(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 109302425 A
(43)申请公布日 2019.02.01

(21)申请号 201811434173.9

(22)申请日 2018.11.28

(71)申请人 河北省科学院应用数学研究所
地址 050081 河北省石家庄市桥西区友谊南大街46号1号楼

(72)发明人 黄世中 黎彤亮 李晓云 赵环宇 王怀瑞 慕晓蕾

(74)专利代理机构 石家庄国为知识产权事务所 13120
代理人 郝伟

(51)Int.Cl.
H04L 29/06(2006.01)
H04L 29/08(2006.01)
H04L 29/12(2006.01)

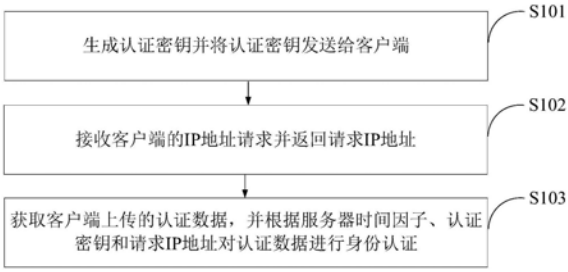
权利要求书2页 说明书8页 附图3页

(54)发明名称

身份认证方法及终端设备

(57)摘要

本发明提供了一种身份认证方法及终端设备，该方法应用于服务器和客户端，应用于服务器端的方法包括：生成认证密钥并将认证密钥发送给客户端；返回请求IP地址；获取客户端上传的认证数据进行身份认证。应用于客户端的方法包括：获取服务器生成的认证密钥；接收服务器返回的请求IP地址；生成认证数据；在资源请求的请求参数中添加认证参数和认证数据；发送资源请求至服务器进行身份认证。本发明提供的身份认证方法及终端设备快速高效，易于实施。



CN 109302425 A

1.一种身份认证方法,其特征在于,所述方法应用于服务器,所述方法包括:

生成认证密钥并将所述认证密钥发送给客户端,所述认证密钥用于指示客户端生成认证数据;

接收客户端的IP地址请求并返回请求IP地址,所述请求IP地址为所述客户端的IP地址,用于指示客户端生成所述认证数据;

获取客户端上传的认证数据,并根据服务器时间因子、所述认证密钥和所述请求IP地址对所述认证数据进行身份认证;所述认证数据为客户端根据客户端时间因子、所述认证密钥和所述请求IP地址生成。

2.如权利要求1所述的身份认证方法,其特征在于,在获取客户端上传的认证数据之前,还包括:

接收客户端的资源请求,若所述资源请求为IP地址请求,则所述允许客户端进行资源访问;

若所述资源请求的请求参数中包含认证参数,则获取客户端上传的认证数据并对所述认证数据进行身份认证。

3.如权利要求2所述的身份认证方法,其特征在于,所述认证参数包括预设名称参数和预设属性参数,所述资源请求的请求参数中包含认证参数,包括:

所述请求参数的URL参数中包含所述预设名称参数;

或所述请求参数的HTTP头中的属性名称包含所述预设属性参数。

4.如权利要求1所述的身份认证方法,其特征在于,所述根据服务器时间因子、所述认证密钥和所述请求IP地址对所述认证数据进行身份认证,包括:

根据服务器时间因子、所述认证密钥和所述请求IP地址确定身份认证值;

若所述身份认证值等于所述认证数据中的认证请求值,则确定身份认证成功。

5.如权利要求4所述的身份认证方法,其特征在于,所述服务器时间因子包括第一时间因子、第二时间因子和第三时间因子,所述根据服务器时间因子、所述认证密钥和所述请求IP地址对所述认证数据进行身份认证,包括:

根据多个时间因子、所述认证密钥和所述请求IP地址确定多个身份认证值;

若所述多个身份认证值中存在一个身份认证值等于所述认证数据中的认证请求值,则确定身份认证成功。

6.一种身份认证方法,其特征在于,所述方法应用于客户端,所述方法包括:

获取服务器生成的认证密钥;

发送IP地址请求至服务器并接收服务器返回的请求IP地址,所述请求IP地址为服务器接收到的IP地址请求的IP地址;

根据客户端时间因子、所述认证密钥和所述请求IP地址生成认证数据,所述认证数据包括根据所述客户端时间因子、所述认证密钥、所述请求IP地址确定的认证请求值和所述请求IP地址;

在资源请求的请求参数中添加认证参数和所述认证数据;所述认证参数用于指示服务器判断是否获取认证数据,所述认证数据用于指示服务器根据服务器时间因子、所述认证密钥和所述请求IP地址对所述认证数据进行身份认证;

发送所述资源请求至服务器进行身份认证。

7.如权利要求6所述的身份认证方法，其特征在于，所述认证参数包括预设名称参数和预设属性参数，所述在资源请求的资源请求参数中添加认证参数，包括：

在所述请求参数的URL参数添加预设名称参数；

或在所述请求参数的HTTP头中的属性名称中添加预设属性参数。

8.如权利要求4至7中任意一项所述的身份认证方法，其特征在于，所述认证请求值或所述身份认证值为：

pSign＝Truncate(S)

其中，pSign为认证请求值或身份认证值，Truncate()为截位函数，S为预设函数；

所述预设函数为：

S＝F(K,ID)

其中，K为认证密钥，ID为杂凑算法、HMAC认证算法或分组密码函数的输入信息，F()为杂凑算法、HMAC认证算法或分组密码函数；

所述输入信息为：

ID＝{T|IP}

其中，T为时间因子，IP为请求IP地址，“|”为连接符；

所述时间因子为：

$T＝T_0÷T_c$

其中，T_0为以UTC时间为计量标准的8字节整数，T_c为以秒为单位的认证变化周期。

9.一种终端设备，包括存储器、处理器以及存储在所述存储器中并可在所述处理器上运行的计算机程序，其特征在于，所述处理器执行所述计算机程序时实现如权利要求1至8任一项所述方法的步骤。

10.一种计算机可读存储介质，所述计算机可读存储介质存储有计算机程序，其特征在于，所述计算机程序被处理器执行时实现如权利要求1至8任一项所述方法的步骤。

身份认证方法及终端设备

技术领域

[0001] 本发明属于网络安全技术领域，更具体地说，是涉及一种身份认证方法及终端设备。

背景技术

[0002] REST(Representational State Transfer，表述性状态转移)是一种新的互联网应用架构。REST充分利用HTTP的优势，以资源为核心，将资源添加数据、读取数据、修改数据、删除数据的操作映射为HTTP的GET、PUT、POST、DELTE等方法。REST式的web服务提供了统一的接口和资源定位，简化了Web服务接口的设计和实现，降低了web服务的复杂度。

[0003] 但是，易于识别的和理解的REST接口也存在着易于破解的危险。如果破坏者通过对Web资源地址的猜解，获得某一资源的接口，就很容易造成敏感资源的外泄和系统数据的破坏。因此REST接口的安全性至关重要。由于REST是无状态的传输，所以客户端的每一次请求都需携带身份认证信息。现有技术中的身份认证的方式有HTTP Basic，HTTP Digest，API KEY，Oauth，JWK等。但前述认证方式或需要用户输入用户名和口令，或需要建立CA中心和使用数字证书，或对其他系统有很强依赖性，使用上均不够快捷和高效。

发明内容

[0004] 本发明的目的在于提供一种身份认证方法及终端设备，以解决现有技术中存在的REST接口身份认证不够快捷高效的技术问题。

[0005] 本发明实施例的第一方面，提供了一种身份认证方法，所述方法应用于服务器，所述方法包括：

[0006] 生成认证密钥并将所述认证密钥发送给客户端，所述认证密钥用于指示客户端生成认证数据；

[0007] 接收客户端的IP地址请求并返回请求IP地址，所述请求IP地址为所述客户端的IP地址，用于指示客户端生成所述认证数据；

[0008] 获取客户端上传的认证数据，并根据服务器时间因子、所述认证密钥和所述请求IP地址对所述认证数据进行身份认证；所述认证数据为客户端根据客户端时间因子、所述认证密钥和所述请求IP地址生成。

[0009] 本发明实施例的第二方面，提供了一种身份认证方法，所述方法应用于客户端，所述方法包括：

[0010] 获取服务器生成的认证密钥；

[0011] 发送IP地址请求至服务器并接收服务器返回的请求IP地址，所述请求IP地址为服务器接收到的IP地址请求的IP地址；

[0012] 根据客户端时间因子、所述认证密钥和所述请求IP地址生成认证数据，所述认证数据包括根据所述客户端时间因子、所述认证密钥、所述请求IP地址确定的认证请求值和所述请求IP地址；

[0013] 在资源请求的请求参数中添加认证参数和所述认证数据;所述认证参数用于指示服务器判断是否获取认证数据,所述认证数据用于指示服务器根据服务器时间因子、所述认证密钥和所述请求IP地址对所述认证数据进行身份认证;

[0014] 发送所述资源请求至服务器进行身份认证。

[0015] 本发明实施例的第三方面,提供了一种终端设备,包括存储器、处理器以及存储在所述存储器中并可在所述处理器上运行的计算机程序,所述处理器执行所述计算机程序时实现上述的身份认证方法的步骤。

[0016] 本发明实施例的第四方面,提供了一种计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,所述计算机程序被处理器执行时实现上述的身份认证方法的步骤。

[0017] 本发明提供的身份认证方法及终端设备的有益效果在于:本发明提供的身份认证方法及终端设备一方面加入了客户端IP认证因子,可以有效地防止外部网络或内部网络其它机器对系统资源的截取攻击。另一方面其采用普通的http协议,不需要安装使用数字证书。服务器端的认证密钥保存到配置文件中,不依赖于数据库系统,身份认证的安全性通过主机的安全来保障。认证时也无需用户输入用户名和口令,部署方便灵活,快捷高效,易于使用。

附图说明

[0018] 为了更清楚地说明本发明实施例中的技术方案,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的前提下,还可以根据这些附图获得其他的附图。

[0019] 图1为本发明一实施例提供的身份认证方法的流程示意图;

[0020] 图2为本发明另一实施例提供的身份认证方法的流程示意图;

[0021] 图3为本发明再一实施例提供的身份认证方法的流程示意图;

[0022] 图4为本发明又一实施例提供的身份认证方法的流程示意图;

[0023] 图5为本发明又一实施例提供的身份认证方法的流程示意图;

[0024] 图6为本发明一实施例提供的终端设备的示意框图。

具体实施方式

[0025] 为了使本发明所要解决的技术问题、技术方案及有益效果更加清楚明白,以下结合附图及实施例,对本发明进行进一步详细说明。应当理解,此处所描述的具体实施例仅仅用以解释本发明,并不用于限定本发明。

[0026] 请参考图1,为本发明一实施例提供的身份认证方法的流程示意图。该方法应用于服务器,包括:

[0027] S101:生成认证密钥并将认证密钥发送给客户端。

[0028] 在本实施例中,服务器会生成长度为128比特的认证密钥并通过加密传输的方式分发给合法客户端,该认证密钥用于指示客户端生成认证数据,每个客户端都持有不同的认证密钥以进行身份的认证。

[0029] S102:接收客户端的IP地址请求并返回请求IP地址。

[0030] 在本实施例中,请求IP地址为客户端的IP地址,用于指示客户端生成认证数据。本实施例中的认证因子,即客户端IP,是客户端从服务器端获取的资源数据,而不是客户端提供给服务器端的。这就避免了在使用局域网、代理服务器等复杂网络情况下,客户端获取自己外部IP地址不准确的问题,也减少了认证数据的传递量。

[0031] 其中,在进行IP地址资源的获取时,可采用分组加密算法在服务器端对IP地址资源进行加密,客户端获取资源后再进行解密处理。具体实现方法为:由服务器在IP地址资源后补充0x80,至少补充一个字节数据,直到其长度达到分组密码的分组长度的整数倍。在客户端进行解密处理时,若解密后的IP地址资源未包含前述补充信息,则确定资源被篡改。本发明实施例提供的IP地址资源的加密传输方法可保证该资源在网络传输的安全性,防止被篡改攻击和伪造攻击。

[0032] S103:获取客户端上传的认证数据,并根据服务器时间因子、认证密钥和请求IP地址对认证数据进行身份认证。

[0033] 在本实施例中认证数据为客户端根据客户端时间因子、认证密钥和请求IP地址生成。在进行身份认证时,服务器会根据自己的时间因子(即服务器时间因子)、认证数据中所携带的请求IP地址对应的认证密钥(即分发给客户端的认证密钥)和请求IP地址生成身份验证值,若此身份验证值等于认证数据中的认证请求值,则验证成功。

[0034] 从上述描述可知,本发明实施例提供的身份认证方法一方面加入了客户端IP认证因子,可以有效地防止外部网络或内部网络其它机器对系统资源的截取攻击。另一方面其采用普通的http协议,不需要安装使用数字证书。服务器端的认证密钥保存到配置文件中,不依赖于数据库系统,身份认证的安全性通过主机的安全来保障。认证时也无需用户输入用户名和口令,部署方便灵活,快捷高效,易于使用。

[0035] 请一并参考图1及图2,图2为本申请另一实施例提供的身份认证方法的流程示意图。在上述实施例的基础上,该方法还包括:

[0036] S201:接收客户端的资源请求,若资源请求为IP地址请求,则允许客户端进行资源访问。

[0037] 在本实施例中,服务器建立一个资源请求过滤器,对客户端所请求的资源进行检查,若客户端请求的资源为客户端本身的IP地址,即资源请求为IP地址请求,则服务器允许客户端进行资源访问并返回客户端的IP地址,即请求IP地址。

[0038] S202:若资源请求的请求参数中包含认证参数,则获取客户端上传的认证数据并对认证数据进行身份认证。

[0039] 在本实施例中,若资源请求的请求参数中包含有认证参数,则服务器继续进行认证数据的检查工作,即进行身份认证。若资源请求既不属于IP地址请求,请求参数中也未包含认证参数,则服务器会拒绝客户端对资源的请求,返回未授权访问的消息。

[0040] 可选地,作为本发明实施例提供的身份认证方法的一种具体实施方式,认证参数包括预设名称参数和预设属性参数,资源请求的请求参数中包含认证参数,包括:

[0041] 请求参数的URL参数中包含预设名称参数。

[0042] 或请求参数的HTTP头中的属性名称包含预设属性参数。

[0043] 在本实施例中,请求参数中的认证参数包括但不限于以上两种:即预设名称参数

和预设属性参数。在进行请求参数的判断时，若资源过滤器发现请求参数中的URL参数含有预设名称参数或请求参数的HTTP头中属性名称包含预设属性参数，则获取客户端上传的认证数据并对认证数据进行身份认证。

[0044] 从上述描述可知，本发明实施例提供的身份认证方法同时支持两种认证参数的上传方式，具有广泛的应用场景和适用范围。

[0045] 请一并参考图1及图3，图3为本发明再一实施例提供的身份认证方法的流程示意图，在上述实施例的基础上，步骤S103详述为：

[0046] S301：根据服务器时间因子、认证密钥和请求IP地址确定身份认证值。

[0047] 在本实施例中，认证密钥即为服务器根据客户端的IP请求地址所生成的认证密钥，为服务器持有。

[0048] S302：若身份认证值等于认证数据中的认证请求值，则确定身份认证成功。

[0049] 在本实施例中，身份认证值是根据服务器时间因子、认证密钥和请求IP地址所确定，此认证密钥为服务器持有。认证请求值是根据客户端时间因子、认证密钥和请求IP地址所确定，此认证密钥为客户端持有。

[0050] 请一并参考图1及图4，图4为本申请又一实施例提供的身份认证方法的流程示意图。服务器时间因子包括第一时间因子、第二时间因子和第三时间因子，在上述实施例的基础上，步骤S103还可详述为：

[0051] S401：根据多个时间因子、认证密钥和请求IP地址确定多个身份认证值。

[0052] 在本实施例中，若第一时间因子为T，则第二时间因子为T+1，第三时间因子为T-1，根据第一时间因子、第二时间因子和第三时间因子可分别确定第一身份认证值、第二身份认证值和第三身份认证值。

[0053] S402：若多个身份认证值中存在一个身份认证值等于认证数据中的认证请求值，则确定身份认证成功。

[0054] 在本实施例中，若第一身份认证值、第二身份认证值和第三身份认证值中存在一个认证值等于认证请求值，则身份认证成功，返回客户端请求的资源，否则认证失败，拒绝客户端的资源请求并返回未授权访问的消息。

[0055] 从上述描述可知，服务器端进行认证数据检查时，同时取了与自己当前时间因子T相邻的两个时间因子T－1、T+1进行认证数据检查，这就既允许客户端和服务器端有一定的时钟误差，又有效避免了重放攻击。

[0056] 请参考图5，图5为本申请又一实施例提供的身份认证方法的流程示意图。该方法应用于客户端，包括：

[0057] S501：获取服务器生成的认证密钥。

[0058] 在本实施例中，客户端会获取服务器生成的认证密钥，在获取过程中，为保证数据安全性，可在服务器端使用分组密码算法对数据进行加密，客户端获取资源后再进行解密处理。

[0059] S502：发送IP地址请求至服务器并接收服务器返回的请求IP地址。

[0060] 在本实施例中，请求IP地址为服务器接收到的IP地址请求的IP地址，即客户端本身的IP地址。在进行资源请求时，可使用HTTP GET的方法，使用“getRemoteIP”命令进行资源获取。相应地，可在服务器端使用分组密码算法对资源“getRemoteIP”进行加密，在客户

端进行解密处理。若解密后的资源“getRemoteIP”中未包含加密所用信息，则确定该资源被篡改。

[0061] S503：根据客户端时间因子、认证密钥和请求IP地址生成认证数据。

[0062] 在本实施例中，认证数据包括根据客户端时间因子、认证密钥、请求IP地址确定的认证请求值和请求IP地址。

[0063] S504：在资源请求的请求参数中添加认证参数和认证数据。

[0064] 在本实施例中，认证参数用于指示服务器判断是否获取认证数据，认证数据用于指示服务器根据服务器时间因子、认证密钥和请求IP地址对认证数据进行身份认证。

[0065] S505：发送资源请求至服务器进行身份认证。

[0066] 在本实施例中，服务器主要根据资源请求参数中的认证参数和认证数据进行资源认证。

[0067] 从上述描述可知，本发明实施例提供的身份认证方法一方面加入了客户端IP认证因子，可以有效地防止外部网络或内部网络其它机器对系统资源的截取攻击。另一方面其采用普通的http协议，不需要安装使用数字证书。服务器端的认证密钥保存到配置文件中，不依赖于数据库系统，身份认证的安全性通过主机的安全来保障。认证时也无需用户输入用户名和口令，部署方便灵活，快捷高效，易于使用。

[0068] 可选地，作为本发明实施例提供的身份认证方法的一种具体实施方式，认证参数包括预设名称参数和预设属性参数，在资源请求的资源请求参数中添加认证参数，包括：

[0069] 在请求参数的URL参数添加预设名称参数。

[0070] 或在请求参数的HTTP头中的属性名称中添加预设属性参数。

[0071] 在本实施例中，请求其它资源时，可以通过在HTTP头中添加自定义属性(例如属性名称为预设名称参数，属性值为客户端计算的身份认证值)的方式来传递认证结果，到服务器进行身份合法性验证，也可以通过在网址URL参数中添加预设名称参数的方式来传递认证结果。

[0072] 可选地，作为本发明实施例提供的身份认证方法的一种具体实施方式，认证请求值或身份认证值为：

[0073] pSign＝Truncate(S)

[0074] 其中，pSign为认证请求值或身份认证值，Truncate()为截位函数，S为预设函数。pSign长度为128比特，若Truncate()的输入超过128比特，则仅取其左侧的128比特值作为输出值，否则输出值和输入值相同。

[0075] 预设函数为：

[0076] S＝F(K,ID)

[0077] 其中，K为认证密钥，长度为128比特，只有认证双方持有，ID为杂凑算法、HMAC认证算法或分组密码函数的输入信息，F()为杂凑算法、HMAC认证算法或分组密码函数。其中，F()包括但不限于杂凑算法中的MD5、国密算法SM3，HMAC认证算法，分组加密算法中的TEA、国密算法SM4等。

[0078] 输入信息为：

[0079] ID＝{T|IP}

[0080] 其中，T为时间因子，IP为请求IP地址，即服务器端获取到的客户端访问IP地址，为

4个字节的整数，“|”为连接符，用于将两组数据按照左右顺序进行拼接。

[0081] 时间因子为：

[0082] T＝T0÷Tc

[0083] 其中，T0为以UTC时间(即当前距格林尼治标准时间1970年1月1日00:00的秒数)为计量标准的8字节整数，Tc为以秒为单位的认证变化周期，最大长度为60秒。

[0084] 在本实施例中，在S＝F(K,ID)环节，当使用杂凑算法时，K|ID为输入参数。

[0085] 当使用HMAC认证算法时，K表示密钥，ID表示输入的内容，HMAC认证函数由密码杂凑算法按照以下定义的公式来设计实现：

[0086] HMAC(K,ID)＝Hash(K⊕opad|Hash(K⊕ipad|ID))

[0087] 其中opad和ipad为两个B比特的常数。B表示杂凑函数Hash中处理块的大小，例如对于SM3密码杂凑函数，该值为512比特，opad可以用0x5c重复64次，ipad可以用0x36重复64次。Hash表示指定的密码杂凑函数。如果K的长度小于opad的长度，则在其后面添加0，使其达到opad的比特长度。公式中的运算顺序是先进行异或操作，再进行字符串的拼接操作。

[0088] 当使用分组密码算法时，设分组长度为m字节，K表示密钥，ID表示输入的明文，如果ID长度小于16字节或其长度不是m的整数倍数时，在其后面补充0x80，直至其长度不小于16字节且长度为m的整数倍。

[0089] 从上述描述可知，本发明实施例提供的身份认证值和认证请求值的计算方法简洁灵活，易于软硬件实现，客户端可将认证密钥存入USB Key等硬件介质中，并在硬件介质内部实现认证算法，从而进一步增加系统的安全性。

[0090] 参见图6，图6为本发明一实施例提供的终端设备的示意框图。如图6所示的本实施例中的终端600可以包括：一个或多个处理器601、一个或多个输入设备602、一个或多个输出设备603及一个或多个存储器604。上述处理器601、输入设备602、则输出设备603及存储器604通过通信总线605完成相互间的通信。存储器604用于存储计算机程序，计算机程序包括程序指令。处理器601用于执行存储器604存储的程序指令。

[0091] 应当理解，在本发明实施例中，所称处理器601可以是中央处理单元(Central Processing Unit,CPU)，该处理器还可以是其他通用处理器、数字信号处理器(Digital Signal Processor,DSP)、专用集成电路(Application Specific Integrated Circuit,ASIC)、现成可编程门阵列(Field-Programmable Gate Array,FPGA)或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。

[0092] 输入设备602可以包括触控板、指纹采传感器(用于采集用户的指纹信息和指纹的方向信息)、麦克风等，输出设备603可以包括显示器(LCD等)、扬声器等。

[0093] 该存储器604可以包括只读存储器和随机存取存储器，并向处理器601提供指令和数据。存储器604的一部分还可以包括非易失性随机存取存储器。例如，存储器604还可以存储设备类型的信息。

[0094] 具体实现中，本发明实施例中所描述的处理器601、输入设备602、输出设备603可执行本发明实施例提供的身份认证方法的第一实施例和第二实施例中所描述的实现方式，也可执行本发明实施例所描述的终端的实现方式，在此不再赘述。

[0095] 在本发明的另一实施例中提供一种计算机可读存储介质，计算机可读存储介质存

储有计算机程序，计算机程序包括程序指令，程序指令被处理器执行时实现上述实施例方法中的全部或部分流程，也可以通过计算机程序来指令相关的硬件来完成，计算机程序可存储于一计算机可读存储介质中，该计算机程序在被处理器执行时，可实现上述各个方法实施例的步骤。其中，计算机程序包括计算机程序代码，计算机程序代码可以为源代码形式、对象代码形式、可执行文件或某些中间形式等。计算机可读介质可以包括：能够携带计算机程序代码的任何实体或装置、记录介质、U盘、移动硬盘、磁碟、光盘、计算机存储器、只读存储器(ROM，Read-Only Memory)、随机存取存储器(RAM，Random Access Memory)、电载波信号、电信信号以及软件分发介质等。需要说明的是，计算机可读介质包含的内容可以根据司法管辖区内立法和专利实践的要求进行适当的增减，例如在某些司法管辖区，根据立法和专利实践，计算机可读介质不包括是电载波信号和电信信号。

[0096] 计算机可读存储介质可以是前述任一实施例的终端的内部存储单元，例如终端的硬盘或内存。计算机可读存储介质也可以是终端的外部存储设备，例如终端上配备的插接式硬盘，智能存储卡(Smart Media Card，SMC)，安全数字(Secure Digital，SD)卡，闪存卡(Flash Card)等。进一步地，计算机可读存储介质还可以既包括终端的内部存储单元也包括外部存储设备。计算机可读存储介质用于存储计算机程序及终端所需的其他程序和数据。计算机可读存储介质还可以用于暂时地存储已经输出或者将要输出的数据。

[0097] 本领域普通技术人员可以意识到，结合本文中所公开的实施例描述的各示例的单元及算法步骤，能够以电子硬件、计算机软件或者二者的结合来实现，为了清楚地说明硬件和软件的可互换性，在上述说明中已经按照功能一般性地描述了各示例的组成及步骤。这些功能究竟以硬件还是软件方式来执行，取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能，但是这种实现不应认为超出本发明的范围。

[0098] 所属领域的技术人员可以清楚地了解到，为了描述的方便和简洁，上述描述的终端和单元的具体工作过程，可以参考前述方法实施例中的对应过程，在此不再赘述。

[0099] 在本申请所提供的几个实施例中，应该理解到，所揭露的终端和方法，可以通过其它的方式实现。例如，以上所描述的装置实施例仅仅是示意性的，例如，单元的划分，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式，例如多个单元或组件可以结合或者可以集成到另一个系统，或一些特征可以忽略，或不执行。另外，所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口、装置或单元的间接耦合或通信连接，也可以是电的，机械的或其它的形式连接。

[0100] 作为分离部件说明的单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本发明实施例方案的目的。

[0101] 另外，在本发明各个实施例中的各功能单元可以集成在一个处理单元中，也可以是各个单元单独物理存在，也可以是两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现，也可以采用软件功能单元的形式实现。

[0102] 以上，仅为本发明的具体实施方式，但本发明的保护范围并不局限于此，任何熟悉本技术领域的技术人员在本发明揭露的技术范围内，可轻易想到各种等效的修改或替换，

这些修改或替换都应涵盖在本发明的保护范围之内。因此，本发明的保护范围应以权利要求的保护范围为准。

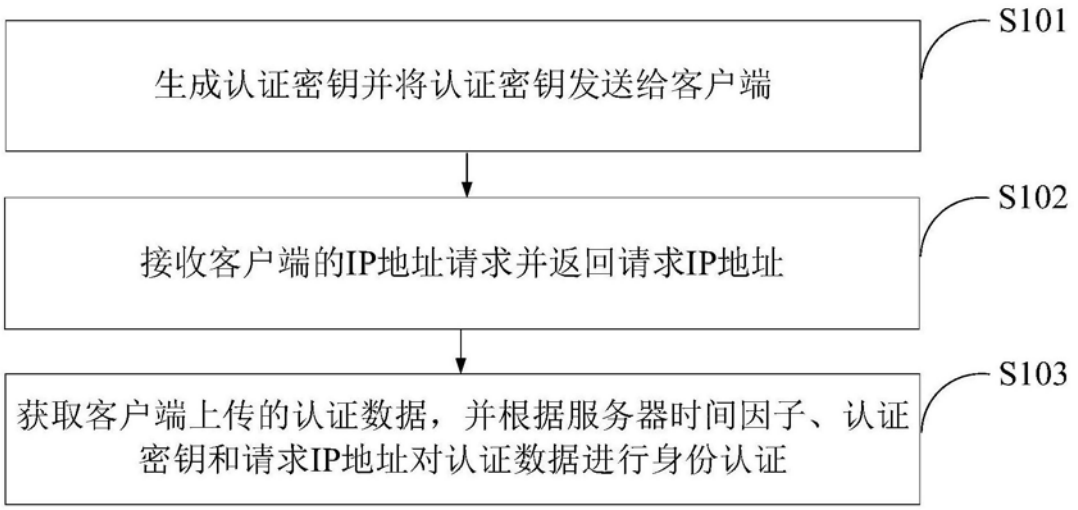

S101
生成认证密钥并将认证密钥发送给客户端
S102
接收客户端的IP地址请求并返回请求IP地址
S103
获取客户端上传的认证数据，并根据服务器时间因子、认证密钥和请求IP地址对认证数据进行身份认证

图1

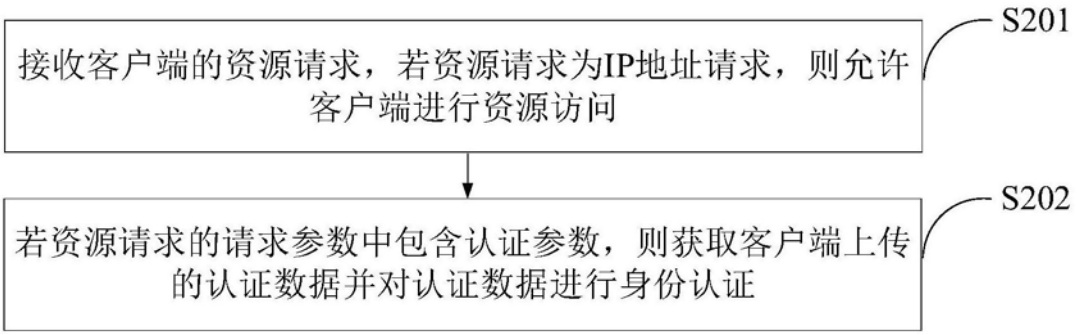

S201
接收客户端的资源请求，若资源请求为IP地址请求，则允许客户端进行资源访问
S202
若资源请求的请求参数中包含认证参数，则获取客户端上传的认证数据并对认证数据进行身份认证

图2

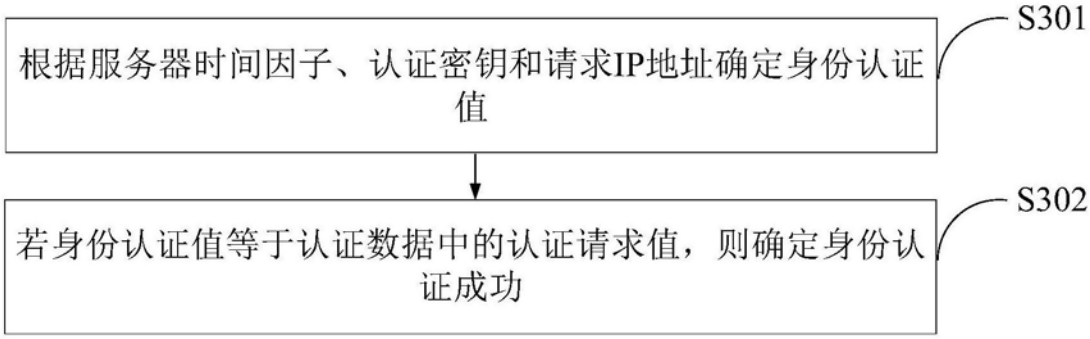

S301
根据服务器时间因子、认证密钥和请求IP地址确定身份认证值
S302
若身份认证值等于认证数据中的认证请求值，则确定身份认证成功

图3

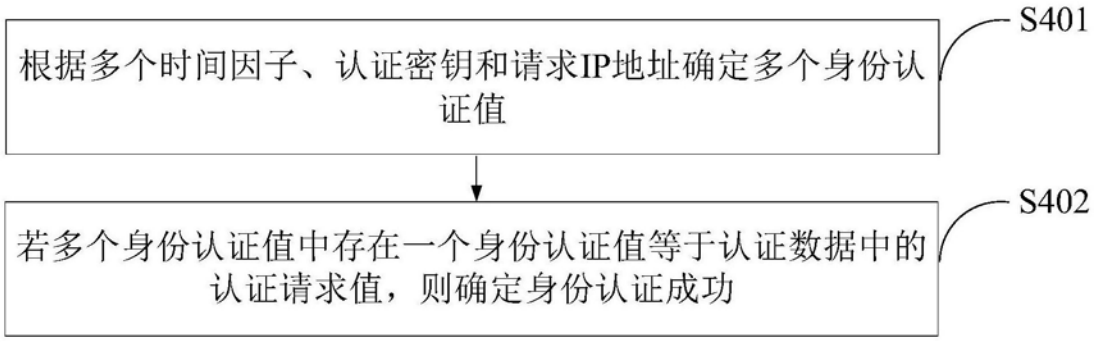

根据多个时间因子、认证密钥和请求IP地址确定多个身份认证值
S401
若多个身份认证值中存在一个身份认证值等于认证数据中的认证请求值，则确定身份认证成功
S402

图4

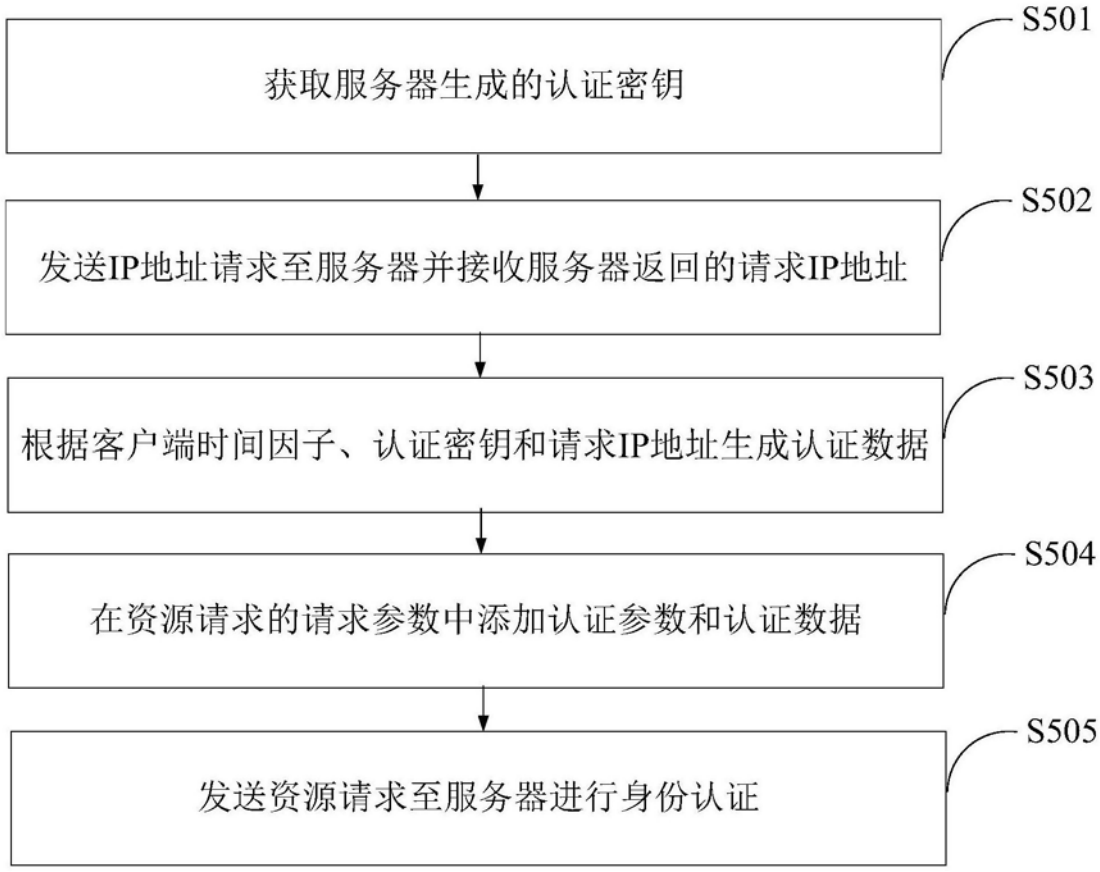

获取服务器生成的认证密钥
S501
发送IP地址请求至服务器并接收服务器返回的请求IP地址
S502
根据客户端时间因子、认证密钥和请求IP地址生成认证数据
S503
在资源请求的请求参数中添加认证参数和认证数据
S504
发送资源请求至服务器进行身份认证
S505

图5

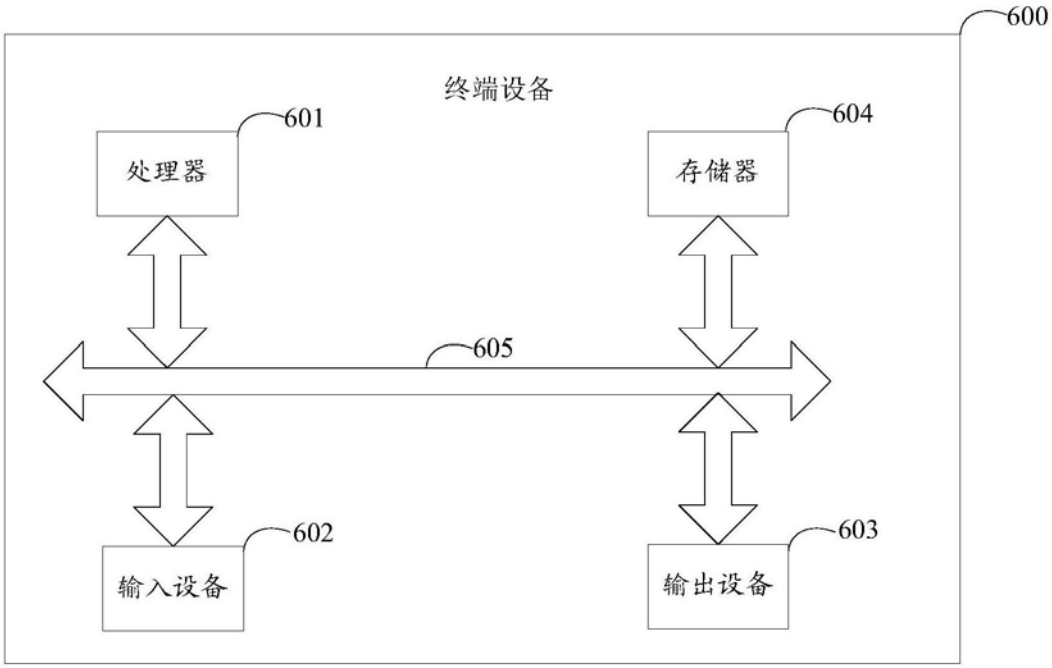


图6