(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 103701761 A
(43)申请公布日 2014.04.02

(21)申请号 201210366885.8

(22)申请日 2012.09.28

(71)申请人 中国电信股份有限公司
地址 100033 北京市西城区金融大街 31 号

(72)发明人 翁颐 蒋铭勋 奚溪 姚良
仝建刚

(74)专利代理机构 中国国际贸易促进委员会专利商标事务所 11038
代理人 毛丽琴

(51)Int.Cl.
H04L 29/06(2006.01)

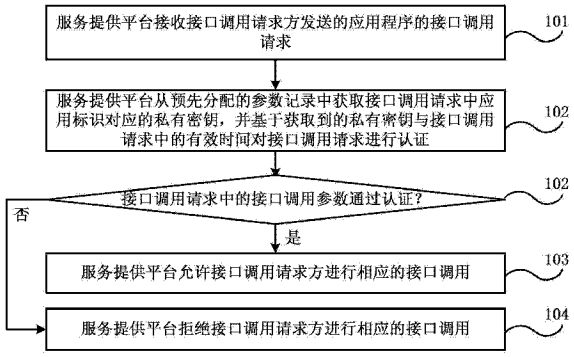
权利要求书4页 说明书11页 附图5页

(54)发明名称

开放接口调用的认证方法与系统

(57)摘要

本发明实施例公开了一种开放接口调用的认证方法与系统，其中，方法包括：服务提供平台接收服务器端或客户端发送的应用程序的接口调用请求，其中的接口调用参数包括应用标识、私有密钥与服务器端为本次接口调用分配的有效时间，私有密钥由服务器端在服务提供平台注册后由服务提供平台分配，客户端在登录服务器端并获得授权后从服务器端获取接口调用参数；从预先分配的参数记录中获取该应用标识对应的私有密钥，并基于获取到的私有密钥对接口调用请求进行认证；若通过认证，允许进行相应的接口调用；否则，拒绝接口调用请求方进行相应的接口调用。本发明实施例可以保证接口调用的安全性，接口调用流程简单，接口调用效率较高。



CN 103701761 A

1. 一种开放接口调用的认证方法，其特征在于，包括：

服务提供平台接收接口调用请求方发送的应用程序的接口调用请求，所述接口调用请求中包括接口调用参数；所述接口调用请求方包括具有紧耦合关系的服务器端或客户端；所述接口调用参数包括唯一标识一个应用程序的应用标识、服务提供平台为所述应用程序分配的私有密钥与服务器端为本次接口调用分配的有效时间；所述私有密钥由服务器端在所述服务提供平台注册后由服务提供平台为所述应用程序分配，所述客户端在登录服务器端并获得所述服务器端授权后从所述服务器端获取所述接口调用参数；

所述服务提供平台从预先分配的参数记录中获取所述应用标识对应的私有密钥，并基于获取到的私有密钥与所述有效时间对所述接口调用请求进行认证；

响应于所述接口调用请求通过认证，所述服务提供平台允许所述接口调用请求方进行相应的接口调用；

否则，响应于所述接口调用请求未通过认证，所述服务提供平台拒绝所述接口调用请求方进行相应的接口调用。

2. 根据权利要求 1 所述的方法，其特征在于，服务提供平台具体接收接口调用请求方通过统一资源标识符 URI 发送的应用程序的接口调用请求；

所述接口调用参数还包括所述 URI；

所述接口调用请求中包括接口调用参数具体为：所述接口调用请求中包括所述应用标识、有效时间与第一认证数据，所述第一认证数据由所述服务器端利用预设加密算法对所述 URI、所述私有密钥与所述有效时间进行加密得到。

3. 根据权利要求 2 所述的方法，其特征在于，所述服务提供平台从预先分配的参数记录中获取所述应用标识对应的私有密钥，并基于获取到的私有密钥与所述有效时间对所述接口调用请求进行认证包括：

所述服务提供平台从预先分配的参数记录中获取所述应用标识对应的私有密钥；

所述服务提供平台利用所述预设加密算法对发送所述接口调用请求的 URI、获取到的私有密钥与所述有效时间进行加密，得到第二认证数据；

所述服务提供平台识别所述第一认证数据与所述第二认证数据是否一致，以及当前时刻是否在所述有效时间内；

若第一认证数据与所述第二认证数据一致，且当前时刻在所述有效时间内，则所述接口调用请求通过认证；

否则，若第一认证数据与所述第二认证数据不一致，和 / 或当前时刻不在所述有效时间内，则所述接口调用请求未通过认证。

4. 根据权利要求 3 所述的方法，其特征在于，所述预设加密算法包括摘要生成算法 HMAC_SHA1。

5. 根据权利要求 3 所述的方法，其特征在于，所述接口调用请求为基于超文本传输协议 HTTP 的 HTTP 请求；

所述接口调用请求中包括所述应用标识、有效时间与第一认证数据具体为：所述 HTTP 请求的报文头部包括所述应用标识、有效时间与第一认证数据。

6. 根据权利要求 5 所述的方法，其特征在于，服务提供平台接收接口调用请求方发送的应用程序的接口调用请求之前，还包括接口调用请求方生成所述 HTTP 请求的操作。

7. 根据权利要求 6 所述的方法，其特征在于，生成所述 HTTP 请求具体包括：

获取基于 HTTP 发送应用程序的接口调用请求的 URI；

获取所述 URI 中的相对地址部分，所述相对地址部分包括第一查询字符串；

将所述第一查询字符串按照预设字典顺序重新排序，得到新的第一查询字符串；

去除新的第一查询字符串中的分割符，得到新的第一字符串；

以所述服务提供平台为所述应用程序分配的私有密钥作为预设加密算法的密钥计算服务提供平台为所述应用程序分配的有效时间，得到第一字节流数组；

对所述第一字节流数组按照内容传输编码 Base64 编码方式进行编码，得到第一接入密钥；

将所述第一接入密钥作为预设加密算法的密钥计算所述新的第一字符串，得到第一消息认证码；

对所述第一消息认证码按照 Base64 编码方式进行编码，得到第一认证数据；

将所述应用标识、所述第一认证数据与服务器端为本次接口调用分配的有效时间加入到基于 HTTP 发送应用程序的接口调用请求的 HTTP 报文的头部，得到所述 HTTP 请求。

8. 根据权利要求 7 所述的方法，其特征在于，所述服务提供平台利用所述预设加密算法对发送所述接口调用请求的 URI、获取到的私有密钥与所述有效时间进行加密，得到第二认证数据具体包括：

获取发送所述接口调用请求的 URI 中的相对地址部分与有效时间，所述相对地址部分包括第二查询字符串；

将所述第二查询字符串按照预设字典顺序重新排序，得到新的第二查询字符串；

去除新的第二查询字符串中的分割符，得到新的第二字符串；

以获取到的私有密钥作为预设加密算法的密钥计算所述 HTTP 请求中携带的有效时间，得到第二字节流数组；

对所述第二字节流数组按照 Base64 编码方式进行编码，得到第二接入密钥；

将所述第二接入密钥作为预设加密算法的密钥计算所述新的第二字符串，得到第二消息认证码；

对所述第二消息认证码按照 Base64 编码方式进行编码，得到第二认证数。

9. 根据权利要求 8 所述的方法，其特征在于，所述接口调用请求方为客户端时，生成所述 HTTP 请求之前还包括：

所述客户端登录服务器端，基于 HTTP 向所述服务器端发送接口调用的请求消息，所述请求消息中包括所述应用标识；

所述服务器端开始执行所述获取基于 HTTP 发送应用程序的接口调用请求的 URI 的操作，并在得到第一接入密钥后将所述第一接入密钥与为本次接口调用分配的有效时间发送给所述客户端；

所述客户端开始执行所述将所述第一接入密钥作为预设加密算法的密钥计算所述新的第一字符串的操作，得到所述 HTTP 请求。

10. 根据权利要求 7、8 或 9 所述的方法，其特征在于，所述分割符包括以下双引号“”中的符号之一：“.”、“&”、“\”“/”。

11. 一种开放接口调用的认证系统，包括服务提供平台与接口调用请求方，所述接口调

用请求方包括具有紧耦合关系的服务器端或客户端；其特征在于，所述接口调用请求方，用于向所述服务提供平台发送应用程序的接口调用请求，所述接口调用请求中包括接口调用参数，所述接口调用参数包括唯一标识一个应用程序的应用标识、服务提供平台为所述应用程序分配的私有密钥与服务器端为本次接口调用分配的有效时间；所述私有密钥由服务器端在所述服务提供平台注册后由服务提供平台为所述应用程序分配，所述客户端在登录服务器端并获得所述服务器端授权后从所述服务器端获取所述接口调用参数；

所述服务提供平台，用于接收接口调用请求方发送的应用程序的接口调用请求；从预先分配的参数记录中获取所述应用标识对应的私有密钥，并基于获取到的私有密钥与所述有效时间对所述接口调用请求进行认证；响应于所述接口调用请求通过认证，允许所述接口调用请求方进行相应的接口调用；否则，响应于所述接口调用请求未通过认证，拒绝所述接口调用请求方进行相应的接口调用。

12. 根据权利要求 11 所述的系统，其特征在于，所述服务提供平台具体接收接口调用请求方通过统一资源标识符 URI 发送的应用程序的接口调用请求；

所述接口调用参数还包括所述 URI；

所述接口调用请求中包括接口调用参数具体为：所述接口调用请求中包括所述应用标识、有效时间与第一认证数据，所述第一认证数据由所述服务器端利用预设加密算法对所述 URI、所述私有密钥与所述有效时间进行加密得到。

13. 根据权利要求 12 所述的系统，其特征在于，所述服务提供平台从预先分配的参数记录中获取所述应用标识对应的私有密钥，并基于获取到的私有密钥与所述有效时间对所述接口调用请求进行认证时，具体从预先分配的参数记录中获取所述应用标识对应的私有密钥；利用所述预设加密算法对发送所述接口调用请求的 URI、获取到的私有密钥与所述有效时间进行加密，得到第二认证数据；识别所述第一认证数据与所述第二认证数据是否一致，以及当前时刻是否在所述有效时间内；若第一认证数据与所述第二认证数据一致，且当前时刻在所述有效时间内，则确认所述接口调用请求通过认证；否则，若第一认证数据与所述第二认证数据不一致，和 / 或当前时刻不在所述有效时间内，则确定所述接口调用请求未通过认证。

14. 根据权利要求 13 所述的系统，其特征在于，所述预设加密算法包括摘要生成算法 HMAC_SHA1。

15. 根据权利要求 13 所述的系统，其特征在于，所述接口调用请求为基于超文本传输协议 HTTP 的 HTTP 请求；

所述接口调用请求中包括所述应用标识、有效时间与第一认证数据具体为：所述 HTTP 请求的报文头部包括所述应用标识、有效时间与第一认证数据。

16. 根据权利要求 15 所述的系统，其特征在于，所述接口调用请求方还用于生成所述 HTTP 请求。

17. 根据权利要求 16 所述的系统，其特征在于，所述接口调用请求方为服务器端；

所述服务器端生成所述 HTTP 请求时，具体获取基于 HTTP 发送应用程序的接口调用请求的 URI；获取所述 URI 中的相对地址部分，所述相对地址部分包括第一查询字符串；将所述第一查询字符串按照预设字典顺序重新排序，得到新的第一查询字符串；去除新的第一查询字符串中的分割符，得到新的第一字符串；以所述服务提供平台为所述应用程序分配

的私有密钥作为预设加密算法的密钥计算服务提供平台为所述应用程序分配的有效时间，得到第一字节流数组；对所述第一字节流数组按照内容传输编码 Base64 编码方式进行编码，得到第一接入密钥；将所述第一接入密钥作为预设加密算法的密钥计算所述新的第一字符串，得到第一消息认证码；对所述第一消息认证码按照 Base64 编码方式进行编码，得到第一认证数据；将所述应用标识、所述第一认证数据与服务器端为本次接口调用分配的有效时间加入到基于 HTTP 发送应用程序的接口调用请求的 HTTP 报文的头部，得到所述 HTTP 请求。

18. 根据权利要求 17 所述的系统，其特征在于，所述服务提供平台利用所述预设加密算法对发送所述接口调用请求的 URI、获取到的私有密钥与所述有效时间进行加密，得到第二认证数据时，具体获取发送所述接口调用请求的 URI 中的相对地址部分与有效时间，所述相对地址部分包括第二查询字符串；将所述第二查询字符串按照预设字典顺序重新排序，得到新的第二查询字符串；去除新的第二查询字符串中的分割符，得到新的第二字符串；以获取到的私有密钥作为预设加密算法的密钥计算所述 HTTP 请求中携带的有效时间，得到第二字节流数组；对所述第二字节流数组按照 Base64 编码方式进行编码，得到第二接入密钥；将所述第二接入密钥作为预设加密算法的密钥计算所述新的第二字符串，得到第二消息认证码；对所述第二消息认证码按照 Base64 编码方式进行编码，得到第二认证数。

19. 根据权利要求 18 所述的系统，其特征在于，所述接口调用请求方具体为客户端；

所述客户端，还用于在生成所述 HTTP 请求之前登录服务器端，基于 HTTP 向所述服务器端发送接口调用的请求消息，所述请求消息中包括所述应用标识；以及在接收到服务器端发送的第一接入密钥时，开始执行所述将所述第一接入密钥作为预设加密算法的密钥计算所述新的第一字符串的操作，得到所述 HTTP 请求；

所述服务器端，还用于在接收到客户端发送的应用程序的接口调用请求时，开始执行所述获取基于 HTTP 发送应用程序的接口调用请求的 URI 的操作，并在得到第一接入密钥后将所述第一接入密钥与为本次接口调用分配的接入密钥发送给所述客户端。

20. 根据权利要求 17、18 或 19 所述的系统，其特征在于，所述分割符包括以下双引号“”中的符号之一：“.”、“&”、“\”“/”。

开放接口调用的认证方法与系统

技术领域

[0001] 本发明涉及互联网技术，尤其是一种开放接口调用的认证方法与系统。

背景技术

[0002] 互联网服务提供平台(Service Platform)提供开放接口供第三方开发者在开发互联网应用程序中进行调用。这些开放接口，例如应用编程接口(REST API)，基于超文本传输协议(Hyper TextTransport Protocol，以下简称：HTTP)进行通信。服务提供平台对于所接收到的接口调用请求，应该进行有效的认证，只响应合法的接口调用请求。

[0003] 现有技术中，针对存在应用程序的服务器端 / 客户端(C/S)以及服务提供平台三方的接口调用场景中，主要通过以下两种接口调用方法进行接口调用：

[0004] 第一种方法中，由应用程序的服务器端(App Server)向服务提供平台申请接口调用所需的密码(APP key)，服务器端使用该密码向服务提供平台发送接口调用请求，以请求进行开放接口的调用。如果应用程序的客户端(APP Client)也需要进行开放接口的调用，则由客户端向服务器端请求服务，则由服务器端直接将该密码开放给客户端，由客户端自由地向服务提供平台发送接口调用请求，以请求进行开放接口的调用；

[0005] 第二种方法中，由应用程序的服务器端向服务提供平台申请接口调用所需的密码，服务器端使用该密码向服务提供平台发送接口调用请求，以请求进行开放接口的调用。如果应用程序的客户端也需要进行开放接口的调用，则由服务器端全权代理客户端每次向服务提供平台发送接口调用请求，再将获得的内容转交给客户端。

[0006] 在实现本发明的过程中，发明人发现上述现有技术的接口调用方法至少存在以下问题：

[0007] 第一种方法中，由服务器端将密码开放给客户端后，客户端可以自由地向服务提供平台请求进行接口调用，服务器端无法对客户端的调用请求进行任何管控，导致接口调用的安全性较低；

[0008] 第二种方法中，需要由服务器端全权代理客户端每次向服务提供平台发送接口调用请求，再将获得的内容转交给客户端，流程迂回复杂，接口调用效率低下，并且增加了服务器端的工作负荷，降低了服务器端的工作性能。

发明内容

[0009] 本发明实施例所要解决的技术问题是：提供一种开放接口调用的认证方法与系统，可以保证接口调用的安全性，并且，接口调用流程简单，接口调用效率较高，客户端在接口调用的过程中不增加服务器的工作负荷。

[0010] 本发明实施例提供的一种开放接口调用的认证方法，包括：

[0011] 服务提供平台接收接口调用请求方发送的应用程序的接口调用请求，所述接口调用请求中包括接口调用参数；所述接口调用请求方包括具有紧耦合关系的服务器端或客户端；所述接口调用参数包括唯一标识一个应用程序的应用标识、服务提供平台为所述应用

程序分配的私有密钥与服务器端为本次接口调用分配的有效时间；所述私有密钥由服务器端在所述服务提供平台注册后由服务提供平台为所述应用程序分配，所述客户端在登录服务器端并获得所述服务器端授权后从所述服务器端获取所述接口调用参数；

[0012] 所述服务提供平台从预先分配的参数记录中获取所述应用标识对应的私有密钥，并基于获取到的私有密钥与所述有效时间对所述接口调用请求进行认证；

[0013] 响应于所述接口调用请求通过认证，所述服务提供平台允许所述接口调用请求方进行相应的接口调用；

[0014] 否则，响应于所述接口调用请求未通过认证，所述服务提供平台拒绝所述接口调用请求方进行相应的接口调用。

[0015] 本发明实施例提供的一种开放接口调用的认证系统，包括服务提供平台与接口调用请求方，所述接口调用请求方包括具有紧耦合关系的服务器端或客户端；

[0016] 所述接口调用请求方，用于向所述服务提供平台发送应用程序的接口调用请求，所述接口调用请求中包括接口调用参数，所述接口调用参数包括唯一标识一个应用程序的应用标识、服务提供平台为所述应用程序分配的私有密钥与服务器端为本次接口调用分配的有效时间；所述私有密钥由服务器端在所述服务提供平台注册后由服务提供平台为所述应用程序分配，所述客户端在登录服务器端并获得所述服务器端授权后从所述服务器端获取所述接口调用参数；

[0017] 所述服务提供平台，用于接收接口调用请求方发送的应用程序的接口调用请求；从预先分配的参数记录中获取所述应用标识对应的私有密钥，并基于获取到的私有密钥与所述有效时间对所述接口调用请求进行认证；响应于所述接口调用请求通过认证，允许所述接口调用请求方进行相应的接口调用；否则，响应于所述接口调用请求未通过认证，拒绝所述接口调用请求方进行相应的接口调用。

[0018] 基于本发明上述实施例提供的开放接口调用的认证方法与系统，服务器端在服务提供平台注册后服务提供平台可以为应用程序分配接口调用参数，客户端在登录服务器端并获得服务器端授权后从服务器端获取该接口调用参数，包括唯一标识一个应用程序的应用标识（AppID）、服务提供平台为该应用程序分配的私有密钥（APPKEY）与服务器端为本次接口调用分配的有效时间（Service-Expires），客户端或服务器端向服务提供平台请求接口调用时，向服务提供平台发送接口调用请求，提供应用标识 AppID、私有密钥 APPKEY 与有效时间 Service-Expires，服务提供平台从预先分配的参数记录中获取接口调用请求中应用标识 AppID 对应的私有密钥 APPKEY，并基于该获取到的私有密钥 APPKEY 与接口调用请求中的有效时间对接口调用请求进行认证，只有在接口调用请求通过认证时，服务提供平台才允许客户端进行相应的接口调用，否则，服务提供平台拒绝客户端进行相应的接口调用。由于客户端只有在登录服务器端并获得服务器端授权后才能从服务器端获取进行接口调用所需的接口调用参数，包括私有密钥 APPKEY 与有效时间 Service-Expires，从而通过服务提供平台的认证后获得对所请求服务的响应，为服务端提供了对客户端接口调用的一定程度管控，与现有技术相比，提高了接口调用的安全性；另外，客户端只有在登录服务器端并获得服务器端授权后才能从服务器端获取进行接口调用所需的接口调用参数后，即可独立地向服务提供平台进行接口调用，无需由服务器端全权代理客户端每次向服务提供平台发送接口调用请求，再将获得的内容转交给客户端，相对于现有技术，本发明实施例为客户

端提供了一种相对独立的接口调用过程，接口调用流程中认证流程简单有效，接口调用效率较高，客户端在接口调用的过程中不增加服务器的工作负荷。本发明可以适用于任意具有服务端 / 客户端结构且具有紧耦合关系的互联网应用程序调用的场景。

[0019] 下面通过附图和实施例，对本发明的技术方案做进一步的详细描述。

附图说明

[0020] 为了更清楚地说明本发明实施例或现有技术中的技术方案，下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本发明的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动性的前提下，还可以根据这些附图获得其他的附图。

[0021] 图 1 为本发明开放接口调用的认证方法一个实施例的流程图。

[0022] 图 2 为本发明开放接口调用的认证方法另一个实施例的流程图。

[0023] 图 3 为本发明开放接口调用的认证方法又一个实施例的流程图。

[0024] 图 4 为本发明开放接口调用的认证方法再一个实施例的流程图。

[0025] 图 5 为本发明开放接口调用的认证系统一个实施例的结构示意图。

具体实施方式

[0026] 下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，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所有其他实施例，都属于本发明保护的范围。

[0027] 本发明实施例针对服务提供平台提供开放接口供具有 C/S 结构且具有紧耦合关系的互联网应用程序调用的环境下，且接口调用基于 HTTP 进行通信，设计一种有效的接口调用认证方法，主要体现在 :1）认证流程的设计简单有效，所需的计算开销小 ;2）适用于具有 C/S 结构的应用，且服务器端控制管理客户端，只有当客户端登录服务器端后，在获取服务器端授权后才有可能调用互联网服务提供平台的开放接口获取相应服务。本发明实施例适用于具有服务端 / 客户端结构的互联网应用程序要求通过服务器端直接调用接口及从客户端调用接口的场景。

[0028] 本发明实施例中，互联网服务提供平台开放基于 HTTP 的接口，第三方所开发的互联网应用程序具有 C/S 结构且具有紧耦合关系，即 :客户端与服务器端可通过内部自定义的通信协议进行安全有效的通信。其中的紧耦合关系是指，客户端与服务器端之间是紧密结合的，某应用程序的服务器端仅向某应用程序的客户端提供服务，并且服务器端对客户端有一定的约束管控能力。

[0029] 其中的互联网服务提供平台，提供开放接口，例如存储服务的开放接口，供开发者调用。服务器端为应用程序的服务端系统，保存所有与客户端应用程序相关的信息，使用服务提供平台提供的服务。客户端为应用程序的客户端，与服务器端具有紧耦合的关系，一般为 C/S 架构，可定义自己的内部通信协议。

[0030] 图 1 为本发明开放接口调用的认证方法一个实施例的流程图。如图 1 所示，该实施例开放接口调用的认证方法包括 :

[0031] 101，服务提供平台接收接口调用请求方发送的应用程序的接口调用请求，该接口调用请求中包括接口调用参数。其中的接口调用参数包括唯一标识一个应用程序的应用标识 AppID、服务提供平台为该应用程序分配的私有密钥 APPKEY 与服务器端为本次接口调用分配的有效时间 Service-Expires。

[0032] 本发明实施例中的接口调用请求方包括服务器端或客户端，服务器端与客户端具有紧耦合关系，一般为 C/S 结构，可以定义自己的内容通信协议。其中的私有密钥 APPKEY 由服务器端在服务提供平台注册后由服务提供平台为该应用程序分配，客户端在登录服务器端并获得服务器端授权后可以从服务器端获取该接口调用参数。

[0033] 102，服务提供平台从预先分配的参数记录中获取接口调用请求中应用标识 AppID 对应的私有密钥 APPKEY，并基于获取到的私有密钥 APPKEY 与接口调用请求中的有效时间，对接口调用请求进行认证。

[0034] 响应于接口调用请求通过认证，执行 103 的操作。否则，响应于接口调用请求中未通过认证，执行 104 的操作。

[0035] 103，服务提供平台允许接口调用请求方进行相应的接口调用。

[0036] 之后，不再执行本实施例的后续操作。

[0037] 104，服务提供平台拒绝接口调用请求方进行相应的接口调用。

[0038] 本发明上述实施例提供的开放接口调用的认证方法，服务器端在服务提供平台注册后服务提供平台可以为应用程序分配接口调用参数，客户端在登录服务器端并获得服务器端授权后从服务器端获取该接口调用参数，客户端或服务器端向服务提供平台请求接口调用时，向服务提供平台发送接口调用请求，提供应用标识、私有密钥与服务器端为本次接口调用分配的有效时间，服务提供平台从预先分配的参数记录中获取接口调用请求中应用标识对应的私有密钥，并基于该该获取到的私有密钥与接口调用请求中的有效时间对接口调用请求进行认证，只有在接口调用请求通过认证时，服务提供平台才允许客户端进行相应的接口调用，否则，服务提供平台拒绝客户端进行相应的接口调用。由于客户端只有在登录服务器端并获得服务器端授权后才能从服务器端获取进行接口调用所需的接口调用参数，包括私有密钥与有效时间，从而通过服务提供平台的认证后获得对所请求服务的响应，为服务端提供了对客户端接口调用的一定程度管控，提高了接口调用的安全性；另外，客户端只有在登录服务器端并获得服务器端授权后才能从服务器端获取进行接口调用所需的接口调用参数后，即可独立地向服务提供平台进行接口调用，无需由服务器端全权代理客户端每次向服务提供平台发送接口调用请求，再将获得的内容转交给客户端，为客户端提供了一种相对独立的接口调用过程，接口调用流程中认证流程简单有效，接口调用效率较高，客户端在接口调用的过程中不增加服务器的工作负荷。本发明可以适用于任意具有服务端 / 客户端结构且具有紧耦合关系的互联网应用程序调用的场景。

[0039] 根据本发明开放接口调用的认证方法的一个示例而非限制，服务提供平台具体接收接口调用请求方通过统一资源标识符(UniformResource Identifier，以下简称：URI)发送的应用程序的接口调用请求。相应地，接口调用请求中的接口调用参数还可以包括 URI。接口调用请求中包括接口调用参数具体可以是：接口调用请求中包括应用标识 AppID 有效时间 Service-Expires 与第一认证数据 Service-Auth，该第一认证数据 Service-Auth 由服务器端利用预设加密算法对发送接口调用请求的 URI、服务提供平台为该应用程序分配的

私有密钥 APPKEY 与服务器端为本次接口调用分配的有效时间 Service-Expires 进行加密得到。

[0040] 示例性地，其中的预设加密算法具体可以包括但不限于摘要生成算法 HMAC_SHA1。本发明实施例中，采用的加密算法 Hmac-Sha1 的计算开销小，使得开放接口调用的认证流程所需的计算开销小。

[0041] 图 2 为本发明开放接口调用的认证方法另一个实施例的流程图。如图 2 所示，该实施例开放接口调用的认证方法包括：

[0042] 201，服务提供平台接收接口调用请求方发送的应用程序的接口调用请求，该接口调用请求中包括接口调用参数。其中的接口调用参数包括请求调用的应用程序的应用标识 AppID、服务提供平台为该应用程序分配的私有密钥 APPKEY 与服务器端为本次接口调用分配的有效时间 Service-Expires。服务提供平台为每个应用程序生成私有密钥 APPKEY 后，可以在后台数据库中更新预先分配的参数记录，包括每个 AppId 及其对应的私有密钥 AppKey，其中的私有密钥 AppKey 唯一且不公开。

[0043] 本发明实施例中的接口调用请求方包括具有紧耦合关系的服务器端或客户端。其中，服务器端在服务提供平台注册后，可以在相应的注册帐号下申请服务，由服务器端为该应用程序分配私有密钥 APPKEY，客户端只有在登录服务器端并获得服务器端授权后才可以从服务器端获取该私有密钥 APPKEY 以及服务器端为本次接口调用分配的有效时间 Service-Expires，以直接向服务提供平台请求服务，具体可以使用 HTTP 协议进行通信。

[0044] 202，服务提供平台从预先分配的参数记录中获取接口调用请求中应用标识 AppID 对应的 APPKEY。

[0045] 203，服务提供平台利用预设加密算法对发送接口调用请求的 URI、基于接口调用请求中应用标识 AppID 从预先分配的参数记录中获取到的 APPKEY 与有效时间 Service-Expires 进行加密，得到第二认证数据 Service-Auth。

[0046] 204，服务提供平台识别第一认证数据与第二认证数据是否一致，以及当前时刻是否在接口调用请求中的有效时间 Service-Expires 内。

[0047] 若第一认证数据与第二认证数据一致，且当前时刻在接口调用请求中的有效时间 Service-Expires 内，则接口调用请求通过认证，执行 205 的操作。

[0048] 否则，若第一认证数据与第二认证数据不一致，和 / 或当前时刻不在接口调用请求中的有效时间 Service-Expires 内，则接口调用请求未通过认证，执行 206 的操作。

[0049] 其中的有效时间 Service-Expires，可以被服务器端用来管控客户端能自由请求接口调用的时间长度。服务提供平台收到接口调用请求后，将当前时刻与接口调用请求中的有效时间比较，如果当前时刻晚于有效时间规定的最后时刻，则认为第一认证数据过期，接口调用请求未通过认证，无法继续使用相应的服务。

[0050] 如果服务器端当前的系统时间与服务提供平台的系统时间不一致，可以设定以服务提供平台的系统时间为准，同时过期时间的判定以服务提供平台在接收到接口调用请求时的当前系统时刻为准。

[0051] 205，服务提供平台允许接口调用请求方进行相应的接口调用。

[0052] 之后，不再执行本实施例的后续操作。

[0053] 基于 204 ～ 205 的操作，服务提供平台确认接口调用请求所请求的资源具有合法

性、时间在有效期内，才允许接口调用请求方进行相应的接口调用，即使互联网中其他人截取到该接口调用请求包括的第一认证数据并复制利用，也只能在短暂的有效期内可以使用，超过有效期无法通过认证，从而提高了认证的安全性与有效性。

[0054] 206，服务提供平台拒绝接口调用请求方进行相应的接口调用。

[0055] 根据本发明开放接口调用的认证方法的另一个具体示例而非限制，接口调用请求方发送的接口调用请求具体可以为基于 HTTP 的 HTTP 请求。相应地，该接口调用请求中包括应用标识 AppID、有效时间 Service-Expires 与第一认证数据 Service-Auth 具体为：在 HTTP 请求的报文头部包括应用标识 AppID、有效时间 Service-Expires 与第一认证数据 Service-Auth。

[0056] 示例性地，在本发明上述实施例中，服务提供平台接收接口调用请求方发送的应用程序的接口调用请求之前，可以先由接口调用请求方生成 HTTP 请求。

[0057] 图 3 为本发明开放接口调用的认证方法又一个实施例的流程图。如图 3 所示，该实施例中，接口调用请求方具体可以通过如下方式生成 HTTP 请求：

[0058] 301，获取基于 HTTP 发送应用程序的接口调用请求的 URI，例如，http://116.228.171.53/Storage/api/File?p=backup/data.txt&appid=storage。

[0059] 302，获取 URI 中的相对地址部分，该相对地址部分包括第一查询字符串。

[0060] 其中，URI 中的相对地址部分，例如可以是 URI 中以 /api/ 开头的部分，即上述 URI 实例中的 api/File?p=backup/data.txt&appid=storage 部分。查询字符串，例如可以是问号(?)之后的键值对构成的字符串，即上述 URI 实例中的 p=backup/data.txt&appid=storage 部分。

[0061] 303，将第一查询字符串按照预设字典顺序重新排序，得到新的第一查询字符串。

[0062] 预设字典顺序，例如依据字典中的顺序，字母 a 在 b 前，数字 1 在 2 前。按照预设字典顺序重新排序，例如，依据字段顺序 k1=v1 在 k2=v2 之前，则重新排序将 k2=v2&k1=v1 变为 k1=v1&k2=v2。

[0063] 304，去除新的第一查询字符串中的分割符，得到新的第一字符串。

[0064] 示例性地，分割符具体可以预先设定，包括但不限于以下双引号“”中的符号之一：“.”、“&”、“\”“/”。

[0065] 305，以服务提供平台为应用程序分配的私有密钥 APPKEY 作为预设加密算法的密钥，计算服务器端为本次接口调用分配的有效时间 Service-Expires，得到第一字节流数组。

[0066] 其中，有效时间 Service-Expires 的格式可以是 yyyy-MM-ddHH:mm:ss，例如，2011-09-1217:39:26。

[0067] 306，对第一字节流数组按照内容传输编码(Base64)编码方式进行编码，得到第一接入密钥(AccessKey)。

[0068] 307，将第一接入密钥 AccessKey 作为预设加密算法的密钥计算新的第一字符串，得到第一消息认证码。

[0069] 308，对第一消息认证码按照 Base64 编码方式进行编码，得到第一认证数据 Service-Auth。

[0070] 309，将应用标识 AppID、第一认证数据 Service-Auth 与服务器端为该应用程序分

配的有效时间 Service-Expires 加入到基于 HTTP 发送应用程序的接口调用请求的 HTTP 报文的头部，得到 HTTP 请求。

[0071] 根据本发明实施例的又一个具体示例而非限制，图 3 所示的实施例中，服务器端可以直接按照图 3 所示实施例的流程生成 HTTP 请求。接口调用请求方为客户端时，客户端可以先登录服务器端，基于 HTTP 向服务器端发送应用程序的接口调用请求，该请求中包括请求调用的应用程序的应用标识 AppID。服务器端接收到客户端发送的该接口调用请求后，可以为该应用程序分配的有效时间并执行图 3 所示实施例中 301 ～ 306 的操作，并在得到第一接入密钥后将该第一接入密钥 AccessKey 与为本次接口调用分配的有效时间 Service-Expires 发送给客户端，之后，由客户端执行 307 ～ 309 的操作，最终生成 HTTP 请求。

[0072] 以下以一个具体的应用实例对图 3 所示的实施例进行进一步说明。

[0073] 假设服务提供平台的连接点为 http://116.228.171.53/Storage，在服务器端以 appid=storage 为参数调用接口 /api/File 获取文件 backup/data.txt。在服务器端以 appid=storage 为参数，通过如下 URI 向服务提供平台提交接口调用请求：

[0074] GET

[0075] http://116.228.171.53/Storage/api/File?p=backup/data.txt&appid=storage

[0076] 基于上述图 3 所示实施例，先将 URI 中的 http://116.228.171.53/Storage/ 部分去掉，变成

[0077] GETapi/File?p=backup/data.txt&appid=storage，然后将问号（?）之后的第一查询字符串 p=backup/data.txt&appid=storage 以分割符（&）分割为两个字符串，按照预设字典顺序排列，将 appid=storage 排列在 p=backup/data.txt 之前，变成

[0078] GETapi/File?appid=storage&p=backup/data.txt，最后去掉其中的分隔符（/.&），得到新的第一字符串如下：

[0079] GETapiFile?appid=storagep=backupdatatxt

[0080] 假设 AppKey=6ffGhwi2pN+UdeK2k1FCgoBeYH4=，ServiceExpires=2011-09-1217:39:26，则经过 305 与 306 的操作，进一步可得到第一 AccessKey 为：

[0081] NxEksznFzdLJhnmzHs6fZz2Btng=

[0082] 用第一 AccessKey 作为 Hmac-Sha1 算法的密钥，计算新的第一字符串 GETapiFile?appid=storagep=backupdatatxt，得到第一消息认证码，进行 Base64 编码后得到如下所示的第一 Service Auth：

[0083] VYUfvxE6tiC4JSPyczxQXokVORE=

[0084] 则最后在进行应用编程接口（API）调用时，服务器端向服务提供平台发送的 HTTP 请求如下：

[0085] GET

[0086] http://116.228.171.53/Storage/api/File?p=backup/data.txt&appid=storageHTTP/1.1

[0087] Content-Type:application/octet-stream

[0088] Service-Expires:2011-09-1217:39:26

[0089] Service-Auth:VYUfvxE6tiC4JSPyczxQXokVORE=

[0090] 上述 HTTP 请求中，Content-Type 表示发送给服务提供平台的具体数据类型，供服务提供平台进行相应处理，application/octet-stream 表示发送的是二进制流。HTTP/1.1 表示支持的 HTTP 版本信息，与其中的 Content-Type 一样是 HTTP 请求的报文头部的常规内容，构建 HTTP 请求时已经生成。

[0091] 仍然假设服务提供平台的连接点为 http://116.228.171.53/Storage，在客户端以 appid=storage 为参数调用接口 /api/File 获取文件 backup/data.txt。客户端在登录服务器端后，向服务器端发送接口调用的请求消息，请求在有效时间 Service Expires 内对服务提供平台进行接口调用，该请求消息中包括应用标识 AppID。

[0092] 服务器端向服务提供平台为应用标识 AppID 申请私有密钥 APPKEY 后，可以自行维护保管应用标识 AppID 与相应的私有密钥 APPKEY。服务器端根据客户端发送的请求消息中的应用表示 AppID 获取对应的私有密钥 AppKey，然后图 3 所示实施例中 301 ～ 306 的操作计算出第一接入密钥 AccessKey 并返回给客户端。客户端在收到第一接入密钥 AccessKey 后，执行 307 ～ 309 的操作，最终生成 HTTP 请求，之后可以采取与服务器端直接向服务提供平台进行接口调用的过程一样，向服务提供平台请求相应的服务。

[0093] 图 4 为本发明开放接口调用的认证方法再一个实施例的流程图。如图 4 所示，与图 3 所示实施例相应地，在该实施例中，图 2 所示实施例的操作 203 中，服务提供平台利用预设加密算法对发送接口调用请求的 URI、应用标识对应的私有密钥与有效时间进行加密，得到第二认证数据的操作具体可以通过如下方式实现：

[0094] 401，获取发送接口调用请求的 URI 中的相对地址部分与有效时间 Service-Expires，该相对地址部分包括第二查询字符串。

[0095] 402，将第二查询字符串按照预设字典顺序重新排序，得到新的第二查询字符串。

[0096] 403，去除新的第二查询字符串中的分割符，得到新的第二字符串。

[0097] 示例性地，分割符具体可以预先设定，包括但不限于以下双引号“”中的符号之一：“.”、“&”、“\”“/”。

[0098] 404，以基于接口调用请求中应用标识 AppID 从预先分配的参数记录中获取到的私有密钥 APPKEY 作为预设加密算法的密钥，计算接口调用请求方发送的 HTTP 请求中携带的有效时间 Service-Expires，得到第二字节流数组。

[0099] 405，对第二字节流数组按照 Base64 编码方式进行编码，得到第二接入密钥 AccessKey。

[0100] 406，将第二接入密钥 AccessKey 作为预设加密算法的密钥计算新的第二字符串，得到第二消息认证码。

[0101] 407，对第二消息认证码按照 Base64 编码方式进行编码，得到第二认证数据 Service-Auth。

[0102] 之后，便可以通过本发明上述图 2 所示实施例流程中的操作 204 ～ 206，由服务提供平台对接口调用请求方进行接口调用控制。

[0103] 图 5 为本发明开放接口调用的认证系统一个实施例的结构示意图。该实施例的开放接口调用的认证系统可用于实现本发明上述各开放接口调用的认证方法实施例的流程。如图 5 所示，其包括服务提供平台 1 与接口调用请求方，该接口调用请求方包括具有紧耦合

关系的服务器端 2 或客户端 3。服务器端 2 与客户端 3 具有紧耦合关系，一般为 C/S 结构，可以定义自己的内容通信协议。

[0104] 其中，接口调用请求方，用于向服务提供平台 1 发送应用程序的接口调用请求，该接口调用请求中包括接口调用参数，接口调用参数包括唯一标识一个应用程序的应用标识、服务提供平台 1 为该应用程序分配的私有密钥与服务器端为本次接口调用分配的有效时间。

[0105] 其中，私有密钥由服务器端 2 在服务提供平台 1 注册后由服务提供平台 1 为该应用程序分配，客户端 3 在登录服务器端 2 并获得服务器端 2 授权后从服务器端 2 获取该接口调用参数。

[0106] 服务提供平台 1，用于接收接口调用请求方发送的应用程序的接口调用请求；从预先分配的参数记录中获取接口调用请求中应用标识对应的接口调用参数，并基于该获取到的私有密钥与接口调用请求中的有效时间对接口调用请求进行认证；响应于接口调用请求通过认证，允许接口调用请求方进行相应的接口调用；否则，响应于接口调用请求未通过认证，拒绝接口调用请求方进行相应的接口调用。

[0107] 本发明上述实施例提供的开放接口调用的认证系统，服务器端在服务提供平台注册后服务提供平台可以为应用程序分配接口调用参数，客户端在登录服务器端并获得服务器端授权后从服务器端获取该接口调用参数，客户端或服务器端向服务提供平台请求接口调用时，向服务提供平台发送接口调用请求，提供应用标识、私有密钥与服务器端为本次接口调用分配的有效时间，服务提供平台从预先分配的参数记录中获取接口调用请求中应用标识对应的私有密钥，并基于该该获取到的私有密钥与接口调用请求中的有效时间对接口调用请求进行认证，只有在接口调用请求通过认证时，服务提供平台才允许客户端进行相应的接口调用，否则，服务提供平台拒绝客户端进行相应的接口调用。由于客户端只有在登录服务器端并获得服务器端授权后才能从服务器端获取进行接口调用所需的接口调用参数，包括私有密钥与有效时间，从而通过服务提供平台的认证后获得对所请求服务的响应，为服务端提供了对客户端接口调用的一定程度管控，提高了接口调用的安全性；另外，客户端只有在登录服务器端并获得服务器端授权后才能从服务器端获取进行接口调用所需的接口调用参数后，即可独立地向服务提供平台进行接口调用，无需由服务器端全权代理客户端每次向服务提供平台发送接口调用请求，再将获得的内容转交给客户端，为客户端提供了一种相对独立的接口调用过程，接口调用流程中认证流程简单有效，接口调用效率较高，客户端在接口调用的过程中不增加服务器的工作负荷。本发明可以适用于任意具有服务端 / 客户端结构且具有紧耦合关系的互联网应用程序调用的场景。

[0108] 根据本发明开放接口调用的认证系统的一个具体示例而非限制，服务提供平台 1 具体接收接口调用请求方通过统一资源标识符 URI 发送的应用程序的接口调用请求。相应地，接口调用参数还包括 URI。接口调用请求中包括接口调用参数具体为：接口调用请求中包括应用标识、有效时间与第一认证数据，该第一认证数据由服务器端 2 利用预设加密算法对发送接口调用请求的 URI、服务提供平台为该应用程序分配的私有密钥 APPKEY 与服务器端为本次接口调用分配的有效时间 Service-Expires 进行加密得到。

[0109] 示例性地，服务提供平台 1 从预先分配的参数记录中接口调用请求中应用标识对应的私有密钥，并基于该获取到的私有密钥对接口调用请求进行认证时，具体可以从预先

分配的参数记录中获取接口调用请求中应用标识对应的私有密钥；利用预设加密算法对发送接口调用请求的URI、获取到的私有密钥与有效时间进行加密，得到第二认证数据；识别第一认证数据与第二认证数据是否一致，以及当前时刻是否在有效时间内；若第一认证数据与第二认证数据一致，且当前时刻在有效时间内，则确认接口调用请求通过认证；否则，若第一认证数据与第二认证数据不一致，和/或当前时刻不在有效时间内，则确定接口调用请求未通过认证。

[0110] 示例性地，其中的预设加密算法具体可以包括但不限于摘要生成算法HMAC_SHA1。

[0111] 根据本发明开放接口调用的认证系统的一个具体示例而非限制，接口调用请求方发送的接口调用请求为基于HTTP的HTTP请求。该接口调用请求中包括应用标识、有效时间与第一认证数据具体为：HTTP请求的报文头部包括应用标识、有效时间与第一认证数据。

[0112] 示例性地，接口调用请求方还用于生成HTTP请求。

[0113] 根据本发明开放接口调用的认证系统的一个具体示例而非限制，接口调用请求方为服务器端2。服务器端2生成HTTP请求时，具体可以通过以下方式：

[0114] 获取基于HTTP发送应用程序的接口调用请求的URI；获取URI中的相对地址部分，该相对地址部分包括第一查询字符串；将第一查询字符串按照预设字典顺序重新排序，得到新的第一查询字符串；去除新的第一查询字符串中的分割符，得到新的第一字符串；以服务提供平台1为应用程序分配的私有密钥作为预设加密算法的密钥计算服务提供平台1为应用程序分配的有效时间，得到第一字节流数组；对第一字节流数组按照Base64编码方式进行编码，得到第一接入密钥；将第一接入密钥作为预设加密算法的密钥计算新的第一字符串，得到第一消息认证码；对第一消息认证码按照Base64编码方式进行编码，得到第一认证数据；将应用标识、第一认证数据与服务器端2为本次接口调用分配的有效时间加入到基于HTTP发送应用程序的接口调用请求的HTTP报文的头部，得到HTTP请求。

[0115] 与上述具体示例中服务器端2生成HTTP请求相应地，服务提供平台1利用预设加密算法对发送接口调用请求的URI、获取到的私有密钥与服务器端2为本次接口调用分配的有效时间进行加密，得到第二认证数据时，具体可以通过如下方式：

[0116] 获取发送接口调用请求的URI中的相对地址部分与有效时间，该相对地址部分包括第二查询字符串；将第二查询字符串按照预设字典顺序重新排序，得到新的第二查询字符串；去除新的第二查询字符串中的分割符，得到新的第二字符串；以基于接口调用请求中应用标识获取到的私有密钥作为预设加密算法的密钥计算HTTP请求中携带的有效时间，得到第二字节流数组；对第二字节流数组按照Base64编码方式进行编码，得到第二接入密钥；将第二接入密钥作为预设加密算法的密钥计算新的第二字符串，得到第二消息认证码；对第二消息认证码按照Base64编码方式进行编码，得到第二认证数。

[0117] 示例性地，上述分割符具体可以预先设定，包括但不限于以下双引号“”中的符号之一：“.”、“&”、“\”“/”。

[0118] 根据本发明开放接口调用的认证系统的另一个具体示例而非限制，接口调用请求方具体为客户端3。相应地，客户端3，还用于在生成HTTP请求之前登录服务器端2，基于HTTP向服务器端2发送接口调用的请求消息，该请求消息中包括应用标识；以及在接收到服务器端2发送的第一接入密钥时，开始执行将第一接入密钥作为预设加密算法的密钥计算新的第一字符串的操作，得到HTTP请求。

[0119] 服务器端 2，还用于在接收到客户端 3 发送的应用程序的接口调用请求时，开始执行获取基于 HTTP 发送应用程序的接口调用请求的 URI 的操作，并在得到第一接入密钥后将该第一接入密钥与为本次接口调用分配的接入密钥发送给客户端 3。

[0120] 本说明书中各个实施例均采用递进的方式描述，每个实施例重点说明的都是与其它实施例的不同之处，各个实施例之间相同或相似的部分相互参见即可。对于系统实施例而言，由于其与方法实施例基本相似，所以描述的比较简单，相关之处参见方法实施例的部分说明即可。

[0121] 可能以许多方式来实现本发明的方法和系统。例如，可通过软件、硬件、固件或者软件、硬件、固件的任何组合来实现本发明的方法和系统。用于所述方法的步骤的上述顺序仅是为了进行说明，本发明的方法的步骤不限于以上具体描述的顺序，除非以其它方式特别说明。此外，在一些实施例中，还可将本发明实施为记录在记录介质中的程序，这些程序包括用于实现根据本发明的方法的机器可读指令。因而，本发明还覆盖存储用于执行根据本发明的方法的程序的记录介质。

[0122] 本领域普通技术人员可以理解 :实现上述方法实施例的全部或部分步骤可以通过程序指令相关的硬件来完成，前述的程序可以存储于一计算机可读取存储介质中，该程序在执行时，执行包括上述方法实施例的步骤 ;而前述的存储介质包括 :ROM、RAM、磁碟或者光盘等各种可以存储程序代码的介质。

[0123] 本发明实施例中，客户端只有在登录服务器端并获得服务器端授权后才能从服务器端获取进行接口调用所需的接口调用参数，从而通过服务提供平台的认证后获得对所请求服务的响应，为服务端提供了对客户端接口调用的一定程度管控，提高了接口调用的安全性 ;另外，客户端只有在登录服务器端并获得服务器端授权后才能从服务器端获取进行接口调用所需的接口调用参数后，即可独立地向服务提供平台进行接口调用，无需由服务器端全权代理客户端每次向服务提供平台发送接口调用请求，再将获得的内容转交给客户端，为客户端提供了一种相对独立的接口调用过程，接口调用流程中认证流程简单有效，接口调用效率较高，客户端在接口调用的过程中不增加服务器的工作负荷。本发明可以适用于任意具有服务端 / 客户端结构且具有紧耦合关系的互联网应用程序调用的场景。

[0124] 本发明的描述是为了示例和描述起见而给出的，而并不是无遗漏的或者将本发明限于所公开的形式。很多修改和变化对于本领域的普通技术人员而言是显然的。选择和描述实施例是为了更好说明本发明的原理和实际应用，并且使本领域的普通技术人员能够理解本发明从而设计适于特定用途的带有各种修改的各种实施例。

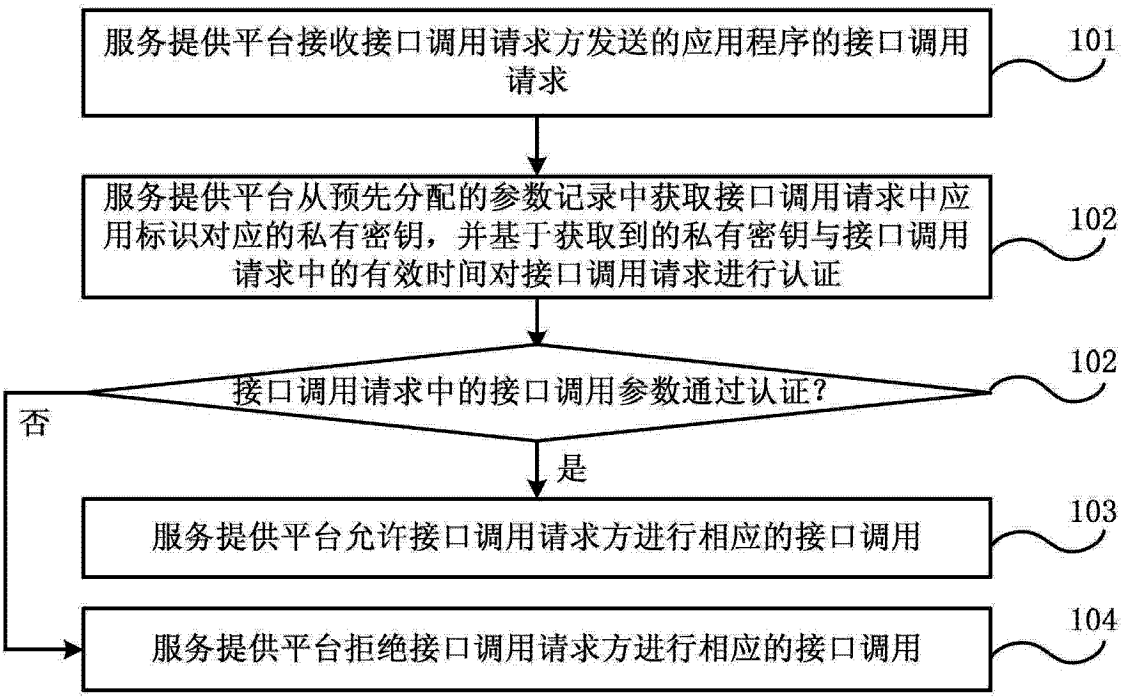

服务提供平台接收接口调用请求方发送的应用程序的接口调用请求
101
服务提供平台从预先分配的参数记录中获取接口调用请求中应用标识对应的私有密钥，并基于获取到的私有密钥与接口调用请求中的有效时间对接口调用请求进行认证
102
接口调用请求中的接口调用参数通过认证？
102
否
是
服务提供平台允许接口调用请求方进行相应的接口调用
103
服务提供平台拒绝接口调用请求方进行相应的接口调用
104

图 1

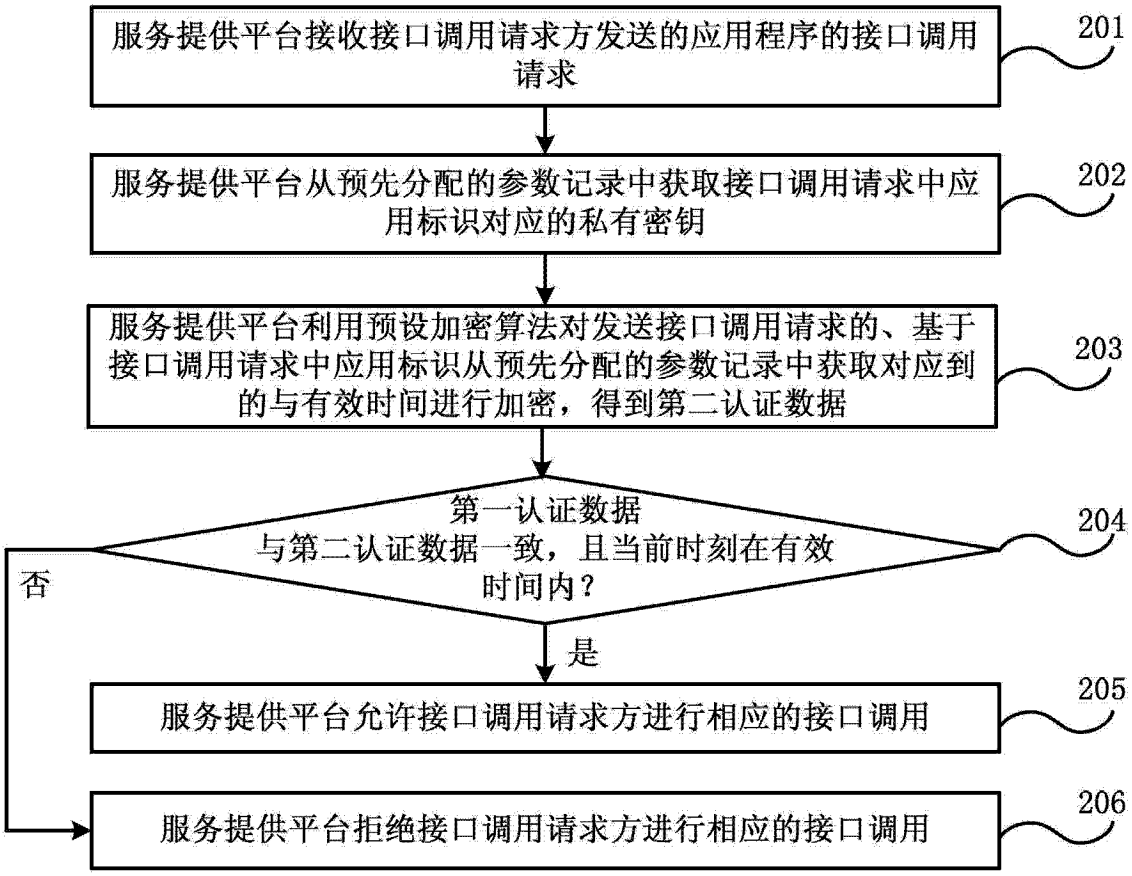

服务提供平台接收接口调用请求方发送的应用程序的接口调用请求
201
服务提供平台从预先分配的参数记录中获取接口调用请求中应用标识对应的私有密钥
202
服务提供平台利用预设加密算法对发送接口调用请求的、基于接口调用请求中应用标识从预先分配的参数记录中获取对应到的与有效时间进行加密，得到第二认证数据
203
第一认证数据与第二认证数据一致，且当前时刻在有效时间内？
204
否
是
服务提供平台允许接口调用请求方进行相应的接口调用
205
服务提供平台拒绝接口调用请求方进行相应的接口调用
206

图 2

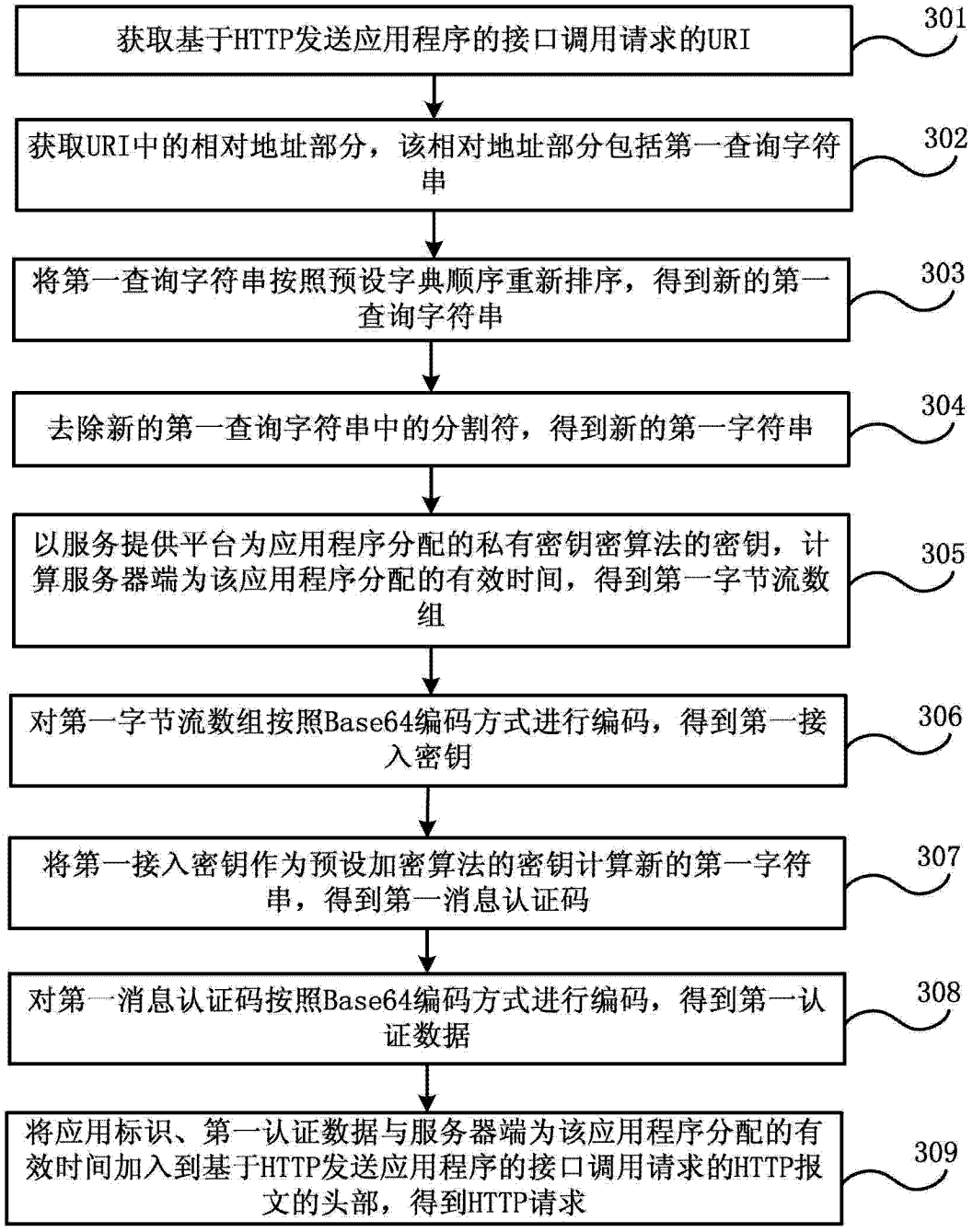

获取基于HTTP发送应用程序的接口调用请求的URI
301
获取URI中的相对地址部分，该相对地址部分包括第一查询字符串
302
将第一查询字符串按照预设字典顺序重新排序，得到新的第一查询字符串
303
去除新的第一查询字符串中的分割符，得到新的第一字符串
304
以服务提供平台为应用程序分配的私有密钥密算法的密钥，计算服务器端为该应用程序分配的有效时间，得到第一字节流数组
305
对第一字节流数组按照Base64编码方式进行编码，得到第一接入密钥
306
将第一接入密钥作为预设加密算法的密钥计算新的第一字符串，得到第一消息认证码
307
对第一消息认证码按照Base64编码方式进行编码，得到第一认证数据
308
将应用标识、第一认证数据与服务器端为该应用程序分配的有效时间加入到基于HTTP发送应用程序的接口调用请求的HTTP报文的头部，得到HTTP请求
309

图 3

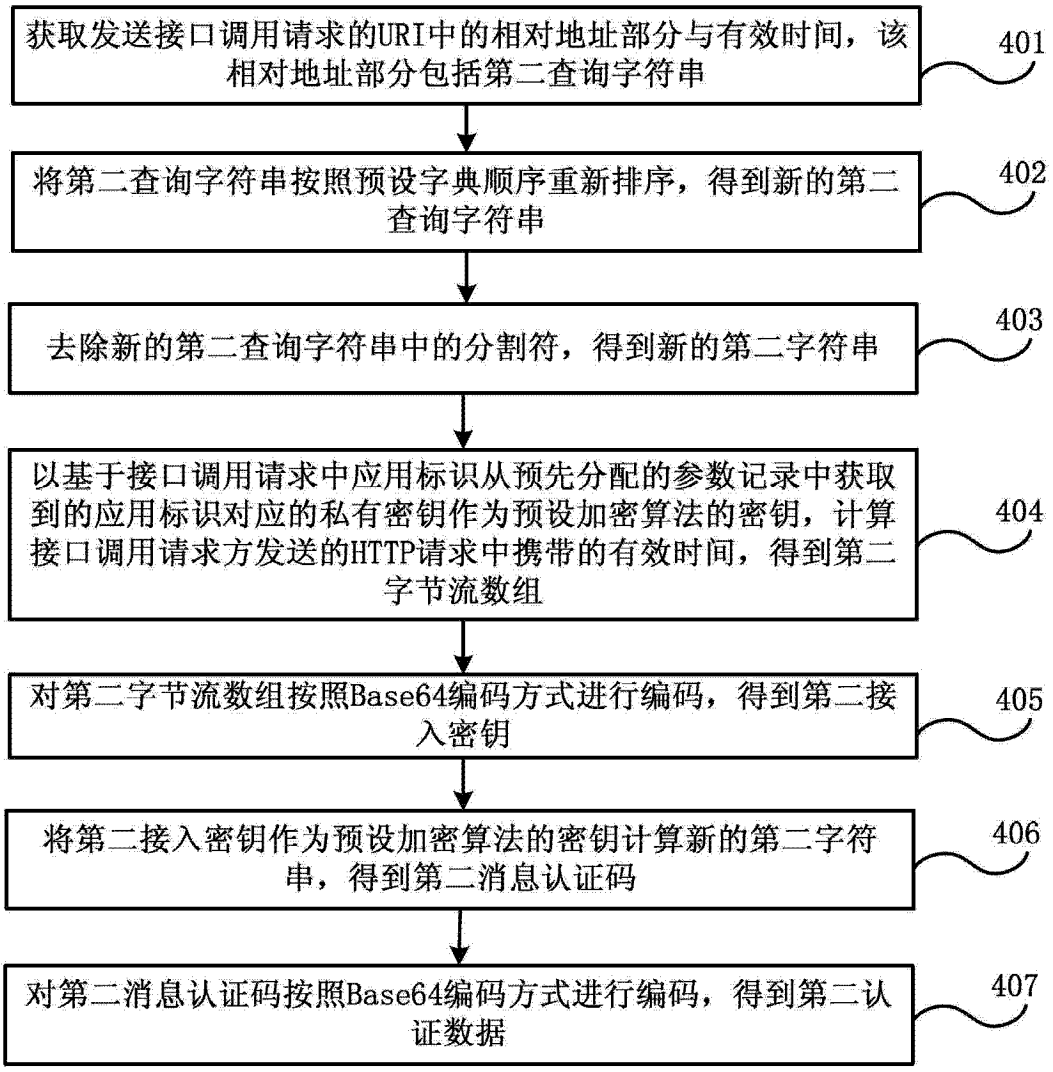

获取发送接口调用请求的URI中的相对地址部分与有效时间，该相对地址部分包括第二查询字符串
401
将第二查询字符串按照预设字典顺序重新排序，得到新的第二查询字符串
402
去除新的第二查询字符串中的分割符，得到新的第二字符串
403
以基于接口调用请求中应用标识从预先分配的参数记录中获取到的应用标识对应的私有密钥作为预设加密算法的密钥，计算接口调用请求方发送的HTTP请求中携带的有效时间，得到第二字节流数组
404
对第二字节流数组按照Base64编码方式进行编码，得到第二接入密钥
405
将第二接入密钥作为预设加密算法的密钥计算新的第二字符串，得到第二消息认证码
406
对第二消息认证码按照Base64编码方式进行编码，得到第二认证数据
407

图 4

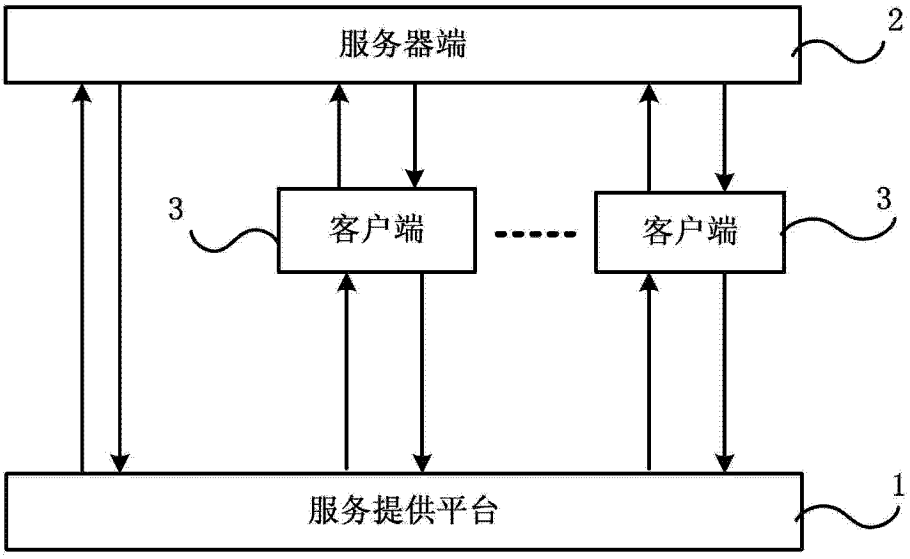


图 5