(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 114844644 A
(43)申请公布日 2022.08.02

(21)申请号 202210259476.1

(22)申请日 2022.03.16

(71)申请人 深信服科技股份有限公司
地址 518055 广东省深圳市南山区学苑大道1001号南山智园A1栋

(72)发明人 李想 朱昌亮 钟武杰

(74)专利代理机构 北京派特恩知识产权代理有限公司 11270
专利代理师 王军红 张颖玲

(51)Int.Cl.
H04L 9/32(2006.01)
H04L 9/40(2022.01)

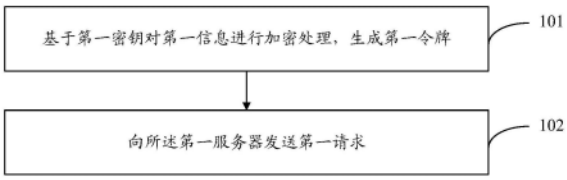
权利要求书2页 说明书15页 附图5页

(54)发明名称

资源请求方法、装置、电子设备及存储介质

(57)摘要

本申请公开了一种资源请求方法、装置、电子设备及存储介质。终端基于与服务器的会话密钥，加密请求对应的信息得到令牌，并将令牌和请求对应的信息随请求发送至服务器。服务器基于请求中的Cookie确定终端，根据与确定出的终端的会话密钥，加密请求携带的信息生成对应的令牌，并基于生成的令牌验证请求携带的令牌，在请求携带的令牌和生成的令牌匹配的情况下，服务器向确定出的终端发送对应的资源。在上述方案中，通过令牌验证请求终端的身份，攻击者即使获取请求携带的终端Cookie和令牌，也无法请求除令牌对应的资源以外的其它资源，这样，能够避免Cookie泄露导致的会话劫持，提升了服务器和终端设备会话的安全性。



CN 114844644 A

1.一种资源请求方法,其特征在于,应用于第一终端,所述方法包括:

基于第一密钥对第一信息进行加密处理,生成第一令牌;所述第一密钥表征所述第一终端与第一服务器之间的会话密钥;

向所述第一服务器发送第一请求;所述第一请求用于向所述第一服务器请求第一资源,且携带有所述第一令牌和所述第一信息;其中,

所述第一信息包括所述第一资源的描述信息和第一时间;所述第一时间表征与所述第一请求相关的时间;所述第一资源在所述第一服务器基于所述第一密钥和所述第一信息对所述第一令牌验证通过后下发至所述第一终端。

2.根据权利要求1所述的方法,其特征在于,所述基于第一密钥对第一信息进行加密处理,生成第一令牌,包括:

将所述第一信息输入第一组件,得到所述第一组件输出的第一令牌;其中,

所述第一组件用于根据对应的密钥对输入的信息进行加密处理,生成并输出令牌;所述第一组件对应的代码经过混淆处理。

3.根据权利要求2所述的方法,其特征在于,在将所述第一信息输入第一组件之后,在得到所述第一组件输出的第一令牌之前,还包括:

基于对第二信息的第一操作,得到所述第一密钥;其中,

所述第二信息表征对所述第一密钥进行第二操作后得到的信息;所述第二操作表征分段存储操作;所述第一操作表征所述第二操作的逆向操作。

4.根据权利要求2所述的方法,其特征在于,所述第一组件对应的代码表征为第一编程语言的代码,由第二编程语言的代码经过混淆处理后再转换得到。

5.根据权利要求1所述的方法,其特征在于,在所述基于第一密钥对第一信息进行加密处理,生成第一令牌之前,所述方法还包括:

在所述第一终端成功登录所述第一服务器的情况下,接收所述第一服务器下发的所述第一密钥。

6.根据权利要求5所述的方法,其特征在于,在所述接收所述第一服务器下发的所述第一密钥之前,所述方法还包括:

向所述第一服务器发送设定标识;其中,

所述第一服务器在接收到所述设定标识的情况下向所述第一终端下发所述第一密钥。

7.一种资源请求方法,其特征在于,应用于第一服务器,所述方法包括:

接收第二请求;所述第二请求用于请求第二资源,且携带有第二令牌和第三信息;所述第三信息包括所述第二资源的描述信息和第二时间;所述第二时间表征与所述第二请求相关的时间;

基于第二密钥对所述第三信息进行加密处理,生成第三令牌;所述第二密钥表征第二终端与所述第一服务器之间的会话密钥;所述第二终端根据第二请求中的Cookie确定出;

在所述第二令牌与所述第三令牌匹配的情况下,向所述第二终端发送所述第二资源。

8.根据权利要求7所述的方法,其特征在于,在所述接收第二请求之前,所述方法还包括:

在所述第二终端成功登录所述第一服务器的情况下,生成并向所述第二终端下发所述第二密钥。

9.根据权利要求8所述的方法,其特征在于,所述生成并向所述第二终端下发所述第二密钥,包括:

在接收到所述第二终端发送的设定标识的情况下,生成并向所述第二终端下发所述第二密钥。

10.根据权利要求7所述的方法,其特征在于,所述生成第三令牌,包括:

在所述第二请求满足第一设定条件的情况下,生成第三令牌;

所述第一设定条件包括:

请求中的Cookie验证通过;

和/或,

请求携带的第二时间在设定时间段内。

11.根据权利要求7所述的方法,其特征在于,所述方法还包括:

在所述第二令牌与所述第三令牌不匹配的情况下,根据所述第二请求中的Cookie删除所述第一服务器存储的对应Cookie。

12.一种资源请求装置,其特征在于,应用于第一终端,包括:

第一生成单元,用于基于第一密钥对第一信息进行加密处理,生成第一令牌;所述第一密钥表征所述第一终端与第一服务器之间的会话密钥;

第一发送单元,用于向所述第一服务器发送第一请求;所述第一请求用于向所述第一服务器请求第一资源,且携带有所述第一令牌和所述第一信息;其中,

所述第一信息包括所述第一资源的描述信息和第一时间;所述第一时间表征与所述第一请求相关的时间;所述第一资源在所述第一服务器基于所述第一密钥和所述第一信息对所述第一令牌验证通过后下发至所述第一终端。

13.一种资源请求装置,其特征在于,应用于第一服务器,包括:

第一接收单元,用于接收第二请求;所述第二请求用于请求第二资源,且携带有第二令牌和第三信息;所述第三信息包括所述第二资源的描述信息和第二时间;所述第二时间表征与所述第二请求相关的时间;

第二生成单元,用于基于第二密钥对所述第三信息进行加密处理,生成第三令牌;所述第二密钥表征第二终端与所述第一服务器之间的会话密钥;所述第二终端根据第二请求中的Cookie确定出;

第二发送单元,用于在所述第二令牌与所述第三令牌匹配的情况下,向所述第二终端发送所述第二资源。

14.一种电子设备,其特征在于,包括:处理器和用于存储能够在处理器上运行的计算机程序的存储器,

其中,所述处理器用于运行所述计算机程序时,执行权利要求1至6任一项所述方法的步骤,或执行权利要求7至11任一项所述方法的步骤。

15.一种存储介质,其上存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现以下至少之一:

权利要求1至6任一项所述方法的步骤;

权利要求7至11任一项所述方法的步骤。

资源请求方法、装置、电子设备及存储介质

技术领域

[0001] 本申请涉及网络技术领域，尤其涉及一种资源请求方法、装置、电子设备及存储介质。

背景技术

[0002] Cookie，是服务器等电子设备为了辨别用户身份，进行Session会话跟踪而储存在终端设备上的数据。Cookie泄露后，攻击者能够利用Cookie劫持会话，会话的安全性低。

发明内容

[0003] 有鉴于此，本申请实施例提供一种资源请求方法、装置、电子设备及存储介质，以至少解决相关技术存在的会话的安全性低的问题。

[0004] 本申请实施例的技术方案是这样实现的：

[0005] 本申请实施例提供了一种资源请求方法，应用于第一终端，所述方法包括：

[0006] 基于第一密钥对第一信息进行加密处理，生成第一令牌；所述第一密钥表征所述第一终端与第一服务器之间的会话密钥；

[0007] 向所述第一服务器发送第一请求；所述第一请求用于向所述第一服务器请求第一资源，且携带有所述第一令牌和所述第一信息；其中，

[0008] 所述第一信息包括所述第一资源的描述信息和第一时间；所述第一时间表征与所述第一请求相关的时间；所述第一资源在所述第一服务器基于所述第一密钥和所述第一信息对所述第一令牌验证通过后下发至所述第一终端。

[0009] 其中，上述方案中，所述基于第一密钥对第一信息进行加密处理，生成第一令牌，包括：

[0010] 将所述第一信息输入第一组件，得到所述第一组件输出的第一令牌；其中，

[0011] 所述第一组件用于根据对应的密钥对输入的信息进行加密处理，生成并输出令牌；所述第一组件对应的代码经过混淆处理。

[0012] 上述方案中，在将所述第一信息输入第一组件之后，在得到所述第一组件输出的第一令牌之前，还包括：

[0013] 基于对第二信息的第一操作，得到所述第一密钥；其中，

[0014] 所述第二信息表征对所述第一密钥进行第二操作后得到的信息；所述第二操作表征分段存储操作；所述第一操作表征所述第二操作的逆向操作。

[0015] 上述方案中，所述第一组件对应的代码表征为第一编程语言的代码，由第二编程语言的代码经过混淆处理后再转换得到。

[0016] 上述方案中，在所述基于第一密钥对第一信息进行加密处理，生成第一令牌之前，所述方法还包括：

[0017] 在所述第一终端成功登录所述第一服务器的情况下，接收所述第一服务器下发的所述第一密钥。

[0018] 上述方案中，在所述接收所述第一服务器下发的所述第一密钥之前，所述方法还包括：

[0019] 向所述第一服务器发送设定标识；其中，

[0020] 所述第一服务器在接收到所述设定标识的情况下向所述第一终端下发所述第一密钥。

[0021] 本申请实施例还提供了一种资源请求方法，应用于第一服务器，所述方法包括：

[0022] 接收第二请求；所述第二请求用于请求第二资源，且携带有第二令牌和第三信息；所述第三信息包括所述第二资源的描述信息和第二时间；所述第二时间表征与所述第二请求相关的时间；

[0023] 基于第二密钥对所述第三信息进行加密处理，生成第三令牌；所述第二密钥表征第二终端与所述第一服务器之间的会话密钥；所述第二终端根据第二请求中的Cookie确定出；

[0024] 在所述第二令牌与所述第三令牌匹配的情况下，向所述第二终端发送所述第二资源。

[0025] 上述方案中，在所述接收第二请求之前，所述方法还包括：

[0026] 在所述第二终端成功登录所述第一服务器的情况下，生成并向所述第二终端下发所述第二密钥。

[0027] 上述方案中，所述生成并向所述第二终端下发所述第二密钥，包括：

[0028] 在接收到所述第二终端发送的设定标识的情况下，生成并向所述第二终端下发所述第二密钥。

[0029] 上述方案中，所述生成第三令牌，包括：

[0030] 在所述第二请求满足第一设定条件的情况下，生成第三令牌；

[0031] 所述第一设定条件包括：

[0032] 请求中的Cookie验证通过；

[0033] 和/或，

[0034] 请求携带的第二时间在设定时间段内。

[0035] 上述方案中，所述方法还包括：

[0036] 在所述第二令牌与所述第三令牌不匹配的情况下，根据所述第二请求中的Cookie删除所述第一服务器存储的对应Cookie。

[0037] 本申请实施例还提供了一种资源请求装置，应用于第一终端，包括：

[0038] 第一生成单元，用于基于第一密钥对第一信息进行加密处理，生成第一令牌；所述第一密钥表征所述第一终端与第一服务器之间的会话密钥；

[0039] 第一发送单元，用于向所述第一服务器发送第一请求；所述第一请求用于向所述第一服务器请求第一资源，且携带有所述第一令牌和所述第一信息；其中，

[0040] 所述第一信息包括所述第一资源的描述信息和第一时间；所述第一时间表征与所述第一请求相关的时间；所述第一资源在所述第一服务器基于所述第一密钥和所述第一信息对所述第一令牌验证通过后下发至所述第一终端。

[0041] 本申请实施例还提供了一种资源请求装置，应用于第一服务器，包括：

[0042] 第一接收单元，用于接收第二请求；所述第二请求用于请求第二资源，且携带有第

二令牌和第三信息;所述第三信息包括所述第二资源的描述信息和第二时间;所述第二时间表征与所述第二请求相关的时间;

[0043] 第二生成单元,用于基于第二密钥对所述第三信息进行加密处理,生成第三令牌;所述第二密钥表征第二终端与所述第一服务器之间的会话密钥;所述第二终端根据第二请求中的Cookie确定出;

[0044] 第二发送单元,用于在所述第二令牌与所述第三令牌匹配的情况下,向所述第二终端发送所述第二资源。

[0045] 本申请实施例还提供了一种电子设备,包括:处理器和用于存储能够在处理器上运行的计算机程序的存储器,

[0046] 其中,所述处理器用于运行所述计算机程序时,执行上述任一种资源请求方法的步骤。

[0047] 本申请实施例还提供了一种存储介质,其上存储有计算机程序,所述计算机程序被处理器执行时实现上述任一种资源请求方法的步骤。

[0048] 在本申请实施例中,终端基于与服务器的会话密钥加密请求对应的信息,包括待请求资源的描述信息和请求相关的时间,得到令牌,并将令牌和请求对应的上述信息随请求发送至服务器。服务器基于请求中的Cookie确定终端,根据与确定出的终端的会话密钥,加密请求携带的信息,包括请求资源的描述信息和请求相关的时间,生成对应的令牌,并基于生成的令牌验证请求携带的令牌,在请求携带的令牌和生成的令牌匹配的情况下,服务器向确定出的终端发送对应的资源。在上述基于令牌的验证结果下发资源的方案中,通过令牌验证请求终端的身份,由于令牌与请求相关,攻击者即使获取请求携带的终端Cookie和令牌,也无法请求除令牌对应的资源以外的其它资源,这样,能够避免Cookie泄露导致的会话劫持,提升了服务器和终端设备会话的安全性。

附图说明

[0049] 图1为本申请实施例提供的资源请求方法的终端侧实现流程示意图;

[0050] 图2为本申请另一实施例提供的资源请求方法中生成第一令牌的实现流程示意图;

[0051] 图3为本申请另一实施例提供的资源请求方法中编程语言转换示意图;

[0052] 图4为本申请另一实施例提供的资源请求方法中分组示意图;

[0053] 图5为本申请另一实施例提供的资源请求方的实现流程示意图;

[0054] 图6为本申请另一实施例提供的资源请求方法的服务器侧实现流程示意图;

[0055] 图7为本申请另一实施例提供的资源请求方法的实现流程示意图;

[0056] 图8为本申请另一实施例提供的资源请求方法的实现流程示意图;

[0057] 图9为本申请应用实施例提供的资源请求方法的交互示意图;

[0058] 图10为本申请实施例提供的资源请求装置的结构示意图;

[0059] 图11为本申请另一实施例提供的资源请求装置的结构示意图;

[0060] 图12为本申请实施例提供的一种电子设备的结构示意图。

具体实施方式

[0061] Cookie,是服务器等电子设备为了辨别用户身份,进行Session会话跟踪而储存在终端设备上的数据。在一次会话过程当中,攻击者能够利用Cookie作为第三方参与,在数据包中插入恶意数据、请求资源、监听会话,甚至可以是代替一方接管会话,会话的安全性低。

[0062] 基于此,在本申请的各种实施例中,终端基于与服务器的会话密钥加密请求对应的信息,包括待请求资源的描述信息和请求相关的时间,得到令牌,并将令牌和请求对应的上述信息随请求发送至服务器。服务器基于请求中的Cookie确定终端,根据与确定出的终端的会话密钥,加密请求携带的信息,包括请求资源的描述信息和请求相关的时间,生成对应的令牌,并基于生成的令牌验证请求携带的令牌,在请求携带的令牌和生成的令牌匹配的情况下,服务器向确定出的终端发送对应的资源。在上述基于令牌的验证结果下发资源的方案中,通过令牌验证请求终端的身份,由于令牌与请求相关,攻击者即使获取请求携带的终端Cookie和令牌,也无法请求除令牌对应的资源以外的其它资源,这样,能够避免Cookie泄露导致的会话劫持,提升了服务器和终端设备会话的安全性。

[0063] 为了使本申请的目的、技术方案及优点更加清楚明白,以下结合附图及实施例,对本申请进行进一步详细说明。应当理解,此处描述的具体实施例仅仅用以解释本申请,并不用于限定本申请。

[0064] 下面将通过实施例并结合附图具体地对本申请的技术方案以及本申请的技术方案如何解决上述技术问题进行详细说明。下面这几个具体的实施例可以相互结合,对于相同或相似的概念或过程可能在某些实施例中不再赘述。

[0065] 图1为本申请实施例提供的资源请求方法的实现流程示意图,本申请实施例提供了一种资源请求方法,应用于第一终端,其中,第一终端包括但不限于手机、平板等电子设备。第一终端可以通过装载浏览器等方式实现资源请求方法。资源请求方法包括:

[0066] 步骤101:基于第一密钥对第一信息进行加密处理,生成第一令牌。

[0067] 其中,所述第一密钥表征所述第一终端与第一服务器之间的会话密钥。

[0068] 步骤102:向所述第一服务器发送第一请求;所述第一请求用于向所述第一服务器请求第一资源,且携带有所述第一令牌和所述第一信息。

[0069] 其中,所述第一信息包括所述第一资源的描述信息和第一时间;所述第一时间表征与所述第一请求相关的时间;所述第一资源在所述第一服务器基于所述第一密钥和所述第一信息对所述第一令牌验证通过后下发至所述第一终端。

[0070] 在步骤101中,第一终端与第一服务器之间预先建立了会话关系,第一终端拥有与第一服务器的会话密钥(也就是第一密钥)。第一终端确定待请求的第一资源的描述信息,并根据当前请求相关的时间确定出第一时间。第一终端以第一信息(包括第一资源的描述信息和第一时间)为计算的输入,利用第一密钥进行哈希函数计算,得到计算输出的第一令牌。令牌的计算可以通过基于哈希的消息身份验证码(HMAC,Hash-based Message Authentication Code),基于哈希函数和会话密钥实现。

[0071] 其中,第一信息表征与当前请求对应的信息,包括第一时间和第一资源的描述信息。基于第一信息能够生成唯一的令牌,以标识对应的请求。第一时间能够将当前请求与此前或此后的请求区分开,例如,可以是在当前生成请求的时间段内的时刻或时间段。优选地,第一终端根据每次请求都需要执行的设定行为的执行时间确定第一时间,包括但不限

于:终端接收到指示发起资源请求的指令的时间;终端生成第一令牌的时间。在一些实施例中,第一时间可以是时间戳的形式。

[0072] 第一资源是指可以被访问的对象,例如文档、图像、声音等数据。第一资源的描述信息表征相应资源位置的描述信息,根据请求方式的不同,描述信息包括:请求网址和/或web表单信息。由于每次请求对应的第一时间不同,每次请求对应的第一令牌不同,第一令牌与第一请求存在对应关系。

[0073] 第一密钥表征第一终端和第一服务器之间会话的会话密钥,由终端和服务器通过协商确定出,会话密钥可以代表这次会话中与服务器通信的终端的身份。

[0074] 在步骤102中,第一终端至少基于生成的第一令牌、第一信息(也就是第一令牌对应的信息),生成并向第一服务器发送第一请求的请求报文,以请求第一服务器下发第一资源。在第一服务器对第一令牌验证通过的情况下,第一终端接收第一服务器基于第一请求对应发送的第一资源。这里,请求报文中还携带有第一终端的Cookie,服务器根据接收到的请求报文携带的Cookie判断对应终端的身份,在服务器存储的会话密钥中确定出与终端身份对应的会话密钥,以实现根据会话密钥和第一信息验证第一令牌。

[0075] 在上述基于令牌的验证结果下发资源的方案中,对于每次资源请求,第一终端需要执行步骤101生成对应的令牌,并在步骤102发送请求时携带生成令牌,以获得服务器在请求携带的令牌验证通过后下发的资源。

[0076] 作为本申请的另一实施例,如图2所示,基于第一密钥对第一信息进行加密处理,生成第一令牌,包括:

[0077] 步骤1011:将所述第一信息输入第一组件,得到所述第一组件输出的第一令牌。

[0078] 其中,所述第一组件用于根据对应的密钥对输入的信息进行加密处理,生成并输出令牌;所述第一组件对应的代码经过混淆处理。

[0079] 第一终端将第一信息输入第一组件,由第一组件根据对应的密钥(也就是第一密钥)对输入的第一信息进行加密处理生成第一令牌,第一终端得到第一组件输出的第一令牌。其中,第一组件对应的代码经过混淆处理,第一组件相当于黑盒,从而保护第一组件的代码逻辑,提高了攻击者逆向分析破解的难度,也就是说,对第一组件对应的代码的混淆处理,能够降低被逆向破解的风险。

[0080] 第一终端通过调用代码经过混淆处理的第一组件,实现基于第一密钥对第一信息的加密处理得到第一令牌。在资源请求过程中利用混淆处理过的组件生成令牌,这样,生成第一令牌的代码逻辑对前端不可见,提升了前端操作的安全性,增加了攻击者逆向工程的难度和时间成本,从而提高了会话的安全性。

[0081] 作为本申请的一个实施例,所述第一组件对应的代码表征为第一编程语言的代码,由第二编程语言的代码经过混淆处理后再转换得到。

[0082] 首先生成第一组件对应的第二编程语言的代码,再对第二编程语言的代码进行代码混淆处理,将混淆处理后的第二编程语言代码转换为第一编程语言的代码。在实际应用中,第二编程语言可以是C/C++,第一编程语言可以是JS。

[0083] 对不同编程语言的代码进行转换,可以借助OLLVM的后端实现,增加了逆向工程的时间成本和难度。

[0084] 这样,增加了攻击者逆向工程的难度和时间成本,降低了被逆向破解的风险,从而

提高了会话的安全性。

[0085] 作为本申请的一个实施例，结合图3示出编程语言转换示意图进行说明，生成第一组件对应的C/C++(第二编程语言)代码，通过Clang前端将C/C++的代码转换为LLVM IR code，再借助OLLVM的后端对前端生成的IR code进行负优化，使用的混淆策略有基本块分割、指令膨胀、虚假块填充和控制流平坦。利用LLVM webasm的后端和wasm-ld链接器将混淆后的IR code转换为前端js(第一编程语言)代码。

[0086] 作为本申请的一个实施例，在将所述第一信息输入第一组件之后，在得到所述第一组件输出的第一令牌之前，还包括：

[0087] 基于对第二信息的第一操作，得到所述第一密钥；其中，

[0088] 所述第二信息表征对所述第一密钥进行第二操作后得到的信息；所述第二操作表征分段存储操作；所述第一操作表征所述第二操作的逆向操作。

[0089] 第一终端对第一密钥经过分段存储操作(第二操作)，得到存储在设定存储介质中的第二信息。在将第一信息输入第一组件之后，第一组件对设定存储介质中存储的第二信息进行数据读取和还原处理(第一操作)，得到第一密钥。其中，分段存储操作包括将会话密钥根据图4拆分成第一设定数量的字段，并将拆分后的字段分组，将每组数据分段存储于存储介质。设定组件读取分段存储于不同位置的数据，再根据图4的逆向操作还原得到第一密钥。在分组时，可以将拆分的字段等分或不等分至第二设定数量的组内。

[0090] 这里，第二操作可以认为是一种编码操作，第一操作是对应的解码操作，将第一密钥以对应的编码结果存储，在第一组件还原得到第一密钥。

[0091] 实际应用中，以图4示出的分组示意图为例进行说明，将会话密钥拆分成16个字段，将拆分后的字段以如图的方式进行排列，并划分为4组。

[0092] 这样，第一密钥被隐藏在黑盒的第一组件中，保证会话密钥对前端不可见，增加了攻击者逆向工程的难度和时间成本，从而提高了会话的安全性。

[0093] 作为本申请的另一个实施例，如图5所示，在所述基于第一密钥对第一信息进行加密处理，生成第一令牌之前，所述方法还包括：

[0094] 步骤501：在所述第一终端成功登录所述第一服务器的情况下，接收所述第一服务器下发的所述第一密钥。

[0095] 第一终端将登录信息发送至第一服务器，第一服务器基于登录信息对第一终端进行用户认证。在服务器确定用户认证成功之后，也就是在第一终端成功登录第一服务器之后，第一终端和第一服务器之间成功建立会话关系，第一服务器为新创建的会话生成Cookie和会话密钥(也就是第一密钥)，并向第一终端下发Cookie和第一密钥。其中，登录信息用于服务器对终端的认证，包括但不限于：用户标识(uid)和/或密码(pwd)。

[0096] 作为本申请的一个实施例，在所述接收所述第一服务器下发的所述第一密钥之前，所述方法还包括：

[0097] 向所述第一服务器发送设定标识。

[0098] 其中，所述第一服务器在接收到所述设定标识的情况下向所述第一终端下发所述第一密钥。

[0099] 在第一终端与第一服务器建立会话关系的过程中，第一终端向第一服务器发送设定标识，以表明第一终端支持特定的会话认证方式和协议版本，这里的特定会话认证方式

和协议版本与本申请实施例的资源请求方法对应，区别于标准Cookies认证方式。服务器接收到设定标识，则确定第一终端支持特定会话认证方式和协议版本，向第一终端下发Cookie和第一密钥。

[0100] 第一终端向第一服务器发送设定标识的方式，可以是与登录信息一起的发送给第一服务器，也可以是额外发送给第一服务器，在此不进行限定。优选地，设定标识以报文字头的方式携带于登录信息中。

[0101] 通过设定标识，第一终端向第一服务器传递了终端支持特定会话认证方式和协议版本的信息，第一服务器获知该信息后确定以本申请实施例对应的方式与第一终端通信。

[0102] 图6为本申请另一实施例提供的资源请求方法的实现流程示意图，本申请实施例提供了一种资源请求方法，应用于第一服务器。资源请求方法包括：

[0103] 步骤601：接收第二请求。

[0104] 其中，所述第二请求用于请求第二资源，且携带有第二令牌和第三信息；所述第三信息包括所述第二资源的描述信息和第二时间；所述第二时间表征与所述第二请求相关的时间。

[0105] 步骤602：基于第二密钥对所述第三信息进行加密处理，生成第三令牌。

[0106] 其中，所述第二密钥表征第二终端与所述第一服务器之间的会话密钥；所述第二终端根据第二请求中的Cookie确定出。

[0107] 步骤603：在所述第二令牌与所述第三令牌匹配的情况下，向所述第二终端发送所述第二资源。

[0108] 第一服务器预先与一些终端建立了会话关系，生成并下发了对应的会话密钥和Cookie。第一服务器接收到的请求，可能是用户通过终端使用自己的身份发送的，也可能是攻击者冒用的用户终端的身份发送的，而第一服务器根据请求携带的Cookie只能确定出请求对应的终端身份，无法辨别请求发送方是否为Cookie的身份，也就是说，第一服务器不能区分请求是由用户终端或攻击者发送的。

[0109] 在步骤601中，第一服务器接收到第二请求，第二请求携带有第二令牌、第二令牌对应的第三信息和Cookie。其中，第三信息表征与当前请求对应的信息，包括第二时间和第二资源的描述信息。基于第三信息能够生成唯一的令牌，以标识对应的请求。对于用户通过终端发送的请求，第二时间能够将当前请求与此前或此后的请求区分开，例如，可以是在当前生成请求的时间段内的时刻或时间段。优选地，第二终端根据每次请求都需要执行的设定行为的执行时间确定第二时间。在一些实施例中，第二时间可以是时间戳的形式。第二令牌由终端通过会话密钥加密处理对应的源数据得到，第二令牌的源数据可以是第三信息，也可以不是第三信息，例如，出于攻击的目的，第三方(攻击者)截获并利用第二令牌发起资源请求，所请求的资源通常不同于原请求的资源，那么第二令牌的源数据不是第三信息。

[0110] 因而，在步骤602中，第一服务器根据第二请求携带的Cookie确定出终端身份为第二终端，并利用与第二终端的会话密钥(也就是第二密钥)加密第三信息得到第三令牌，以第三令牌对第二令牌进行验证，从而实现对发送方的辨认。这里第一服务器以第三令牌对第二令牌进行验证的方式，可以是将第三令牌与第二令牌进行比对，在第三令牌与第二令牌的数据相同的情况下，确定第二令牌与第三令牌匹配。这里，根据Cookie确定出的第二终端，可能是发送第二请求的用户终端的身份，也可能是攻击者冒用的身份。

[0111] 作为本申请的一个实施例,所述生成第三令牌,包括:

[0112] 在所述第二请求满足第一设定条件的情况下,生成第三令牌;

[0113] 所述第一设定条件包括:

[0114] 请求中的Cookie验证通过;

[0115] 和/或,

[0116] 请求携带的第二时间在设定时间段内。

[0117] 第一服务器判断接收到的第二请求是否满足第一设定条件,在满足第一设定条件的情况下,生成第三令牌。其中,第一设定条件可以是请求中的Cookie验证通过,可以是请求携带的第二时间在设定时间段内,还可以是请求中的Cookie验证通过且请求携带的第二时间在设定时间段内。需要说明的是,第二请求满足第一设定条件,是第一服务器生成第三令牌的前提条件。

[0118] 实际应用中,以第一设定条件包括请求中的Cookie验证通过和请求携带的第二时间在设定时间段内为例,对第三令牌生成前的判断条件进行说明:第一服务器验证Cookie,如果Cookie验证不通过则返回提示信息;如果Cookie验证通过,则判断第二时间是否在设定时间段内;如果第二时间在设定时间段内,第一服务器生成第三令牌。其中,判断第二时间是否在设定时间段内,需要以终端与服务器的时间对齐为前提。时间段为一个设定时间范围,可根据各个服务器需要进行设定,如果安全需求较高,则设定时间范围应该更小。

[0119] 在步骤603中,在请求携带的第二令牌与生成的第三令牌匹配的情况下,第一服务器确定第二令牌的源数据是第三信息,换句话说,第二请求是由用户终端发送的。此时,第一服务器向第二终端发送第二请求所请求的第二资源。

[0120] 作为本申请的一个实施例,如图7所示,所述方法还包括:

[0121] 步骤604:在所述第二令牌与所述第三令牌不匹配的情况下,根据所述第二请求中的Cookie删除所述第一服务器存储的对应Cookie。

[0122] 在请求携带的第二令牌与生成的第三令牌不匹配的情况下,第一服务器确定第二请求并非用户终端发送的,而是由第三方(攻击者)发送的。由于第三方已经获取对应用户终端的Cookie信息,并利用Cookie信息发起第二请求,与第二终端的会话的Cookie已经泄露,第一服务器删除对应Cookie,还可以发送消息要求第二终端重新登录。

[0123] 通过令牌验证请求终端的身份,由于令牌与请求相关,攻击者即使获取请求携带的终端Cookie和令牌,也无法请求除令牌对应的资源以外的其它资源,这样,能够避免Cookie泄露导致的会话劫持,提升了服务器和终端设备会话的安全性。

[0124] 作为本申请的另一个实施例,如图8所示,在所述接收第二请求之前,所述方法还包括:

[0125] 步骤801:在所述第二终端成功登录所述第一服务器的情况下,生成并向所述第二终端下发所述第二密钥。

[0126] 第二终端将登录信息发送至第一服务器,第一服务器基于登录信息对第二终端进行用户认证。在服务器确定用户认证成功之后,也就是在第二终端成功登录第一服务器之后,第二终端和第一服务器之间成功建立会话关系,第一服务器为新创建的会话生成Cookie和会话密钥(也就是第二密钥),并向第二终端下发Cookie和第二密钥。其中,登录信息用于服务器对终端的认证,包括但不限于:用户标识(uid)和/或密码(pwd)。

[0127] 作为本申请的一个实施例,所述生成并向所述第二终端下发所述第二密钥,包括:

[0128] 在接收到所述第二终端发送的设定标识的情况下,生成并向所述第二终端下发所述第二密钥。

[0129] 在第二终端与第一服务器建立会话关系的过程中,第二终端向第一服务器发送设定标识,以表明第二终端支持特定的会话认证方式和协议版本,这里的特定会话认证方式和协议版本与本申请实施例的资源请求方法对应,区别于标准Cookies认证方式。第一服务器在接收到第二终端发送的设定标识的情况下,则确定第二终端支持特定会话认证方式和协议版本,生成并向第二终端下发Cookie和第二密钥。需要说明的是,第一服务器接收到第二终端发送的设定标识,是第一服务器生成并向第二终端下发Cookie和第二密钥的前提条件。

[0130] 第二终端向第一服务器发送设定标识的方式,可以是与登录信息一起的发送给第一服务器,也可以是额外发送给第一服务器,在此不进行限定。优选地,设定标识以报文字头的方式携带于登录信息中。

[0131] 通过设定标识,第二终端向第一服务器传递了终端支持特定会话认证方式和协议版本的信息,第一服务器获知该信息后确定以本申请实施例对应的方式与第二终端通信。

[0132] 下面结合应用实施例对本申请再作进一步详细的描述。

[0133] Cookie泄露后,攻击者能够重复利用Cookie发起会话,会话的安全性低。同时,服务器难以将攻击者的请求与用户的请求区分开,导致资源信息泄露。

[0134] 基于此,本申请应用实施例提出了一种基于黑盒实现OTC的防Cookie劫持认证加固方案。图9示出了本申请应用实施例提供的交互示意图,图中浏览器/客户端表征终端侧,服务器表征服务侧(服务器侧),整个过程中至少包括:

[0135] (1)初始化阶段

[0136] 初始化阶段发生于用户登录时,服务器分发代表这次会话身份的会话密钥(k_s),浏览器/客户端接收到密钥后将利用黑盒实现方法隐藏该会话密钥。

[0137] 1.1登录操作

[0138] 浏览器向服务器发送用户标识(uid)和密码(pwd)以及一个特殊的HTTP头字段:X-OTC(即设定标识)。该报头字段表明浏览器支持OTC会话认证和OTC协议版本(v)。

[0139] 1.2认证用户

[0140] 在成功的用户认证之后,服务器检查请求中是否存在X-OTC报头字段。

[0141] 如果该报头字段存在,则服务器为新创建的会话(cid)生成Cookie和会话密钥(k_s)。该会话密钥代表了此次与服务器通信的浏览器/客户端身份。

[0142] 如果浏览器请求中不存在X-OTC报头字段,则可以切换到标准Cookies认证,也可以停止通信并通知用户OTC支持是强制性的。

[0143] 随后,服务器存储用户标识(uid)、会话标识(cid)和该会话密钥(k_s)。

[0144] 1.3返回操作

[0145] 服务器将生成的Cookie和对称加密后的会话密钥(k_s)返还给浏览器/客户端。

[0146] 1.4隐藏密钥操作

[0147] 为了保证浏览器/客户端存储的机密性,浏览器/客户端在前端通过设定组件将加密后的会话密钥(k_s)进行分段存储处理。

[0148] 其中，将加密后的会话密钥拆分成第一设定数量的字段，并将拆分后的字段分组，将每组数据分段存储于存储介质。结合图4对分段存储处理过程进行说明，将加密后的会话密钥拆分成16个字段，并将拆分后的字段按照图4的方式进行分组。

[0149] 为了保证前端操作的安全性，对设定组件对应的JS代码进行代码混淆，这里的设定组件可以执行本应用实施例中的隐藏密钥操作、取密钥操作和部分请求操作，具体地，可以执行会话密钥的编解码处理和令牌生成。

[0150] (2)请求阶段

[0151] 请求阶段发生于客户通过终端发起的每次请求时，浏览器/客户端逆向操作会话密钥(k_s)，并生成唯一的OTC令牌。

[0152] 2.1取密钥操作

[0153] 设定组件将分段存储于不同位置的数据取出，再根据图4的逆向操作还原，得到加密后的会话密钥(k_s)，最后解密获得会话密钥。

[0154] 2.2请求操作

[0155] 对于每一个请求，浏览器会在Cookie字段之外附加一个OTC令牌(即第一令牌)。随后，浏览器/客户端将Cookie和OTC令牌(令牌生成时间(t)和HMAC值)发送给服务器。

[0156] 其中，OTC令牌使用会话密钥生成的基于哈希的消息身份验证码(HMAC(k_s,url|t|data))。HMAC计算包括请求的网址(url)、令牌生成时间(t)，对于POST请求还包含的任何web表单信息(data)，而GET请求的参数包含在网址中。因此，对于每个请求携带的令牌，对应的生成时间不同，对应一个唯一的消息身份验证码，从而保证令牌的唯一性。即使请求相同的资源(请求的网址和POST请求中包含的任何web表单信息)，对应的令牌也不相同。攻击者只能重放完全相同的请求，请求相同的资源，因为修改请求的任何载荷，对应的HMAC都会改变，从而无法通过验证。因此，攻击者不能重用OTC令牌来非法重定向会话。

[0157] (3)验证阶段

[0158] 验证阶段发生于请求阶段之后，服务器将对用户的请求进行验证。如果验证成功则返回数据，否则要求用户重新登录。

[0159] 3.1验证操作

[0160] 服务器首先验证Cookie中的信息，若验证不成功，则返回Cookie失效或Cookie错误信息。其次，服务器检查令牌生成时间(t)是否在设定时间段内。此步骤的前提是需要对齐浏览器/客户端与服务器的时间，判断请求携带的时间戳是否在时间段内。时间段为一个设定时间范围，可根据各个服务器需要进行设定，如果安全需求较高，则设定时间范围应该更小。随后，服务器使用会话密钥k_s和请求中的信息(url和data)计算新的HMAC。然后，服务器将新计算的HMAC与令牌中包含的HMAC进行比较。

[0161] 3.2返回操作

[0162] 如果HMAC值匹配，则请求验证通过，服务器返回请求对应的资源。如果HMAC值不匹配，或者上文提及的任何检查失败，请求将被拒绝，返回校验不通过。

[0163] (4)注销阶段

[0164] 4.1注销操作

[0165] 会话将继续，直到会话票证过期或用户指示注销。注销时，浏览器/客户端向服务器发送Cookie和一个带有相应OTC令牌的请求，该报头字段只包含值为零(0)的HMAC(k_s,

0),该HMAC指示浏览器删除此域的凭证(浏览器强制策略)。

[0166] 4.2注销验证操作

[0167] 浏览器/服务器端基于HMAC删除此域的凭证。同时,防止攻击者欺骗服务器响应任意删除或修改OTC凭证。

[0168] 4.3注销成功

[0169] 注销成功,此次会话完成。

[0170] 前文提及,对设定组件对应的JS代码进行代码混淆。优选地,将相关的C/C++代码转换为LLVM Ir code,随后通过OLLVM进行代码混淆。若在浏览器/服务器端,则通过LLVM-webasm后端编译为前端js代码。如图3示出的,具体流程如下:

[0171] 首先,生成实现上述OTC算法的C/C++的代码。

[0172] 第二,Clang前端将C/C++的代码转换为LLVM IR code。

[0173] 第三,借助OLLVM的后端,对前端生成的IR code进行负优化,使用的混淆策略有基本块分割、指令膨胀、虚假块填充和控制流平坦。

[0174] 第四,利用LLVM webasm的后端和wasm-ld链接器将混淆后的IR code转换为前端js代码。

[0175] 在本应用实施例提供的方案,通过与用户(终端)会话绑定的会话密钥,为每个请求包(包括时间戳和请求资源的描述信息)生成基于哈希的消息身份验证码,进行完整性校验,防止Cookie被劫持后重复利用。同时,黑盒实现方法能够保证设备密钥的获取和HMAC的实现均对用户不可见。在本应用实施例提供的方案,至少具有以下一种效果:

[0176] (1)Cookie泄露后无法重复利用,能够防止攻击者获取更多系统信息;

[0177] (2)混淆后的代码为黑盒,逆向难以分析得到源码本身逻辑,这样,降低了被逆向破解的风险;

[0178] (3)运行实现代码混淆的设定组件所消耗的系统资源(运行性能)减少。

[0179] 这里,对本应用实施例中出现的术语进行解释。

[0180] OTC(One-Time Cookie):一次性Cookie,是一种防止Cookie泄露后重复利用的客户端防劫持方案。Cookie是储存在用户终端上的数据,当浏览器再请求该网站时,浏览器把请求的网址连同Cookie一同提交给服务器。服务器检查该Cookie,以此来辨认用户状态。

[0181] OLLVM:一个基于LLVM编译器的开源混淆方案,极大地增加了逆向工程的时间成本和难度。主要的混淆策略有基本块分割、指令膨胀、虚假块填充和控制流平坦。

[0182] Clang:基于LLVM后端的前端编译器,完全支持C++11标准并且与GNU C语言规范几乎完全兼容。在编译过程中主要负责词法分析、语法分析、语意分析、中间代码(IR)生成。

[0183] LLVM-webAsm:LLVM后端中的一种,可将中间代码转换为JS代码。编译器后端的职责是将前端生成的中间代码(IR code)进行优化,并根据编译时指定的目标平台来生成对应的终端代码。在这里webAsm可被看成一种平台就如同x86,x86_64,aarch64一样。

[0184] HMAC(Hash-based Message Authentication Code):是一种利用密码学中的散列函数来进行消息认证的一种机制,所能提供的消息认证包括消息完整性验证和信源身份认证。HMAC算法更像是一种加密算法,由于HMAC算法引入了密钥,安全性已经不完全依赖于所使用的Hash算法。

[0185] 为实现本申请实施例的方法,本申请实施例还提供了一种资源请求装置,应用于

第一终端，如图10所示，该装置包括：

[0186] 第一生成单元1001，用于基于第一密钥对第一信息进行加密处理，生成第一令牌；所述第一密钥表征所述第一终端与第一服务器之间的会话密钥；

[0187] 第一发送单元1002，用于向所述第一服务器发送第一请求；所述第一请求用于向所述第一服务器请求第一资源，且携带有所述第一令牌和所述第一信息；其中，

[0188] 所述第一信息包括所述第一资源的描述信息和第一时间；所述第一时间表征与所述第一请求相关的时间；所述第一资源在所述第一服务器基于所述第一密钥和所述第一信息对所述第一令牌验证通过后下发至所述第一终端。

[0189] 其中，在一个实施例中，所述第一生成单元1001，用于：

[0190] 将所述第一信息输入第一组件，得到所述第一组件输出的第一令牌；其中，

[0191] 所述第一组件用于根据对应的密钥对输入的信息进行加密处理，生成并输出令牌；所述第一组件对应的代码经过混淆处理。

[0192] 在一个实施例中，所述装置还包括：

[0193] 操作单元，用于在将所述第一信息输入第一组件之后，在得到所述第一组件输出的第一令牌之前，基于对第二信息的第一操作，得到所述第一密钥；其中，所述第二信息表征对所述第一密钥进行第二操作后得到的信息；所述第二操作表征分段存储操作；所述第一操作表征所述第二操作的逆向操作。

[0194] 在一个实施例中，所述第一组件对应的代码表征为第一编程语言的代码，由第二编程语言的代码经过混淆处理后再转换得到。

[0195] 在一个实施例中，所述装置还包括：

[0196] 第二接收单元，用于在所述第一生成单元1001基于第一密钥对第一信息进行加密处理，生成第一令牌之前，在所述第一终端成功登录所述第一服务器的情况下，接收所述第一服务器下发的所述第一密钥。

[0197] 在一个实施例中，所述装置还包括：

[0198] 第三发送单元，用于在所述第二接收单元接收所述第一服务器下发的所述第一密钥之前，向所述第一服务器发送设定标识；其中，所述第一服务器在接收到所述设定标识的情况下向所述第一终端下发所述第一密钥。

[0199] 实际应用时，所述第一发送单元1002、所述第二接收单元、所述第三发送单元可由基于资源请求装置中的通信接口实现，所述第一生成单元1001、所述操作单元可由基于资源请求装置中的处理器实现。

[0200] 需要说明的是：上述实施例提供的资源请求装置在进行资源请求时，仅以上述各程序模块的划分进行举例说明，实际应用中，可以根据需要而将上述处理分配由不同的程序模块完成，即将装置的内部结构划分成不同的程序模块，以完成以上描述的全部或者部分处理。另外，上述实施例提供的资源请求装置与资源请求方法实施例属于同一构思，其具体实现过程详见方法实施例，这里不再赘述。

[0201] 为实现本申请实施例的方法，本申请实施例还提供了一种资源请求装置，应用于第一服务器，如图11所示，该装置包括：

[0202] 第一接收单元1101，用于接收第二请求；所述第二请求用于请求第二资源，且携带有第二令牌和第三信息；所述第三信息包括所述第二资源的描述信息和第二时间；所述第

二时间表征与所述第二请求相关的时间；

[0203] 第二生成单元1102，用于基于第二密钥对所述第三信息进行加密处理，生成第三令牌；所述第二密钥表征第二终端与所述第一服务器之间的会话密钥；所述第二终端根据第二请求中的Cookie确定出；

[0204] 第二发送单元1103，用于在所述第二令牌与所述第三令牌匹配的情况下，向所述第二终端发送所述第二资源。

[0205] 其中，在一个实施例中，所述装置还包括：

[0206] 第四发送单元，用于在所述第一接收单元1101接收第二请求之前，在所述第二终端成功登录所述第一服务器的情况下，生成并向所述第二终端下发所述第二密钥。

[0207] 在一个实施例中，所述第四发送单元，用于：

[0208] 在接收到所述第二终端发送的设定标识的情况下，生成并向所述第二终端下发所述第二密钥。

[0209] 在一个实施例中，所述第二生成单元1102，用于：

[0210] 在所述第二请求满足第一设定条件的情况下，生成第三令牌；

[0211] 所述第一设定条件包括：

[0212] 请求中的Cookie验证通过；

[0213] 和/或，

[0214] 请求携带的第二时间在设定时间段内。

[0215] 在一个实施例中，所述装置还包括：

[0216] 删除单元，用于在所述第二令牌与所述第三令牌不匹配的情况下，根据所述第二请求中的Cookie删除所述第一服务器存储的对应Cookie。

[0217] 实际应用时，所述第一接收单元1101、所述第二发送单元1103可由基于资源请求装置中的通信接口实现，所述第二生成单元1102、删除单元可由基于资源请求装置中的处理器实现，所述第四发送单元可由基于资源请求装置中的处理器结合通信接口实现。

[0218] 需要说明的是：上述实施例提供的资源请求装置在进行资源请求时，仅以上述各程序模块的划分进行举例说明，实际应用中，可以根据需要而将上述处理分配由不同的程序模块完成，即将装置的内部结构划分成不同的程序模块，以完成以上描述的全部或者部分处理。另外，上述实施例提供的资源请求装置与资源请求方法实施例属于同一构思，其具体实现过程详见方法实施例，这里不再赘述。

[0219] 基于上述程序模块的硬件实现，且为了实现本申请实施例资源请求方法，本申请实施例还提供了一种电子设备。图12为本申请实施例电子设备的硬件组成结构示意图，如图12所示，电子设备包括：

[0220] 通信接口1，能够与其它设备比如网络设备等进行信息交互；

[0221] 处理器2，与通信接口1连接，以实现与其它设备进行信息交互，用于运行计算机程序时，执行上述一个或多个技术方案提供的方法。而所述计算机程序存储在存储器3上。

[0222] 当然，实际应用时，电子设备中的各个组件通过总线系统4耦合在一起。可理解，总线系统4用于实现这些组件之间的连接通信。总线系统4除包括数据总线之外，还包括电源总线、控制总线和状态信号总线。但是为了清楚说明起见，在图12中将各种总线都标为总线系统4。

[0223] 本申请实施例中的存储器3用于存储各种类型的数据以支持电子设备的操作。这些数据的示例包括:用于在电子设备上操作的任何计算机程序。

[0224] 可以理解,存储器3可以是易失性存储器或非易失性存储器,也可包括易失性和非易失性存储器两者。其中,非易失性存储器可以是只读存储器(ROM,Read Only Memory)、可编程只读存储器(PROM,Programmable Read-Only Memory)、可擦除可编程只读存储器(EPROM,Erasable Programmable Read-Only Memory)、电可擦除可编程只读存储器(EEPROM,Electrically Erasable Programmable Read-Only Memory)、磁性随机存取存储器(FRAM,ferromagnetic random access memory)、快闪存储器(Flash Memory)、磁表面存储器、光盘、或只读光盘(CD-ROM,Compact Disc Read-Only Memory);磁表面存储器可以是磁盘存储器或磁带存储器。易失性存储器可以是随机存取存储器(RAM,Random Access Memory),其用作外部高速缓存。通过示例性但不是限制性说明,许多形式的RAM可用,例如静态随机存取存储器(SRAM,Static Random Access Memory)、同步静态随机存取存储器(SSRAM,Synchronous Static Random Access Memory)、动态随机存取存储器(DRAM,Dynamic Random Access Memory)、同步动态随机存取存储器(SDRAM,Synchronous Dynamic Random Access Memory)、双倍数据速率同步动态随机存取存储器(DDRSDRAM,Double Data Rate Synchronous Dynamic Random Access Memory)、增强型同步动态随机存取存储器(ESDRAM,Enhanced Synchronous Dynamic Random Access Memory)、同步连接动态随机存取存储器(SLDRAM,SyncLink Dynamic Random Access Memory)、直接内存总线随机存取存储器(DRRAM,Direct Rambus Random Access Memory)。本申请实施例描述的存储器2旨在包括但不限于这些和任意其它适合类型的存储器。

[0225] 上述本申请实施例揭示的方法可以应用于处理器2中,或者由处理器2实现。处理器2可能是一种集成电路芯片,具有信号的处理能力。在实现过程中,上述方法的各步骤可以通过处理器2中的硬件的集成逻辑电路或者软件形式的指令完成。上述的处理器2可以是通用处理器、DSP,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。处理器2可以实现或者执行本申请实施例中的公开的各方法、步骤及逻辑框图。通用处理器可以是微处理器或者任何常规的处理器等。结合本申请实施例所公开的方法的步骤,可以直接体现为硬件译码处理器执行完成,或者用译码处理器中的硬件及软件模块组合执行完成。软件模块可以位于存储介质中,该存储介质位于存储器3,处理器2读取存储器3中的程序,结合其硬件完成前述方法的步骤。

[0226] 处理器2执行所述程序时实现本申请实施例的各个方法中的相应流程,为了简洁,在此不再赘述。

[0227] 在示例性实施例中,本申请实施例还提供了一种存储介质,即计算机存储介质,具体为计算机可读存储介质,例如包括存储计算机程序的存储器3,上述计算机程序可由处理器2执行,以完成前述方法所述步骤。计算机可读存储介质可以是FRAM、ROM、PROM、EPROM、EEPROM、Flash Memory、磁表面存储器、光盘、或CD-ROM等存储器。

[0228] 在本申请所提供的几个实施例中,应该理解到,所揭露的装置、电子设备和方法,可以通过其它的方式实现。以上所描述的设备实施例仅仅是示意性的,例如,所述单元的划分,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,如:多个单元或组件可以结合,或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

各组成部分相互之间的耦合、或直接耦合、或通信连接可以是通过一些接口，设备或单元的间接耦合或通信连接，可以是电性的、机械的或其它形式的。

[0229] 上述作为分离部件说明的单元可以是、或也可以不是物理上分开的，作为单元显示的部件可以是、或也可以不是物理单元，即可以位于一个地方，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全部单元来实现本实施例方案的目的。

[0230] 另外，在本申请各实施例中的各功能单元可以全部集成在一个处理单元中，也可以是各单元分别单独作为一个单元，也可以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，也可以采用硬件加软件功能单元的形式实现。

[0231] 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：移动存储设备、ROM、RAM、磁碟或者光盘等各种可以存储程序代码的介质。

[0232] 或者，本申请上述集成的单元如果以软件功能模块的形式实现并作为独立的产品销售或使用时，也可以存储在一个计算机可读取存储介质中。基于这样的理解，本申请实施例的技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来，该计算机软件产品存储在一个存储介质中，包括若干指令用以使得一台计算机设备(可以是个人计算机、服务器、或网络设备等)执行本申请各个实施例所述方法的全部或部分。而前述的存储介质包括：移动存储设备、ROM、RAM、磁碟或者光盘等各种可以存储程序代码的介质。

[0233] 可以理解的是，在本申请实施例中，涉及到用户信息的数据，当本申请实施例运用到具体产品或技术中时，需要获得用户许可或者同意，且相关数据的收集、使用和处理需要遵守相关国家和地区的相关法律法规和标准。

[0234] 需要说明的是，本申请实施例所记载的技术方案之间，在不冲突的情况下，可以任意组合。除非另有说明和限定，术语“连接”应做广义理解，例如，可以是电连接，也可以是两个元件内部的连通，可以是直接相连，也可以通过中间媒介间接相连，对于本领域的普通技术人员而言，可以根据具体情况理解上述术语的具体含义。

[0235] 另外，在本申请实例中，“第一”、“第二”等是用于区别类似的对象，而不必用于描述特定的顺序或先后次序。应该理解“第一\第二\第三”区分的对象在适当情况下可以互换，以使这里描述的本申请的实施例可以除了在这里图示或描述的那些以外的顺序实施。

[0236] 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，本文中术语“至少一个”表示多个中的任意一个或多个中的至少两个的任意组合，例如，包括A、B、C中的至少一个，可以表示包括从A、B和C构成的集合中选择的任意一个或多个元素。

[0237] 以上所述，仅为本申请的具体实施方式，但本申请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申请揭露的技术范围内，可轻易想到变化或替换，都应涵盖在本申请的保护范围之内。因此，本申请的保护范围应以所述权利要求的保护范围为准。

[0238] 在具体实施方式中所描述的各个实施例中的各个具体技术特征，在不矛盾的情况下，可以进行各种组合，例如通过不同的具体技术特征的组合可以形成不同的实施方式，为了避免不必要的重复，本申请中各个具体技术特征的各种可能的组合方式不再另行说明。

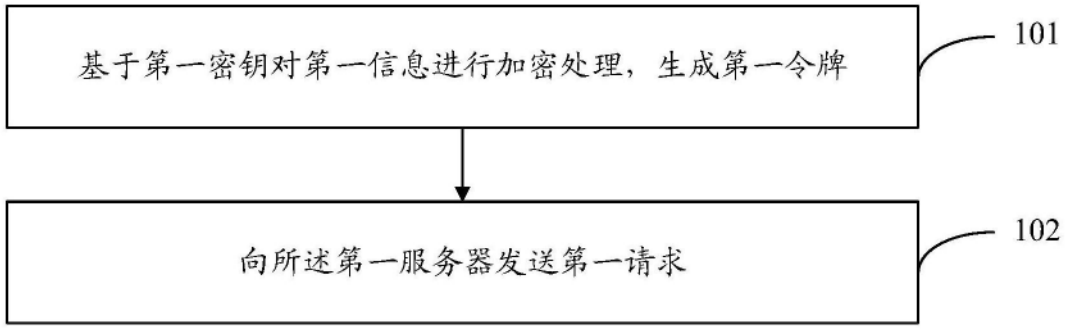

基于第一密钥对第一信息进行加密处理，生成第一令牌
101
向所述第一服务器发送第一请求
102

图1

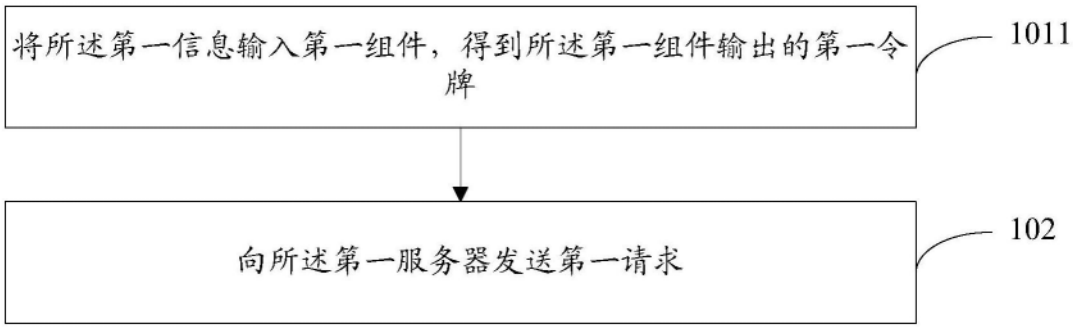

将所述第一信息输入第一组件，得到所述第一组件输出的第一令牌
1011
向所述第一服务器发送第一请求
102

图2

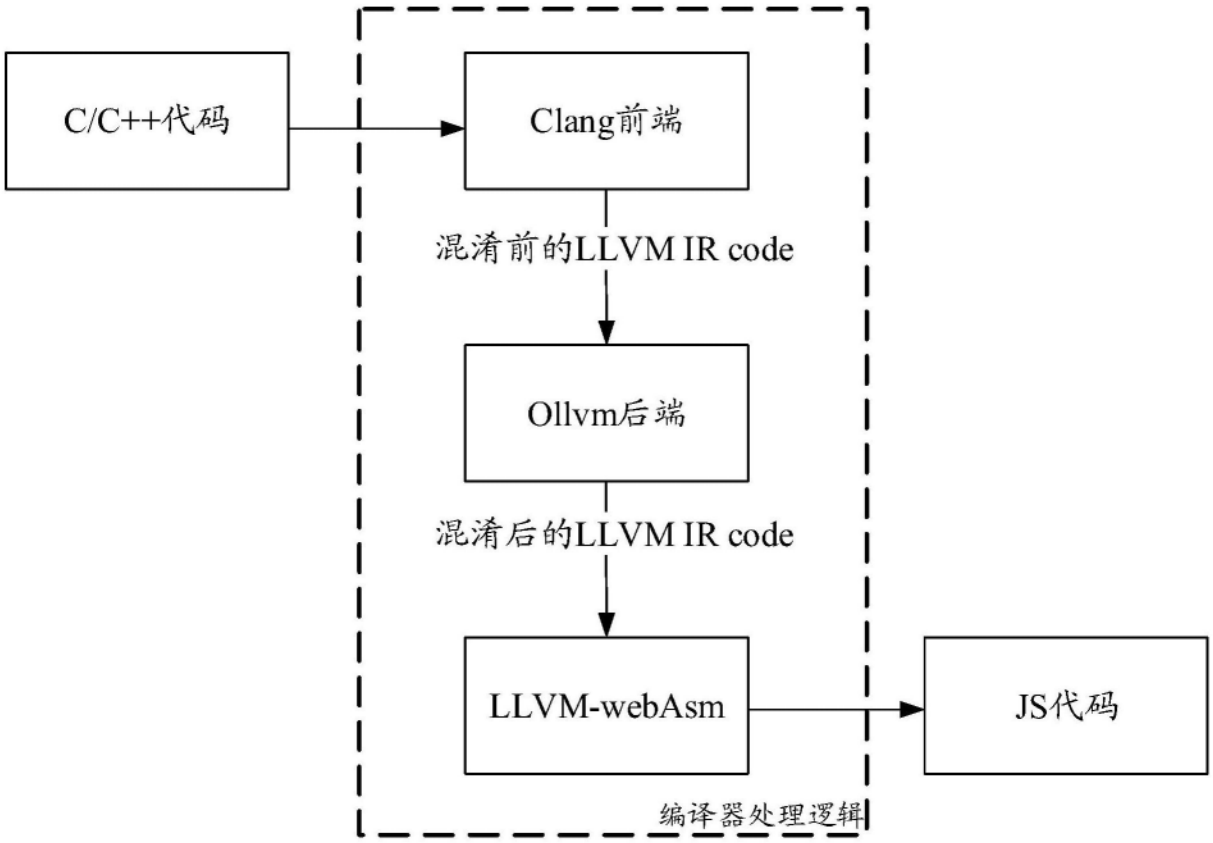

C/C++代码
Clang前端
混淆前的LLVM IR code
Ollvm后端
混淆后的LLVM IR code
LLVM-webAsm
JS代码
编译器处理逻辑

图3

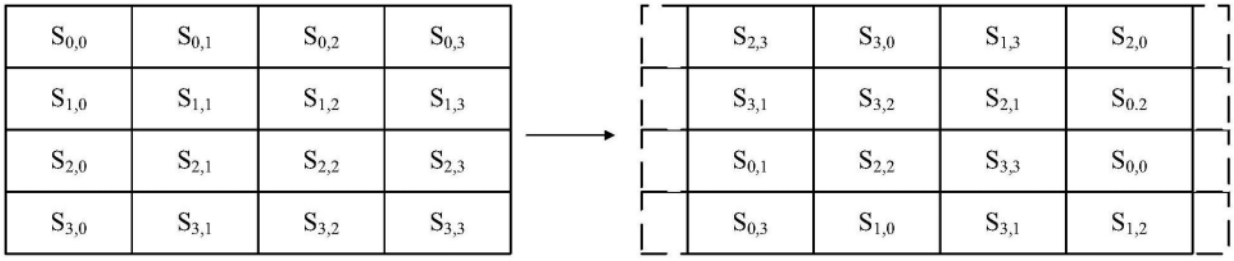

S0,0 S0,1 S0,2 S0,3
S1,0 S1,1 S1,2 S1,3
S2,0 S2,1 S2,2 S2,3
S3,0 S3,1 S3,2 S3,3
S2,3 S3,0 S1,3 S2,0
S3,1 S3,2 S2,1 S0.2
S0,1 S2,2 S3,3 S0,0
S0,3 S1,0 S3,1 S1,2

图4

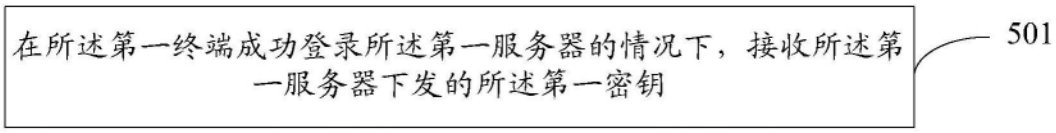

在所述第一终端成功登录所述第一服务器的情况下，接收所述第一服务器下发的所述第一密钥
501

图5

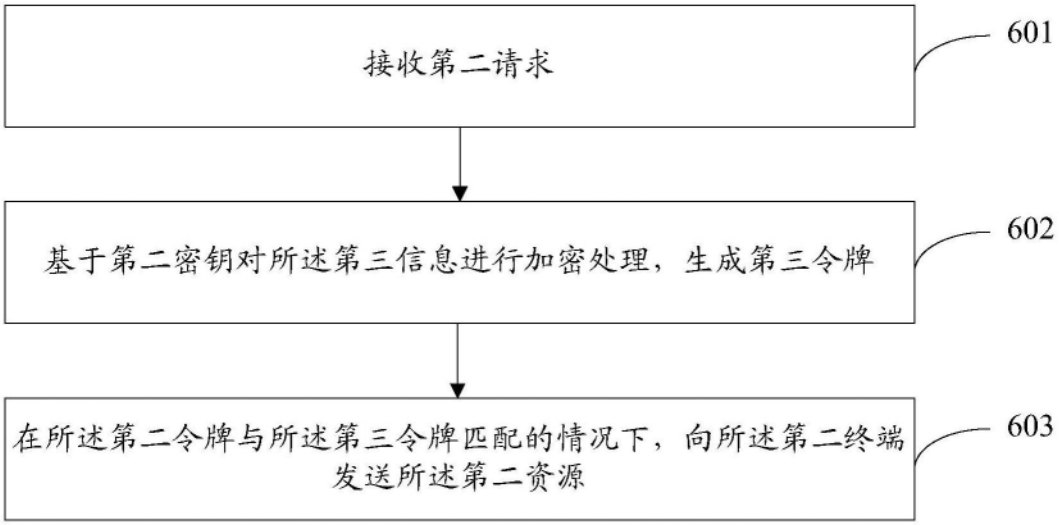

接收第二请求
601
基于第二密钥对所述第三信息进行加密处理，生成第三令牌
602
在所述第二令牌与所述第三令牌匹配的情况下，向所述第二终端发送所述第二资源
603

图6

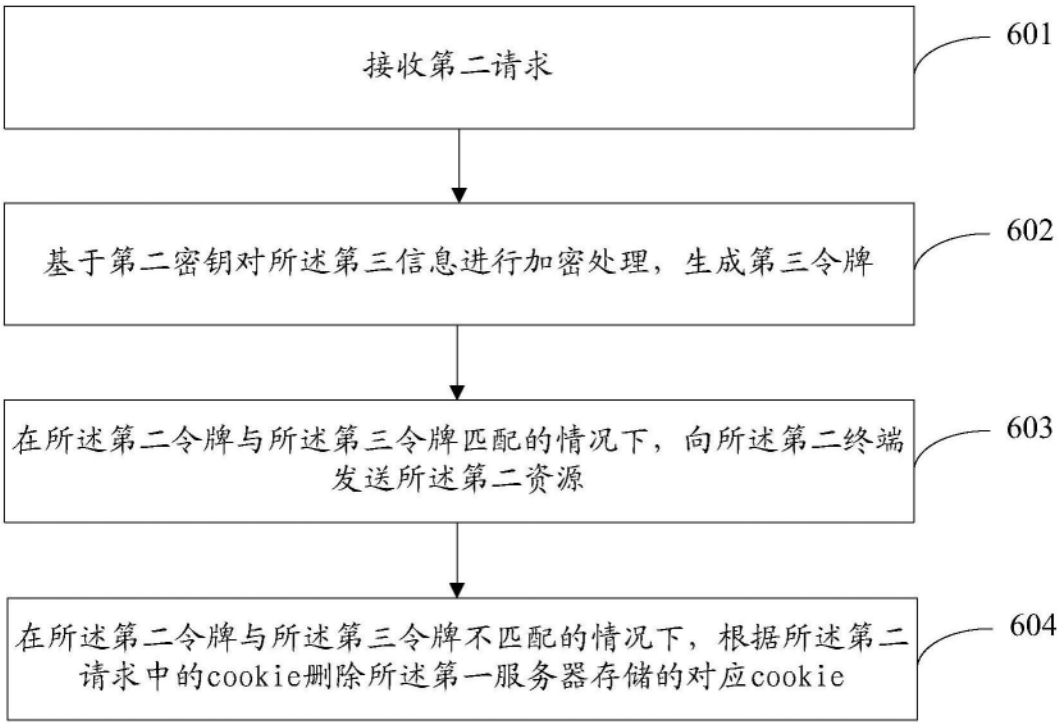

接收第二请求
601
基于第二密钥对所述第三信息进行加密处理，生成第三令牌
602
在所述第二令牌与所述第三令牌匹配的情况下，向所述第二终端发送所述第二资源
603
在所述第二令牌与所述第三令牌不匹配的情况下，根据所述第二请求中的cookie删除所述第一服务器存储的对应cookie
604

图7

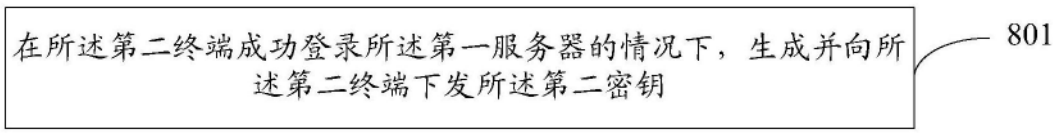

在所述第二终端成功登录所述第一服务器的情况下，生成并向所述第二终端下发所述第二密钥
801

图8

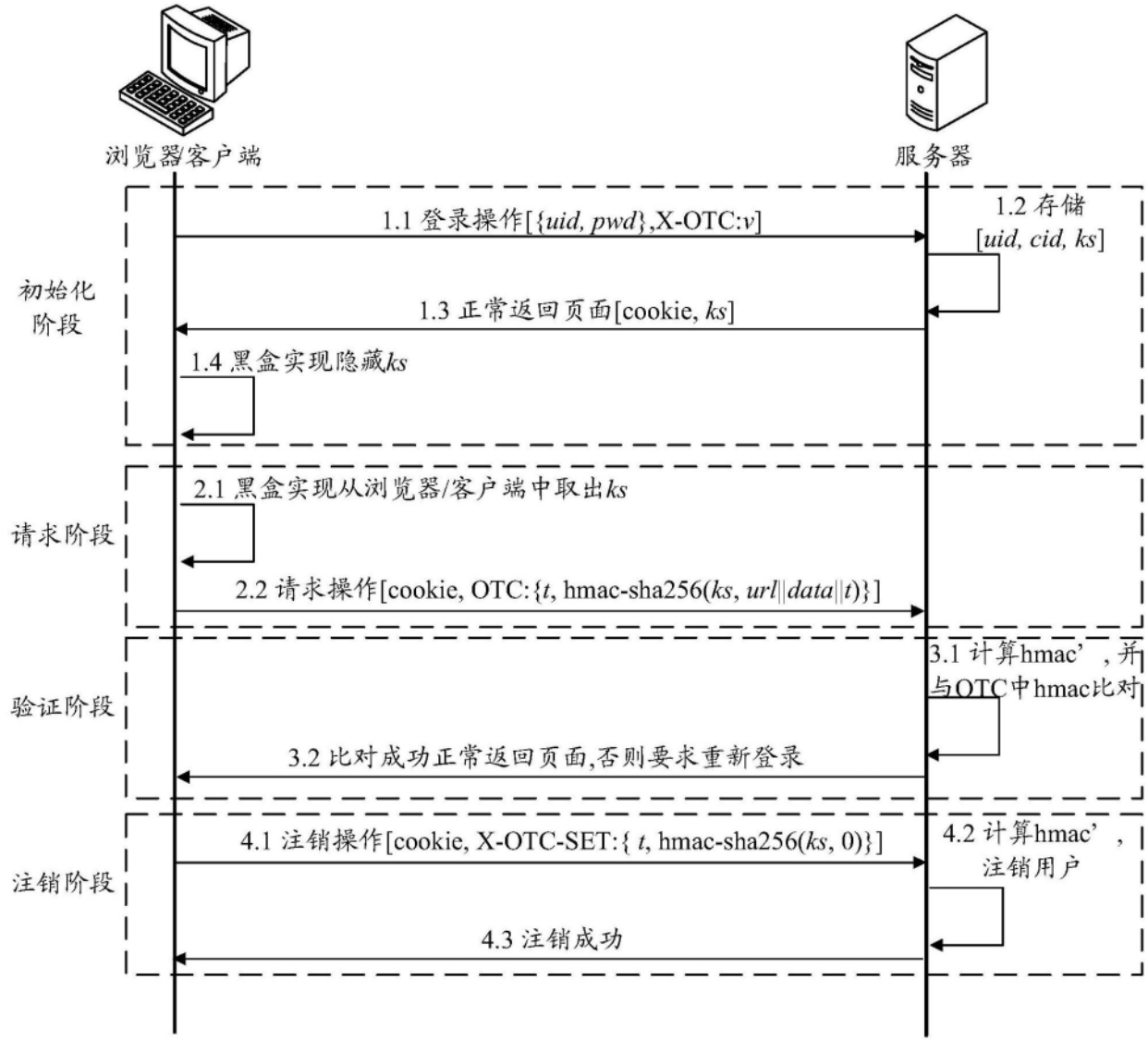

浏览器/客户端
服务器
1.1 登录操作[{uid, pwd},X-OTC:v]
1.2 存储[uid, cid, ks]
初始化阶段
1.3 正常返回页面[cookie, ks]
1.4 黑盒实现隐藏ks
2.1 黑盒实现从浏览器/客户端中取出ks
请求阶段
2.2 请求操作[cookie, OTC:{t, hmac-sha256(ks, url||data||t)}]
3.1 计算hmac'，并与OTC中hmac比对
验证阶段
3.2 比对成功正常返回页面,否则要求重新登录
4.1 注销操作[cookie, X-OTC-SET:{ t, hmac-sha256(ks, 0)}]
4.2 计算hmac'，注销用户
注销阶段
4.3 注销成功

图9

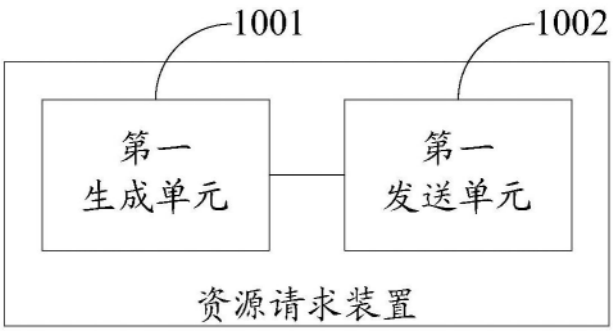

1001
1002
第一生成单元
第一发送单元
资源请求装置

图10

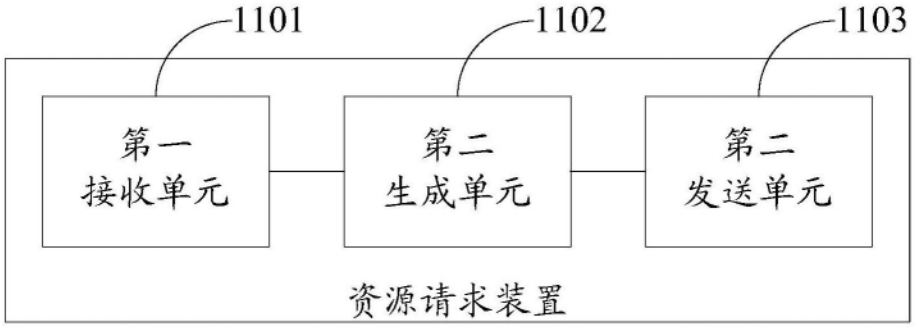

1101
1102
1103
第一
接收单元
第二
生成单元
第二
发送单元
资源请求装置

图11

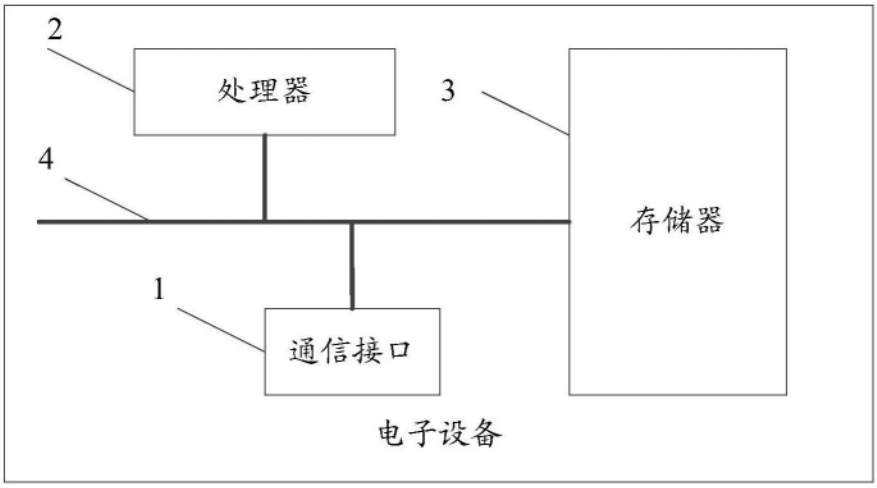

2
处理器
3
4
存储器
1
通信接口
电子设备

图12