中进行解混淆,得到Javascript源文件包括:

将解密得到的Javascript文件导入沙盒中,通过与所述Javascript源文件混淆对应的解混淆策略,对所述Javascript文件解混淆,得到Javascript源文件。

7.根据权利要求6所述的方法,其特征在于,所述通过预设的加密策略,对所述抽象语法树中的节点进行加密包括:

遍历所述抽象语法树,确定所述抽象语法树中每一个节点对应的代码结构;

通过预设的加密策略,对所述代码结构进行加密。

8.一种电子设备,包括存储器和处理器,所述存储器上存储有计算机程序,其特征在于,所述处理器执行所述程序时实现如权利要求1~7中任一项所述的方法。

9.一种计算机可读存储介质,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1~7中任一项所述的方法。

对Javascript加密文件进行传输的方法

技术领域

[0001] 本公开的实施例一般涉及数据处理领域，并且更具体地，涉及对Javascript加密文件进行传输的方法、设备和计算机可读存储介质。

背景技术

[0002] 前端技术的发展是互联网自身发生变化的一个缩影，随着物联网的发展，前端技术也逐渐从简单的展示图文信息到需要处理一些复杂逻辑实现良好的交互模式使用户体验有了空前的改变。在前端技术领域中，其核心技术就包含了HTML、CSS、Javascript。

[0003] HTML是页面的核心，CSS决定了界面的视觉效果、Javascript编写前端交互组件。早期的web系统中Javascript仅仅负责一些简单的表单校验，文件内容也非常简单，基本没有保护的必要，但是随着项目的扩展和技术的优化，Javascript也开始逐渐变得复杂，很多后端的业务逻辑已经向前端转移，于此同时给了更多的不法分子可乘之机，如，模拟成正常用户来实施一些恶意行为。所以前端对于Javascript文件的保护，已经成为一个必不可少的工作。

[0004] 当前常见的对Javascript代码保护的方法为，对代码进行压缩与混淆，这种代码混淆是在原有代码基本结构不变的条件基础上进行的，更像是一种替换编译，将常见的变量名和函数名通过一些计算方式混淆替换。从而降低Javascript文件的体积，以及可读性。

[0005] 但是，实际上代码并没有被有效的保护，浏览器可通过自带的功能对Javascript文件进行更改，例如，通过浏览器的美化功能对Javascript文件进行转换；通过debug功能对Javascript文件进行调试等。

发明内容

[0006] 根据本公开的实施例，提供了一种对Javascript加密文件进行传输方案。

[0007] 在本公开的第一方面，提供了一种对Javascript加密文件进行传输方法。该方法包括：

客户端向服务端发送资源信息请求；所述请求包括客户端的公钥；

服务端响应于所述请求，通过所述客户端的公钥对与所述请求对应的Javascript文件进行加密，将加密得到的Javascript加密文件发送至所述客户端；所述Javascript文件是通过对Javascript源文件进行混淆得到的；

客户端接收所述Javascript加密文件，通过私钥对所述Javascript加密文件进行解密；将解密得到的Javascript文件导入沙盒中进行解混淆，得到Javascript源文件；运行所述Javascript源文件。

[0008] 进一步地，所述对Javascript源文件进行混淆包括：

对所述Javascript源文件进行无损压缩，并将压缩后的Javascript源文件转换成抽象语法树；

通过预设的加密策略，对所述抽象语法树中的节点进行加密；

将完成加密的抽象语法树重新转换成Javascript文件，完成对所述Javascript源文件的混淆。

[0009] 进一步地，所述预设的加密策略包括：

对所述Javascript源文件中的变量进行混淆处理；

加入代码防转换操作；

禁止控制台的输出和调试；和/或

设置安全域名。

[0010] 进一步地，所述加入代码防转换操作包括：

加入防转换代码，防止浏览器对所述Javascript文件源文件进行转换。

[0011] 进一步地，所述禁止控制台的输出和调试包括：

通过闭包或设置隐藏域的方式，禁止控制台输出和调试所述Javascript源文件。

[0012] 进一步地，所述将解密得到的Javascript文件导入沙盒中进行解混淆，得到Javascript源文件包括：

将解密得到的Javascript文件导入沙盒中，通过与所述Javascript源文件混淆对应的解混淆策略，对所述Javascript文件解混淆，得到Javascript源文件。

[0013] 进一步地，所述通过预设的加密策略，对所述抽象语法树中的节点进行加密包括：

遍历所述抽象语法树，确定所述抽象语法树中每一个节点对应的代码结构；

通过预设的加密策略，对所述代码结构进行加密。

[0014] 在本公开的第二方面，提供了一种电子设备。该电子设备包括：存储器和处理器，所述存储器上存储有计算机程序，所述处理器执行所述程序时实现如以上所述的方法。

[0015] 在本公开的第三方面，提供了一种计算机可读存储介质，其上存储有计算机程序，所述程序被处理器执行时实现如根据本公开的第一方面的方法。

[0016] 本申请实施例提供的对Javascript加密文件进行传输的方法，通过客户端向服务端发送资源信息请求；所述请求包括客户端的公钥；服务端响应于所述请求，通过所述客户端的公钥对与所述请求对应的Javascript文件进行加密，将加密得到的Javascript加密文件发送至所述客户端；所述Javascript文件是通过对Javascript源文件进行混淆得到的；客户端接收所述Javascript加密文件，通过私钥对所述Javascript加密文件进行解密；将解密得到的Javascript文件导入沙盒中进行解混淆，得到Javascript源文件；运行所述Javascript源文件，实现了对Javascript文件的保护。

[0017] 应当理解，发明内容部分中所描述的内容并非旨在限定本公开的实施例的关键或重要特征，亦非用于限制本公开的范围。本公开的其它特征将通过以下的描述变得容易理解。

附图说明

[0018] 结合附图并参考以下详细说明，本公开各实施例的上述和其他特征、优点及方面将变得更加明显。在附图中，相同或相似的附图标记表示相同或相似的元素，其中：

图1示出了能够在其中实现本公开的实施例的示例性运行环境的示意图；

图2示出了根据本公开的实施例的对Javascript加密文件进行传输的方法的流程图；

图3示出了能够实施本公开的实施例的示例性电子设备的方框图。

具体实施方式

[0019] 为使本公开实施例的目的、技术方案和优点更加清楚，下面将结合本公开实施例中的附图，对本公开实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例是本公开一部分实施例，而不是全部的实施例。基于本公开中的实施例，本领域普通技术人员在没有作出创造性劳动前提下所获得的全部其他实施例，都属于本公开保护的范围。

[0020] 另外，本文中术语“和/或”，仅仅是一种描述关联对象的关联关系，表示可以存在三种关系，例如，A和/或B，可以表示：单独存在A，同时存在A和B，单独存在B这三种情况。另外，本文中字符“/”，一般表示前后关联对象是一种“或”的关系。

[0021] 图1示出了能够在其中实现本公开的实施例的示例性运行环境100的示意图。在运行环境100中包括客户端101、网络102和服务端103。

[0022] 网络102用以在客户端101和服务端103之间提供通信链路的介质。网络102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。

[0023] 用户可以使用客户端101通过网络102与服务端103交互，以接收或发送消息等。例如，接收服务端103发送的Javascript文件等。

[0024] 图1中的客户端、网络和服务端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客户端、网络和服务端。特别地，在目标数据不需要从远程获取的情况下，上述系统架构可以不包括网络，而只包括客户端或服务端。

[0025] 图2示出了根据本公开实施例的对Javascript加密文件进行传输的方法200的流程图。方法200包括：

S210，客户端向服务端发送资源信息请求；所述请求包括客户端的公钥。

[0026] 在一些实施例中，用于对Javascript加密文件进行传输的方法的执行主体(例如图1所示的服务端)可以通过有线方式或者无线连接的方式获取资源信息请求。

[0027] 进一步地，上述执行主体可以获取与之通信连接的电子设备(例如图1所示的客户端)发送的资源信息请求。

[0028] 通常情况下，为了保证数据的传输安全，客户端101在发送所述请求时会将自身的公钥一同发送至服务端103，便于服务端103用所述公钥对返回的响应信息进行加密。

[0029] S220，服务端响应于所述请求，通过所述客户端的公钥对与所述请求对应的Javascript文件进行加密，将加密得到的Javascript加密文件发送至所述客户端；所述Javascript文件是通过对Javascript源文件进行混淆得到的。

[0030] 在一些实施例中，服务端103响应于客户端101发送的请求，通过所客户端101的公钥对与所述请求对应的Javascript文件进行加密，将加密得到的Javascript加密文件发送至客户端101；

其中，所述Javascript文件是通过对Javascript源文件进行混淆得到的。

[0031] 在一些实施例中，可通过如下方法对Javascript源文件进行混淆：

对Javascript源文件进行无损压缩，去除所述Javascript源文件中的多余符号和注释，减少Javascript源文件的体积，将压缩后的Javascript源文件转换成抽象语法树，在保证代码整体的结构关系不发生改变的同时，可以保证加密之后的Javascript文件仍然可

以在项目中正常运行；

将Javascript源文件中的代码转化成抽象语法树中的一个对象(一段代码转化为一个对象)；

如通过以下方法将Javascript源文件中的代码“var AST=“is Tree”;”转化为抽象语法树中的一个对象：

其中,所述var是一个关键字；

所述AST是一个定义者；

所述“is Tree”是一个字符串；

转化为抽象语法数的对象为：

```
{“type”: “Program”,
 “body”:[{
      “type”: “VariableDeclaration”,
      “kind”: “var”
      “declaration”:[{
            “type”: “VariableDeclarator”,
            “id”:{
                 “type”: “Identifier”
                 “name”: “AST”
            },
            “init”:{
                  “type”: “Literal”,
                  “value”: “is tree”,
                  “raw”: “\“is tree\” ”
                  }
              }]
}]
}
```

其中,对象中可包含一个顶级的type属性“program”和属性body(数组),body数组中存放的每一项都是一个对象,里面包含了所有的对于该语句的描述：

type:描述该语句的类型 --变量声明语句；

kind:变量声明的关键字 - var；

declaration: 声明的内容数组,里面的每一项也是一个对象；

type: 描述该语句的类型；

id: 描述变量名称的对象；

type:定义；

name: 是变量的名字；

init: 初始化变量值得对象；

type: 类型；

value: 值 "is tree" 不带引号；

row: "\"is tree"\" 带引号;

进一步地,遍历抽象语法树上的所有节点,通过预设的加密策略,对所述抽象语法树中的节点进行加密;

所述加密策略包括:

对所述Javascript源文件中的变量进行混淆处理;加入代码防转换操作;禁止控制台的输出和调试;和/或设置安全域名等。

[0032] 其中,对所述Javascript源文件中的变量进行混淆处理包括:

加入特定的变量混淆规则,对所述Javascript源文件中的变量进行混淆;

如对Javascript源文件中的名称进行混淆;通常所述Javascript源文件中的名称包括变量名、函数名和标签名;

进一步地,变量混淆规则包括:

对作用域链上边的声明过的名字进行混淆;

对函数声明时的参数名进行混淆;

对参数列表中的名字进行混淆,如:

```
function A(){
    var myName = "A";
    function B(){
        myName = "B";
        with(obj){
            myName = "with";
        }
    }
}
```

加入代码防转换操作包括:

现有浏览器中,大部分提供了Javascript格式化功能,能够对Javascript文件进行转换(美化),即,会对源代码进行破坏,因此,在本公开中加入了代码防转换操作,防止浏览器对所述Javascript源文件进行转换(美化);

所述禁止控制台的输出和调试包括:

在本公开中,可通过闭包或设置隐藏域的方式,禁止控制台输出和调试所述Javascript源文件;

其中,通过闭包的方式,禁止控制台输出和调试所述Javascript源文件包括:

```
//在window对象中定义一个dome对象。
window.dome=(function(){
  function _creat1(){
  //_create方法只能在window下的dome对象中访问,离开dome对象无法访问。
    alert('create table1');
  }
  function _create2(){
   //_create方法能在window下的dome对象外被访问,因为在return中被返回
```

```
        alert('create table2');
      }
      function start(){
        _create1();//结果是"create table1"
      }
      return{
      //写在return里面的key-value可让外部调用访问
        start:start,
        _create2:_create2
      };
    })();
    window.dome.start();//当dome加载完毕后,马上执行dome对象里的start方法。
    _creat1();//调用错误,方法不存在
    dome._creat2();//调用正确,结果是"create table2"
```

通过设置隐藏域的方式,禁止控制台输出和调试所述Javascript源文件包括:

先把脚本(Javascript)注入一个隐藏域中;

再将原脚本的text设置为空;

再需要的时候,将隐藏域的脚本注入到一个新的脚本对象;

所述设置安全域名包括:

设置安全域名,仅允许加密后的Javascript文件在设置的域名下运行,在非设置的域名下禁止Javascript文件运行。

[0036] S230,客户端接收所述Javascript加密文件,通过私钥对所述Javascript加密文件进行解密;将解密得到的Javascript文件导入沙盒中进行解混淆,得到Javascript源文件;运行所述Javascript源文件。

[0037] 在一些实施例中,客户端101接收服务端103发送的Javascript加密文件;所述Javascript加密文件为二次加密文件,即,通过混淆加密后,再次通过客户端101公钥加密;

通过客户端101的私钥进行一次解密,即,对接收的Javascript加密文件进行脱壳处理;

将一次解密后的Javascript文件,导入JS沙箱,保证沙箱内所有的变量访问都在可监控范围内,以防止读取来的内容对当前项目环境造成代码污染,进行解混淆处理,即,将抽象语法树转换为Javascript源文件(代码),运行Javascript源文件;所述JS沙箱中包括与上述加密策略对应的解码规则。

[0038] 根据本公开的实施例,实现了以下技术效果:

将Javascript源文件转换为抽象语法树,通过预设的加密策略对所述抽象语法树中的节点进行加密(混淆),将混淆后的Javascript源文件通过端到端(客户端到服务端)的加密(公钥加密、私钥解密)进行传输,实现了对Javascript源文件的保护,防止了浏览器通过调试台获取Javascript源文件内容。

[0039] 需要说明的是,对于前述的各方法实施例,为了简单描述,故将其都表述为一系列的动作组合,但是本领域技术人员应该知悉,本公开并不受所描述的动作顺序的限制,因为

依据本公开，某些步骤可以采用其他顺序或者同时进行。其次，本领域技术人员也应该知悉，说明书中所描述的实施例均属于可选实施例，所涉及的动作和模块并不一定是本公开所必须的。

[0040] 以上是关于方法实施例的介绍，以下通过装置实施例，对本公开所述方案进行进一步说明。

[0041] 图3示出了可以用来实施本公开的实施例的电子设备300的示意性框图。如图所示，设备300包括中央处理单元(CPU)301，其可以根据存储在只读存储器(ROM)302中的计算机程序指令或者从存储单元308加载到随机访问存储器(RAM)303中的计算机程序指令，来执行各种适当的动作和处理。在RAM 303中，还可以存储设备300操作所需的各种程序和数据。CPU 301、ROM 302以及RAM 303通过总线304彼此相连。输入/输出(I/O)接口305也连接至总线304。

[0042] 设备300中的多个部件连接至I/O接口305，包括：输入单元306，例如键盘、鼠标等；输出单元307，例如各种类型的显示器、扬声器等；存储单元308，例如磁盘、光盘等；以及通信单元309，例如网卡、调制解调器、无线通信收发机等。通信单元309允许设备300通过诸如因特网的计算机网络和/或各种电信网络与其他设备交换信息/数据。

[0043] 处理单元301执行上文所描述的各个方法和处理，例如方法200。例如，在一些实施例中，方法200可被实现为计算机软件程序，其被有形地包含于机器可读介质，例如存储单元308。在一些实施例中，计算机程序的部分或者全部可以经由ROM 302和/或通信单元309而被载入和/或安装到设备300上。当计算机程序加载到RAM 303并由CPU 301执行时，可以执行上文描述的方法200的一个或多个步骤。备选地，在其他实施例中，CPU 301可以通过其他任何适当的方式(例如，借助于固件)而被配置为执行方法200。

[0044] 本文中以上描述的功能可以至少部分地由一个或多个硬件逻辑部件来执行。例如，非限制性地，可以使用的示范类型的硬件逻辑部件包括：场可编程门阵列(FPGA)、专用集成电路(ASIC)、专用标准产品(ASSP)、芯片上系统的系统(SOC)、负载可编程逻辑设备(CPLD)等等。

[0045] 用于实施本公开的方法的程序代码可以采用一个或多个编程语言的任何组合来编写。这些程序代码可以提供给通用计算机、专用计算机或其他可编程数据处理装置的处理器或控制器，使得程序代码当由处理器或控制器执行时使流程图和/或框图中所规定的功能/操作被实施。程序代码可以完全在机器上执行、部分地在机器上执行，作为独立软件包部分地在机器上执行且部分地在远程机器上执行或完全在远程机器或服务器上执行。

[0046] 在本公开的上下文中，机器可读介质可以是有形的介质，其可以包含或存储以供指令执行系统、装置或设备使用或与指令执行系统、装置或设备结合地使用的程序。机器可读介质可以是机器可读信号介质或机器可读储存介质。机器可读介质可以包括但不限于电子的、磁性的、光学的、电磁的、红外的、或半导体系统、装置或设备，或者上述内容的任何合适组合。机器可读存储介质的更具体示例会包括基于一个或多个线的电气连接、便携式计算机盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦除可编程只读存储器(EPROM或快闪存储器)、光纤、便捷式紧凑盘只读存储器(CD-ROM)、光学储存设备、磁储存设备、或上述内容的任何合适组合。

[0047] 此外，虽然采用特定次序描绘了各操作，但是这应当理解为要求这样操作以所示

出的特定次序或以顺序次序执行，或者要求所有图示的操作应被执行以取得期望的结果。在一定环境下，多任务和并行处理可能是有利的。同样地，虽然在上面论述中包含了若干具体实现细节，但是这些不应当被解释为对本公开的范围的限制。在单独的实施例的上下文中描述的某些特征还可以组合地实现在单个实现中。相反地，在单个实现的上下文中描述的各种特征也可以单独地或以任何合适的子组合的方式实现在多个实现中。

[0048] 尽管已经采用特定于结构特征和/或方法逻辑动作的语言描述了本主题，但是应当理解所附权利要求书中所限定的主题未必局限于上面描述的特定特征或动作。相反，上面所描述的特定特征和动作仅仅是实现权利要求书的示例形式。

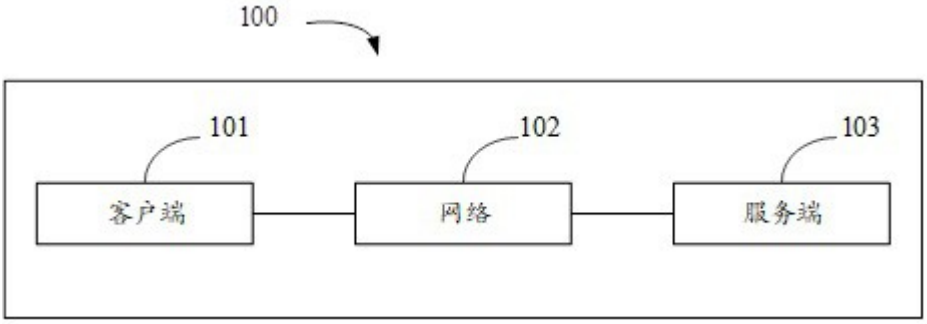

100
101
102
103
客户端
网络
服务端

图 1

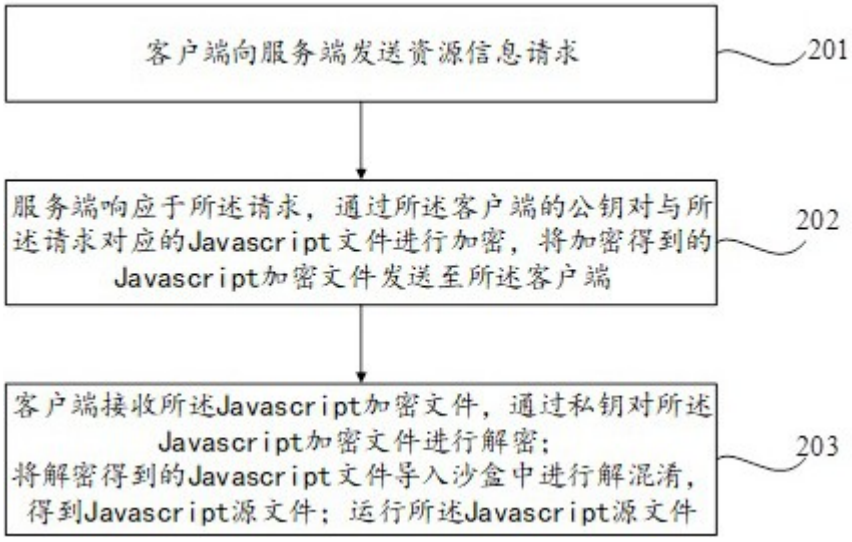

客户端向服务端发送资源信息请求
201
服务端响应于所述请求，通过所述客户端的公钥对与所述请求对应的Javascript文件进行加密，将加密得到的Javascript加密文件发送至所述客户端
202
客户端接收所述Javascript加密文件，通过私钥对所述Javascript加密文件进行解密；
将解密得到的Javascript文件导入沙盒中进行解混淆，得到Javascript源文件；运行所述Javascript源文件
203

图 2

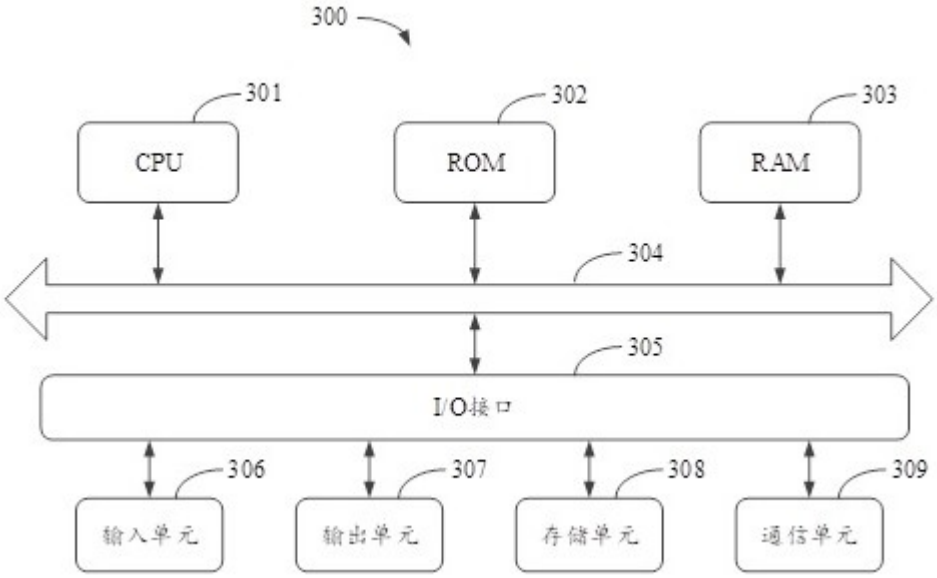

300
301
CPU
302
ROM
303
RAM
304
305
I/O接口
306
输入单元
307
输出单元
308
存储单元
309
通信单元

图 3