



(12) 发明专利申请

(10) 申请公布号 CN 105119888 A

(43) 申请公布日 2015. 12. 02

(21) 申请号 201510406152. 6

(22) 申请日 2015. 07. 10

(71) 申请人 小米科技有限责任公司

地址 100085 北京市海淀区清河中街 68 号
华润五彩城购物中心二期 13 层

(72) 发明人 陈灏 任桥 李飞云

(74) 专利代理机构 北京尚伦律师事务所 11477

代理人 代治国

(51) Int. Cl.

H04L 29/06(2006. 01)

H04L 9/32(2006. 01)

G06F 21/51(2013. 01)

G06F 21/64(2013. 01)

G06F 9/445(2006. 01)

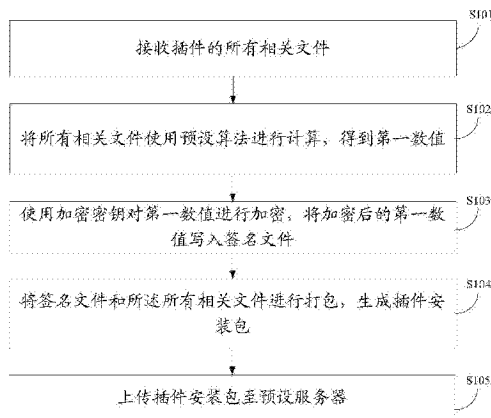
权利要求书4页 说明书16页 附图16页

(54) 发明名称

插件安装包上传方法、安装方法及装置

(57) 摘要

本公开是关于一种插件安装包上传方法、安装方法及装置。其中,插件上传方法包括:接收插件的所有相关文件;将所有相关文件使用预设算法进行计算,得到第一数值;使用加密密钥对第一数值进行加密,将加密后的第一数值写入签名文件;将签名文件和所述所有相关文件进行打包,生成插件安装包,上传至预设服务器。通过该技术方案,将通过对插件的所有相关文件进行计算得到的第一数值写入签名文件,将签名文件和插件的所有相关文件打包成插件安装包上传,在获取到插件安装包时,重新根据插件的所有相关文件计算,得到第二数值,将第一数值与第二数值进行对比,即可判断插件安装包里的文件是否被篡改,从而保证插件安装包的安装安全,提升用户使用体验。



1. 一种插件安装包上传方法,其特征在于,所述方法包括:
接收插件的所有相关文件;
将所述所有相关文件使用预设算法进行计算,得到第一数值;
使用加密密钥对所述第一数值进行加密,将所述加密后的第一数值写入签名文件;
将签名文件和所述所有相关文件进行打包,生成插件安装包;
上传所述插件安装包至预设服务器。
2. 根据权利要求 1 所述的方法,其特征在于,所述预设算法包括哈希算法。
3. 根据权利要求 1 所述的方法,其特征在于,所述方法还包括:
接收输入的密钥生成命令;
根据所述密钥生成命令,生成所述加密密钥和与所述加密密钥对应的解密密钥。
4. 根据权利要求 3 所述的方法,其特征在于,所述方法还包括:
上传所述解密密钥至所述预设服务器,所述预设服务器用于将所述插件安装包和所述解密密钥进行对应存储。
5. 根据权利要求 3 所述的方法,其特征在于,所述生成插件安装包,包括:
将所述解密密钥、签名文件、所述所有相关文件进行打包生成插件安装包。
6. 根据权利要求 1 至 5 中任一项所述的方法,其特征在于,所述方法还包括:
上传所述插件安装包的属性信息至所述预设服务器,其中,所述属性信息包括下述信息中的一项或多项:
插件安装包的名称、插件类型、插件所支持的最低应用版本。
7. 一种插件安装包安装方法,其特征在于,所述方法包括:
在从预设服务器获取到插件安装包后,获取所述插件安装包对应的解密密钥,其中,所述插件安装包中包括签名文件和插件的所有相关文件,所述解密密钥与所述签名文件中的加密后的第一数值的加密密钥相对应;
使用所述解密密钥对所述加密后的第一数值进行解密,得到第一数值;
将所述所有相关文件使用预设算法进行计算,得到第二数值;
将所述第一数值和所述第二数值进行比较,得到比较结果;
根据比较结果确定是否允许安装所述插件安装包。
8. 根据权利要求 7 所述的方法,其特征在于,所述预设算法包括哈希算法。
9. 根据权利要求 7 所述的方法,其特征在于,所述根据比较结果确定是否允许安装所述插件安装包,包括:
在所述第一数值与所述第二数值相同时,确定所述插件的所有相关文件未被篡改,允许安装所述插件安装包;
在所述第一数值与所述第二数值不同时,确定所述插件的所有相关文件被篡改,不允许安装所述插件安装包。
10. 根据权利要求 9 所述的方法,其特征在于,确定所述插件的所有相关文件未被篡改之后,还包括:
从所述预设服务器中获取所述插件安装包的属性信息,其中,所述属性信息包括下述信息中的一项或多项:插件安装包的名称、插件类型、插件支持的最低应用版本号;
判断欲安装的所述插件安装包的应用的当前版本号;

在所述当前版本号不低于所述最低应用版本号时,转向允许安装所述插件安装包的步骤;

在所述当前版本低于所述最低应用版本号时,转向不允许安装所述插件安装包的步骤。

11. 根据权利要求7至10中任一项所述的方法,其特征在于,所述在从预设服务器获取到插件安装包后,获取所述插件安装包对应的解密密钥,包括:

在从所述预设服务器获取到插件安装包后,发送所述插件安装包对应的解密密钥的获取请求至所述预设服务器;

接收所述预设服务器发送的与所述插件安装包对应的解密密钥;或者

在从所述预设服务器获取到插件安装包后,对所述插件安装包进行解析获取其中包含的解密密钥。

12. 一种插件安装包上传装置,其特征在于,所述装置包括:

第一接收模块,用于接收插件的所有相关文件;

计算模块,用于将所述所有相关文件使用预设算法进行计算,得到第一数值;

加密模块,用于使用加密密钥对所述第一数值进行加密,将所述加密后的第一数值写入签名文件;

第一生成模块,用于将签名文件和所述所有相关文件进行打包,生成插件安装包;

第一上传模块,用于上传所述插件安装包至预设服务器。

13. 根据权利要求12所述的装置,其特征在于,所述预设算法包括哈希算法。

14. 根据权利要求12所述的装置,其特征在于,所述装置还包括:

第二接收模块,用于接收输入的密钥生成命令;

第二生成模块,用于根据所述密钥生成命令,生成所述加密密钥和与所述加密密钥对应的解密密钥。

15. 根据权利要求14所述的装置,其特征在于,所述装置还包括:

第二上传模块,用于上传所述解密密钥至所述预设服务器,所述预设服务器用于将所述插件安装包和所述解密密钥进行对应存储。

16. 根据权利要求14所述的装置,其特征在于,

所述第一生成模块包括:

安装包生成子模块,用于将所述解密密钥、签名文件、所述所有相关文件进行打包生成插件安装包。

17. 根据权利要求12至16中任一项所述的装置,其特征在于,所述装置还包括:

第三上传模块,用于上传所述插件安装包的属性信息至所述预设服务器,其中,所述属性信息包括下述信息中的一项或多项:

插件安装包的名称、插件类型、插件所支持的最低应用版本。

18. 一种插件安装包安装装置,其特征在于,所述装置包括:

密钥获取模块,用于在从预设服务器获取到插件安装包后,获取所述插件安装包对应的解密密钥,其中,所述插件安装包中包括签名文件和插件的所有相关文件,所述解密密钥与所述签名文件中的加密后的第一数值的加密密钥相对应;

解密模块,用于使用所述解密密钥对所述加密后的第一数值进行解密,得到第一数

值；

计算模块,用于将所述所有相关文件使用预设算法进行计算,得到第二数值；

比较模块,用于将所述第一数值和所述第二数值进行比较,得到比较结果；

确定模块,用于根据比较结果确定是否允许安装所述插件安装包。

19. 根据权利要求 18 所述的装置,其特征在于,所述预设算法包括哈希算法。

20. 根据权利要求 18 所述的装置,其特征在于,所述确定模块包括：

第一确定子模块,用于在所述第一数值与所述第二数值相同时,确定所述插件的所有相关文件未被篡改,允许安装所述插件安装包；

第二确定子模块,用于在所述第一数值与所述第二数值不同时,确定所述插件的所有相关文件被篡改,不允许安装所述插件安装包。

21. 根据权利要求 20 所述的装置,其特征在于,所述确定模块还包括：

信息获取子模块,用于在确定所述插件的所有相关文件未被篡改后,从所述预设服务器中获取所述插件安装包的属性信息,其中,所述属性信息包括下述信息中的一项或多项：插件安装包的名称、插件类型、插件支持的最低应用版本号；

判断子模块,用于判断欲安装的所述插件安装包的应用的当前版本号；

所述第一确定子模块用于,在所述当前版本号不低于所述最低应用版本号时,转向允许安装所述插件安装包的步骤；

所述第二确定子模块用于,在所述当前版本低于所述最低应用版本号时,转向不允许安装所述插件安装包的步骤。

22. 根据权利要求 18 至 21 中任一项所述的装置,其特征在于,所述密钥获取模块包括：

发送子模块,用于在从所述预设服务器获取到插件安装包后,发送所述插件安装包对应的解密密钥的获取请求至所述预设服务器；

接收子模块,用于接收所述预设服务器发送的与所述插件安装包对应的解密密钥；或者

解析子模块,用于在从所述预设服务器获取到插件安装包后,对所述插件安装包进行解析获取其中包含的解密密钥。

23. 一种插件安装包上传装置,其特征在于,包括：

处理器；

用于存储处理器可执行指令的存储器；

其中,所述处理器被配置为：

接收插件的所有相关文件；

将所述所有相关文件使用预设算法进行计算,得到第一数值；

使用加密密钥对所述第一数值进行加密,将所述加密后的第一数值写入签名文件；

将签名文件和所述所有相关文件进行打包,生成插件安装包；

上传所述插件安装包至预设服务器。

24. 一种插件安装包安装装置,其特征在于,包括：

处理器；

用于存储处理器可执行指令的存储器；

其中,所述处理器被配置为:

在从预设服务器获取到插件安装包后,获取所述插件安装包对应的解密密钥,其中,所述插件安装包中包括签名文件和插件的所有相关文件,所述解密密钥与所述签名文件中的加密后的第一数值的加密密钥相对应;

使用所述解密密钥对所述加密后的第一数值进行解密,得到第一数值;

将所述所有相关文件使用预设算法进行计算,得到第二数值;

将所述第一数值和所述第二数值进行比较,得到比较结果;

根据比较结果确定是否允许安装所述插件安装包。

插件安装包上传方法、安装方法及装置

技术领域

[0001] 本公开涉及插件安装技术领域，尤其涉及插件安装包上传方法、插件安装包安装方法及插件安装包上传装置、插件安装包安装装置。

背景技术

[0002] 插件化开发方式是当前手机开发过程中，一种比较不错的解决应用臃肿化的方案。开发者可以将应用按照功能、结构划分为多个模块，分别开发维护；用户也只需下载精巧的具备基本功能的应用，当有拓展功能需求时可以通过下载安装相关插件实现上述功能。

[0003] 开发者在完成插件软件开发后，需要将其上传到服务器，经过审核后才能够上线。

发明内容

[0004] 本公开实施例提供，包括如下技术方案：

[0005] 根据本公开实施例的第一方面，提供一种插件安装包上传方法，所述方法包括：

[0006] 接收插件的所有相关文件；

[0007] 将所述所有相关文件使用预设算法进行计算，得到第一数值；

[0008] 使用加密密钥对所述第一数值进行加密，将所述加密后的第一数值写入签名文件；

[0009] 将签名文件和所述所有相关文件进行打包，生成插件安装包；

[0010] 上传所述插件安装包至预设服务器。

[0011] 在该实施例中，使用签名文件对所有相关文件进行签名，得到插件安装包，这样，即使插件安装包里的相关文件被篡改，由于签名文件已进行加密，而没有解密密钥并不能对其进行操作，因此，在将插件安装包上传至预设服务器的过程中，可以保证签名文件的安全，进而根据签名文件判断插件安装包里的相关文件是否被篡改，从而保证用户安装插件安装包的安全，防止被第三方恶意篡改，提升用户的使用体验。

[0012] 在一个实施例中，所述预设算法包括哈希算法。

[0013] 在一个实施例中，所述方法还包括：

[0014] 接收输入的密钥生成命令；

[0015] 根据所述密钥生成命令，生成所述加密密钥和与所述加密密钥对应的解密密钥。

[0016] 在该实施例中，根据加密密钥生成与其对应的解密密钥，从而组成密钥对，实现对第一数值的加密和解密。其中，加密密钥可以是私钥，解密密钥可以是公钥。

[0017] 在一个实施例中，所述方法还包括：

[0018] 上传所述解密密钥至所述预设服务器，所述预设服务器用于将所述插件安装包和所述解密密钥进行对应存储。

[0019] 在该实施例中，将解密密钥上传至预设服务器，并与插件安装包对应存储，这样，应用程序或终端设备等在欲安装插件安装包时，可以向预设服务器发送解密密钥获取请

求,从而获取解密密钥,对加密后的签名文件进行解密,验证插件安装包是否被伪造或被篡改。

[0020] 在一个实施例中,生成插件安装包,包括:

[0021] 将所述解密密钥、签名文件、所述所有相关文件进行打包生成插件安装包。

[0022] 在该实施例中,还可以将解密密钥和签名后的所有相关文件一起打包生成插件安装包,这样,在应用程序或终端设备等在欲安装插件安装包时,可以直接从插件安装包中解析出解密密钥,进而根据解密密钥对加密后的签名文件进行解密,验证插件安装包是否被伪造或被篡改。本公开给出了多种解密密钥的提供方式,使得插件安装包上传方法的实施方式更丰富、灵活。

[0023] 在一个实施例中,所述方法还包括:

[0024] 上传所述插件安装包的属性信息至所述预设服务器,其中,所述属性信息中包括下述一项或多项:

[0025] 插件安装包的名称、插件类型、插件所支持的最低应用版本。

[0026] 在该实施例中,上传插件安装包的属性信息至预设服务器,如上传插件安装包名称、插件类型、插件所支持的最低应用版本等至预设服务器,这样,在应用程序或终端设备等在欲安装插件安装包时,目标使用用户可以清楚的获知插件的详细信息,同时也方便应用程序或终端设备等根据插件类型、名称、所支持的最低版本等信息识别和正确安装该插件,从而避免因插件不能安装给用户带来的困扰,提升用户的使用体验。

[0027] 根据本公开实施例的第二方面,提供一种插件安装包安装方法,所述方法包括:

[0028] 在从预设服务器获取到插件安装包后,获取所述插件安装包对应的解密密钥,其中,所述插件安装包中包括插件的签名文件和所有相关文件,所述解密密钥与所述签名文件的加密密钥相对应;

[0029] 使用所述解密密钥对所述加密后的第一数值进行解密,得到第一数值;

[0030] 将所述所有相关文件使用预设算法进行计算,得到第二数值;

[0031] 将所述第一数值和所述第二数值进行比较,得到比较结果;

[0032] 根据比较结果确定是否允许安装所述插件安装包。

[0033] 在该实施例中,使用签名文件对所有相关文件进行签名,得到插件安装包,这样,即使插件安装包里的相关文件被篡改,由于签名文件已进行加密,而没有解密密钥并不能对其进行操作,因此,在将插件安装包上传至预设服务器的过程中,可以保证签名文件的安全。而在对插件安装包进行安装时,可以对所有相关文件进行计算,从而判断计算得到的数值,是否与从签名文件中解密得到的数值是否相同,如果两者相同,则说明插件安装包中的所有相关文件并没有被篡改,如果两者不同,则说明插件安装包中的所有相关文件被伪造或篡改。从而有效地保证用户安装插件安装包的安全,防止被第三方恶意篡改。

[0034] 在一个实施例中,所述预设算法包括哈希算法。

[0035] 在一个实施例中,所述根据比较结果确定是否允许安装所述插件安装包,包括:

[0036] 在所述第一数值与所述第二数值相同时,确定所述插件的所有相关文件未被篡改,允许安装所述插件安装包;

[0037] 在所述第一数值与所述第二数值不同时,确定所述插件的所有相关文件被篡改,不允许安装所述插件安装包。

[0038] 在该实施例中,根据第一数值和第二数值是否相同,来判断插件的所有相关文件是否被篡改,从而确定插件安装包是否安全,在安全时,允许安装插件安装包,在不安全时,不允许安装插件安装包,从而保证用户的安装安全,防止被第三方恶意篡改,提升用户的使用体验。

[0039] 在一个实施例中,确定所述插件的所有相关文件未被篡改之后,还包括:

[0040] 从所述预设服务器中获取所述插件安装包的属性信息,其中,所述属性信息包括下述信息中的一项或多项:插件安装包的名称、插件类型、插件支持的最低应用版本号;

[0041] 判断欲安装的所述插件安装包的应用的当前版本号;

[0042] 在所述当前版本号不低于所述最低应用版本号时,转向允许安装所述插件安装包的步骤;

[0043] 在所述当前版本低于所述最低应用版本号时,转向不允许安装所述插件安装包的步骤。

[0044] 在该实施例中,上传插件安装包的属性信息至预设服务器,如上传插件安装包名称、插件类型、插件所支持的最低应用版本等至预设服务器,这样,在应用程序或终端设备等在欲安装插件安装包时,方向用户清楚的获知插件的详细信息,同时也方便应用程序或终端设备等根据插件所支持的最低版本判断是否可以安装该插件,从而避免因插件不能安装给用户带来的困扰,提升用户的使用体验。

[0045] 在一个实施例中,所述在从预设服务器获取到插件安装包后,获取所述插件安装包对应的解密密钥,包括:

[0046] 在从所述预设服务器获取到插件安装包后,发送所述插件安装包对应的解密密钥的获取请求至所述预设服务器;

[0047] 接收所述预设服务器发送的与所述插件安装包对应的解密密钥;或者

[0048] 在从所述预设服务器获取到插件安装包后,对所述插件安装包进行解析获取其中包含的解密密钥。

[0049] 在该实施例中,应用程序或终端设备等在欲安装插件安装包时,可以向预设服务器发送解密密钥获取请求,从而获取解密密钥,对加密后的签名文件进行解密,验证插件安装包是否被伪造或被篡改。也可以在上传时,将解密密钥和签名后的所有相关文件一起打包生成插件安装包,这样,在应用程序或终端设备等在欲安装插件安装包时,可以直接从插件安装包中解析出解密密钥,进而根据解密密钥对加密后的签名文件进行解密,验证插件安装包是否被伪造或被篡改。

[0050] 根据本公开实施例的第三方面,提供一种插件安装包上传装置,所述装置包括:

[0051] 第一接收模块,用于接收插件的所有相关文件;

[0052] 计算模块,用于将所述所有相关文件使用预设算法进行计算,得到第一数值;

[0053] 加密模块,用于使用加密密钥对所述第一数值进行加密,将所述加密后的第一数值写入签名文件;

[0054] 第一生成模块,用于将签名文件和所述所有相关文件进行打包,生成插件安装包;

[0055] 第一上传模块,用于上传所述插件安装包至预设服务器。

[0056] 在一个实施例中,所述预设算法包括哈希算法。

- [0057] 在一个实施例中,所述装置还包括:
- [0058] 第二接收模块,用于接收输入的密钥生成命令;
- [0059] 第二生成模块,用于根据所述密钥生成命令,生成所述加密密钥和与所述加密密钥对应的解密密钥。
- [0060] 在一个实施例中,所述装置还包括:
- [0061] 第二上传模块,用于上传所述解密密钥至所述预设服务器,所述预设服务器用于将所述插件安装包和所述解密密钥进行对应存储;或者
- [0062] 所述第一生成模块包括:
- [0063] 安装包生成子模块,用于将所述解密密钥、签名文件、所述所有相关文件进行打包生成插件安装包。
- [0064] 在一个实施例中,所述装置还包括:
- [0065] 第三上传模块,用于上传所述插件安装包的属性信息至所述预设服务器,其中,所述属性信息中包括下述一项或多项:
- [0066] 插件安装包的名称、插件类型、插件所支持的最低应用版本。
- [0067] 根据本公开实施例的第四方面,提供一种插件安装包安装装置,所述装置包括:
- [0068] 密钥获取模块,用于在从预设服务器获取到插件安装包后,获取所述插件安装包对应的解密密钥,其中,所述插件安装包中包括插件的签名文件和所有相关文件,所述解密密钥与所述签名文件的加密密钥相对应;
- [0069] 加密模块,用于使用所述解密密钥对所述加密后的第一数值进行解密,得到第一数值;
- [0070] 计算模块,用于将所述所有相关文件使用预设算法进行计算,得到第二数值;
- [0071] 比较模块,用于将所述第一数值和所述第二数值进行比较,得到比较结果;
- [0072] 确定模块,用于根据比较结果确定是否允许安装所述插件安装包。
- [0073] 在一个实施例中,所述预设算法包括哈希算法。
- [0074] 在一个实施例中,所述确定模块包括:
- [0075] 第一确定子模块,用于在所述第一数值与所述第二数值相同时,确定所述插件的所有相关文件未被篡改,允许安装所述插件安装包;
- [0076] 第二确定子模块,用于在所述第一数值与所述第二数值不同时,确定所述插件的所有相关文件被篡改,不允许安装所述插件安装包。
- [0077] 在一个实施例中,所述确定模块还包括:
- [0078] 信息获取子模块,用于在确定所述插件的所有相关文件未被篡改后,从所述预设服务器中获取所述插件安装包的属性信息,其中,所述属性信息包括下述信息中的一项或多项:插件安装包的名称、插件类型、插件支持的最低应用版本号;
- [0079] 判断子模块,用于判断欲安装的所述插件安装包的应用的当前版本号;
- [0080] 所述第一确定子模块用于,在所述当前版本号不低于所述最低应用版本号时,转向允许安装所述插件安装包的步骤;
- [0081] 所述第二确定子模块用于,在所述当前版本低于所述最低应用版本号时,转向不允许安装所述插件安装包的步骤。
- [0082] 在一个实施例中,所述密钥获取模块包括:

- [0083] 发送子模块,用于在从所述预设服务器获取到插件安装包后,发送所述插件安装包对应的解密密钥的获取请求至所述预设服务器;
- [0084] 接收子模块,用于接收所述预设服务器发送的与所述插件安装包对应的解密密钥;或者
- [0085] 解析子模块,用于在从所述预设服务器获取到插件安装包后,对所述插件安装包进行解析获取其中包含的解密密钥。
- [0086] 根据本公开实施例的第五方面,提供一种插件安装包上传装置,所述装置包括:
- [0087] 处理器;
- [0088] 用于存储处理器可执行指令的存储器;
- [0089] 其中,所述处理器被配置为:
- [0090] 接收插件的所有相关文件;
- [0091] 将所述所有相关文件使用预设算法进行计算,得到第一数值;
- [0092] 使用加密密钥对所述第一数值进行加密,将所述加密后的第一数值写入签名文件;
- [0093] 将签名文件和所述所有相关文件进行打包,生成插件安装包;
- [0094] 上传所述插件安装包至预设服务器。
- [0095] 根据本公开实施例的第六方面,提供一种插件安装包安装装置,所述装置包括:
- [0096] 处理器;
- [0097] 用于存储处理器可执行指令的存储器;
- [0098] 其中,所述处理器被配置为:
- [0099] 在从预设服务器获取到插件安装包后,获取所述插件安装包对应的解密密钥,其中,所述插件安装包中包括插件的签名文件和所有相关文件,所述解密密钥与所述签名文件的加密密钥相对应;
- [0100] 使用所述解密密钥对所述加密后的第一数值进行解密,得到第一数值;
- [0101] 将所述所有相关文件使用预设算法进行计算,得到第二数值;
- [0102] 将所述第一数值和所述第二数值进行比较,得到比较结果;
- [0103] 根据比较结果确定是否允许安装所述插件安装包。
- [0104] 本公开的实施例提供的技术方案可以包括以下有益效果:
- [0105] 上述技术方案,将通过对插件的所有相关文件进行计算得到的第一数值写入签名文件,将签名文件和插件的所有相关文件打包成插件安装包上传,并在获取到插件安装包时,重新根据插件的所有相关文件计算,得到第二数值,将第一数值与第二数值进行对比,即可判断插件安装包里的文件是否被篡改,从而保证插件安装包的安装安全,防止被第三方恶意篡改,提升用户的使用体验。
- [0106] 应当理解的是,以上的一般描述和后文的细节描述仅是示例性和解释性的,并不能限制本公开。

附图说明

- [0107] 此处的附图被并入说明书中并构成本说明书的一部分,示出了符合本公开的实施例,并与说明书一起用于解释本公开的原理。

- [0108] 图 1 是根据一示例性实施例示出的插件安装包上传方法的流程图。
- [0109] 图 2 是根据一示例性实施例示出的一种插件安装包上传方法的流程图。
- [0110] 图 3 是根据一示例性实施例示出的另一种插件安装包上传方法的流程图。
- [0111] 图 4 是根据一示例性实施例示出的再一种插件安装包上传方法的流程图。
- [0112] 图 5A 是根据一示例性实施例示出的又一种插件安装包上传方法的流程图。
- [0113] 图 5B 是根据一示例性实施例示出的插件安装包管理界面示意图。
- [0114] 图 6 是根据一示例性实施例示出的插件安装包安装方法的流程图。
- [0115] 图 7 是根据一示例性实施例示出的一种插件安装包安装方法的流程图。
- [0116] 图 8 是根据一示例性实施例示出的另一种插件安装包安装方法的流程图。
- [0117] 图 9 是根据一示例性实施例示出的再一种插件安装包安装方法的流程图。
- [0118] 图 10 是根据一示例性实施例示出的又一种插件安装包安装方法的流程图。
- [0119] 图 11 是根据一示例性实施例示出的插件安装包上传装置的框图。
- [0120] 图 12 是根据一示例性实施例示出的一种插件安装包上传装置的框图。
- [0121] 图 13 是根据一示例性实施例示出的另一种插件安装包上传装置的框图。
- [0122] 图 14 是根据一示例性实施例示出的一种插件安装包上传装置中第一生成模块的框图。
- [0123] 图 15 是根据一示例性实施例示出的再一种插件安装包上传装置的框图。
- [0124] 图 16 是根据一示例性实施例示出的插件安装包安装装置的框图。
- [0125] 图 17 是根据一示例性实施例一示出的插件安装包安装装置中确定模块的框图。
- [0126] 图 18 是根据一示例性实施例二示出的插件安装包安装装置中确定模块的框图。
- [0127] 图 19 是根据一示例性实施例示出的插件安装包安装装置中密钥获取模块的框图。
- [0128] 图 20 是根据一示例性实施例示出的适用于插件安装包上传装置和插件安装包安装装置的框图。

具体实施方式

[0129] 这里将详细地对示例性实施例进行说明,其示例表示在附图中。下面的描述涉及附图时,除非另有表示,不同附图中的相同数字表示相同或相似的要素。以下示例性实施例中所描述的实施方式并不代表与本公开相一致的所有实施方式。相反,它们仅是与如所附权利要求书中所详述的、本公开的一些方面相一致的装置和方法的例子。

[0130] 本公开实施例提供了一种插件安装包上传方法,如图 1 所示,该方法包括步骤 S101-S105:

[0131] 在步骤 S101 中,接收插件的所有相关文件;

[0132] 在步骤 S102 中,将所有相关文件使用预设算法进行计算,得到第一数值。在一个实施例中,预设算法包括哈希算法。当然,预设算法也可以是其他可对所有相关文件进行加密的加密算法,例如 RSA 算法、DES(Data Encryption Standard,数据加密标准)等。

[0133] 在步骤 S103 中,使用加密密钥对第一数值进行加密,将加密后的第一数值写入签名文件。使用加密密钥进行加密,这样可以保证第一数值的安全,保证其不会被篡改或被轻易获取。

[0134] 在步骤 S104 中,将签名文件和所述所有相关文件进行打包,生成插件安装包。

[0135] 在该实施例中,将签名文件和插件的所有相关文件进行打包,生成插件安装包,即进行数字签名。数字签名可以用来证实消息确实是由发送者签发的,而且当数字签名用于存储数据或程序时,可以用来验证数据或程序的完整性。它和传统手写签名类似,应具有以下特征:(1) 不可伪造性:除了签名者外,任何人都不能伪造签名者的合法签名。(2) 认证性:接收者相信这份签名来自签名者。(3) 不可重用性:一个消息的签名不能用于其他消息。(4) 不可修改性:一个消息在签名后不能被修改。(5) 不可抵赖性:签名者事后不能否认自己的签名。

[0136] 在步骤 S105 中,上传插件安装包至预设服务器。将签名文件和所有相关文件进行打包,得到插件安装包,这样,即使插件安装包里的相关文件被篡改,由于签名文件中的第一数值已进行加密,而没有解密密钥并不能对其进行操作,因此,在将插件安装包上传至预设服务器的过程中,可以保证签名文件的安全,进而根据签名文件判断插件安装包里的相关文件是否被篡改,从而有效地保证用户安装插件安装包的安全,防止被第三方恶意篡改,提升用户的使用体验。

[0137] 如图 2 所示,在一个实施例中,上述插件安装包上传方法还包括步骤 S201-S202:

[0138] 在步骤 S201 中,接收输入的密钥生成命令;

[0139] 在步骤 S202 中,根据密钥生成命令,生成加密密钥和与加密密钥对应的解密密钥。

[0140] 在该实施例中,根据加密密钥生成与其对应的解密密钥,从而组成密钥对,实现对第一数值的加密和解密。其中,加密密钥可以是私钥,解密密钥可以是公钥。其中,公钥和私钥是通过一种算法得到的一个密钥对(即一个公钥和一个私钥)其中的一个向外界公开,称为公钥;另一个自己保留,称为私钥。通过这种算法得到的密钥对能保证在世界范围内是唯一的。使用这个密钥对的时候,如果用其中一个密钥加密一段数据,必须用另一个密钥解密。比如用公钥加密数据就必须用私钥解密,如果用私钥加密也必须用公钥解密,否则解密将不会成功。

[0141] 使用公钥和私钥配对的方式,实现插件安装包的加密上传和解密安装,可以有效地保证插件安装包不被第三方篡改,提高安全性。

[0142] 其中,解密密钥可以通过下述两种方式上传至预设服务器。

[0143] 方式一:

[0144] 如图 3 所示,在一个实施例中,上述方法还包括步骤 S301:

[0145] 在步骤 S301 中,上传解密密钥至预设服务器,预设服务器用于将插件安装包和解密密钥进行对应存储。

[0146] 在该实施例中,将解密密钥上传至预设服务器,并与插件安装包对应存储,这样,应用程序或终端设备等在欲安装插件安装包时,可以向预设服务器发送解密密钥获取请求,从而从服务器侧获取解密密钥,对加密后的签名文件进行解密,验证插件安装包是否被伪造或被篡改。

[0147] 解密密钥由服务器保存,由于服务器的权威性和安全性,可以充分保证解密密钥的安全性。

[0148] 在服务器侧保存的解密密钥,需要加密后保存,避免明文保存,在服务器根据解密

密钥获取请求,返回解密密钥时,同样需要将加密之后的解密密钥返回给服务器,避免明文传输,关于解密密钥如何解密,可以预先预定对应的密码。

[0149] 方式二:

[0150] 如图 4 所示,在一个实施例中,上述步骤 S104 包括步骤 S401:

[0151] 在步骤 S401 中,将解密密钥、签名文件、所有相关文件进行打包生成插件安装包。

[0152] 在该实施例中,将解密密钥、签名文件和所有相关文件一起打包生成插件安装包,这样,在应用程序或终端设备等在欲安装插件安装包时,可以直接从插件安装包中解析出解密密钥,进而根据解密密钥对签名文件中加密后的第一数值进行解密,验证插件安装包是否被伪造或被篡改。

[0153] 如图 5A 所示,在一个实施例中,上述方法还包括步骤 S501:

[0154] 在步骤 S501 中,上传插件安装包的属性信息至预设服务器,其中,属性信息中包括下述一项或多项:

[0155] 插件安装包的名称、插件类型、插件所支持的最低应用版本。

[0156] 在该实施例中,如图 5B 所示,在上传插件安装包之前,还可以输出提示界面,提示插件开发者输入如插件安装包名称、插件类型、插件所支持的最低应用版本等信息,进而将这些信息一同上传至预设服务器,这样,在应用程序或终端设备等在欲安装插件安装包时,目标使用用户可以清楚的获知插件的详细信息,同时也方便应用程序或终端设备等根据插件类型、名称、所支持的最低版本等信息识别和正确安装该插件,从而避免因插件不能安装给用户带来的困扰,提升用户的使用体验。

[0157] 本公开实施例还提供了一种插件安装包安装方法,该方法可用于需要安装插件的应用程序或终端设备中,如图 6 所示,该方法包括步骤 S601-S605:

[0158] 在步骤 S601 中,在从预设服务器获取到插件安装包后,获取插件安装包对应的解密密钥,其中,插件安装包中包括插件的签名文件和所有相关文件,解密密钥与签名文件的加密密钥相对应;

[0159] 在步骤 S602 中,使用解密密钥对加密后的第一数值进行解密,得到第一数值;

[0160] 在步骤 S603 中,将所有相关文件使用预设算法进行计算,得到第二数值;在一个实施例中,预设算法包括哈希算法。当然,预设算法也可以是其他可对所有相关文件计算的算法,该算法与在上传插件安装包之前,对所有相关文件进行计算得到第一数值的算法相同。

[0161] 在步骤 S604 中,将第一数值和第二数值进行比较,得到比较结果;

[0162] 在步骤 S605 中,根据比较结果确定是否允许安装插件安装包。

[0163] 在该实施例中,将签名文件和所有相关文件进行打包,得到插件安装包,这样,即使插件安装包里的相关文件被篡改,由于签名文件中的第一数值已进行加密,而没有解密密钥并不能对其进行操作,因此,在将插件安装包上传至预设服务器的过程中,可以保证签名文件的安全。而在对插件安装包进行安装时,可以对所有相关文件进行计算,从而判断计算得到的数值,是否与从签名文件中解密得到的数值是否相同,如果两者相同,则说明插件安装包中的所有相关文件并没有被篡改,如果两者不同,则说明插件安装包中的所有相关文件被伪造或篡改。从而保证用户安装插件安装包的安全。

[0164] 如图 7 所示,在一个实施例中,上述步骤 S605 包括步骤 S701-S702:

[0165] 在步骤 S701 中,在第一数值与第二数值相同时,确定插件的所有相关文件未被篡改,允许安装插件安装包;

[0166] 在步骤 S702 中,在第一数值与第二数值不同时,确定插件的所有相关文件被篡改,不允许安装插件安装包。

[0167] 在该实施例中,根据第一数值和第二数值是否相同,来判断插件的所有相关文件是否被篡改,从而确定插件安装包是否安全,在安全时,允许安装插件安装包,在不安全时,不允许安装插件安装包,从而保证用户的安装安全,提升用户的使用体验。

[0168] 例如,在上传插件安装包时,先对插件的所有相关文件进行哈希计算,得到哈希值 A,并使用加密密钥,如私钥对哈希值 A 进行加密,然后将加密后的哈希值 A 写入签名文件,将签名文件和插件的所有相关文件进行打包,得到插件安装包上传至预设服务器。当有应用想安装该插件时,会从预设服务器中获取到插件安装包以及与签名文件中加密后的哈希值 A 对应的解密密钥,根据解密密钥,对加密后的哈希值 A 进行解密,得到哈希值 A,再重新对插件安装包中的插件的所有相关文件进行哈希计算,如果计算结果也为哈希值 A,则说明插件安装包没有被篡改,而如果计算结果为哈希值 B,与 A 不同,则说明插件安装包被篡改,为了保证安装安全,可以不允许安装插件安装包。

[0169] 如图 8 所示,在一个实施例中,在确定插件的所有相关文件未被篡改之后,还包括步骤 S801-S804:

[0170] 在步骤 S801 中,从预设服务器中获取插件安装包的属性信息,其中,属性信息包括下述信息中的一项或多项:插件安装包的名称、插件类型、插件所支持的最低应用版本;

[0171] 在步骤 S802 中,判断欲安装的插件安装包的应用的当前版本号;

[0172] 在步骤 S803 中,在当前版本号不低于最低应用版本号时,转向允许安装插件安装包的步骤;

[0173] 在步骤 S804 中,在当前版本低于最低应用版本号时,转向不允许安装插件安装包的步骤。

[0174] 在该实施例中,上传插件安装包的属性信息至预设服务器,如上传插件安装包名称、插件类型、插件所支持的最低应用版本等至预设服务器,这样,在应用程序或终端设备等在欲安装插件安装包时,目标使用用户可以清楚的获知插件的详细信息,同时也方便应用程序或终端设备等根据插件类型、名称、所支持的最低版本等信息识别和正确安装该插件,从而避免因插件不能安装给用户带来的困扰,提升用户的使用体验。

[0175] 其中,可以通过下述两种方式获取解密密钥。

[0176] 方式一:

[0177] 如图 9 所示,在一个实施例中,上述步骤 S601 包括步骤 S901-S902:

[0178] 在步骤 S901 中,在从预设服务器获取到插件安装包后,发送插件安装包对应的解密密钥的获取请求至预设服务器;

[0179] 在步骤 S902 中,接收预设服务器发送的与插件安装包对应的解密密钥。

[0180] 应用程序或终端设备等在欲安装插件安装包时,可以向预设服务器发送解密密钥获取请求,从而从服务器侧获取解密密钥,对加密后的签名文件进行解密,验证插件安装包是否被伪造或被篡改。

[0181] 解密密钥由服务器保存,由于服务器的权威性和安全性,可以充分保证解密密钥

的安全性。

[0182] 在服务器侧保存的解密密钥,需要加密后保存,避免明文保存,在服务器根据解密密钥获取请求,返回解密密钥时,同样需要将加密之后的解密密钥返回给服务器,避免明文传输,关于解密密钥如何解密,可以预先预定对应的密码。

[0183] 方式二:

[0184] 如图 10 所示,在一个实施例中,上述步骤 S601 包括步骤 S1001:

[0185] 在步骤 S1001 中,在从预设服务器获取到插件安装包后,对插件安装包进行解析获取其中包含的解密密钥。

[0186] 在该实施例中,在上传时,将解密密钥和签名后的所有相关文件一起打包生成插件安装包,这样,在应用程序或终端设备等在欲安装插件安装包时,可以直接从插件安装包中解析出解密密钥,进而根据解密密钥对加密后的签名文件进行解密,验证插件安装包是否被伪造或被篡改。

[0187] 下述为本公开装置实施例,可以用于执行本公开方法实施例。

[0188] 图 11 是根据一示例性实施例示出的一种插件安装包上传装置的框图,该装置可以通过软件、硬件或者两者的结合实现成为电子设备的部分或者全部。如图 11 所示,该插件安装包上传装置包括:

[0189] 第一接收模块 1101 被配置为接收插件的所有相关文件;

[0190] 计算模块 1102 被配置为将所有相关文件使用预设算法进行计算,得到第一数值;在一个实施例中,预设算法包括哈希算法。

[0191] 加密模块 1103 被配置为使用加密密钥对第一数值进行加密,将加密后的第一数值写入签名文件;

[0192] 第一生成模块 1104 被配置为将签名文件和所述所有相关文件进行打包,生成插件安装包;

[0193] 第一上传模块 1105 被配置为上传插件安装包至预设服务器。

[0194] 如图 12 所示,在一个实施例中,上述装置还包括:

[0195] 第二接收模块 1201 被配置为接收输入的密钥生成命令;

[0196] 第二生成模块 1202 被配置为根据密钥生成命令,生成加密密钥和与加密密钥对应的解密密钥。

[0197] 如图 13 所示,在一个实施例中,上述装置还包括:

[0198] 第二上传模块 1301 被配置为上传解密密钥至预设服务器,预设服务器用于将插件安装包和解密密钥进行对应存储。

[0199] 如图 14 所示,在一个实施例中,上述第一生成模块 1104 包括:

[0200] 安装包生成子模块 1401 被配置为将解密密钥、签名文件、所有相关文件进行打包生成插件安装包。

[0201] 如图 15 所示,在一个实施例中,上述装置还包括:

[0202] 第三上传模块 1501 被配置为上传插件安装包的属性信息至预设服务器,其中,属性信息中包括下述一项或多项:

[0203] 插件安装包的名称、插件类型、插件所支持的最低应用版本。

[0204] 图 16 是根据一示例性实施例示出的一种插件安装包安装装置的框图,该装置可

以通过软件、硬件或者两者的结合实现成为电子设备的部分或者全部。如图 16 所示,该插件安装包安装装置包括:

[0205] 密钥获取模块 1601 被配置为在从预设服务器获取到插件安装包后,获取插件安装包对应的解密密钥,其中,插件安装包中包括插件的签名文件和所有相关文件,解密密钥与签名文件的加密密钥相对应;

[0206] 加密模块 1602 被配置为使用解密密钥对加密后的第一数值进行解密,得到第一数值;

[0207] 计算模块 1603 被配置为将所有相关文件使用预设算法进行计算,得到第二数值;在一个实施例中,预设算法包括哈希算法。

[0208] 比较模块 1604 被配置为将第一数值和第二数值进行比较,得到比较结果;

[0209] 确定模块 1605 被配置为根据比较结果确定是否允许安装插件安装包。

[0210] 如图 17 所示,在一个实施例中,上述确定模块 1605 包括:

[0211] 第一确定子模块 1701 被配置为在第一数值与第二数值相同时,确定插件的所有相关文件未被篡改,允许安装插件安装包;

[0212] 第二确定子模块 1702 被配置为在第一数值与第二数值不同时,确定插件的所有相关文件被篡改,不允许安装插件安装包。

[0213] 如图 18 所示,在一个实施例中,上述确定模块 1605 还包括:

[0214] 信息获取子模块 1801 被配置为在确定插件的所有相关文件未被篡改后,从预设服务器中获取插件安装包的属性信息,其中,属性信息包括下述信息中的一项或多项:插件安装包的名称、插件类型、插件所支持的最低应用版本;

[0215] 判断子模块 1802 被配置为判断欲安装的插件安装包的应用的当前版本号;

[0216] 第一确定子模块 1701 用于,在当前版本号不低于最低应用版本号时,转向允许安装插件安装包的步骤;

[0217] 第二确定子模块 1702 用于,在当前版本低于最低应用版本号时,转向不允许安装插件安装包的步骤。

[0218] 如图 19 所示,在一个实施例中,上述密钥获取模块 1601 包括:

[0219] 发送子模块 1901 被配置为在从预设服务器获取到插件安装包后,发送插件安装包对应的解密密钥的获取请求至预设服务器;

[0220] 接收子模块 1902 被配置为接收预设服务器发送的与插件安装包对应的解密密钥;或者

[0221] 解析子模块 1903 被配置为在从预设服务器获取到插件安装包后,对插件安装包进行解析获取其中包含的解密密钥。

[0222] 根据本公开实施例的第五方面,提供一种插件安装包上传装置,包括:

[0223] 处理器;

[0224] 用于存储处理器可执行指令的存储器;

[0225] 其中,处理器被配置为:

[0226] 接收插件的所有相关文件;

[0227] 将所述所有相关文件使用预设算法进行计算,得到第一数值;

[0228] 使用加密密钥对所述第一数值进行加密,将所述加密后的第一数值写入签名文

件；

[0229] 将签名文件和所述所有相关文件进行打包,生成插件安装包；

[0230] 上传所述插件安装包至预设服务器。

[0231] 上述处理器还可被配置为：

[0232] 所述预设算法包括哈希算法。

[0233] 上述处理器还可被配置为：

[0234] 所述方法还包括：

[0235] 接收输入的密钥生成命令；

[0236] 根据所述密钥生成命令,生成所述加密密钥和与所述加密密钥对应的解密密钥。

[0237] 上述处理器还可被配置为：

[0238] 所述方法还包括：

[0239] 上传所述解密密钥至所述预设服务器,所述预设服务器用于将所述插件安装包和所述解密密钥进行对应存储；或者

[0240] 生成插件安装包,包括：

[0241] 将所述解密密钥、签名文件、所述所有相关文件进行打包生成插件安装包。

[0242] 上述处理器还可被配置为：

[0243] 所述方法还包括：

[0244] 上传所述插件安装包的属性信息至所述预设服务器,其中,所述属性信息中包括下述一项或多项：

[0245] 插件安装包的名称、插件类型、插件所支持的最低应用版本。

[0246] 根据本公开实施例的第六方面,提供一种插件安装包安装装置,包括：

[0247] 处理器；

[0248] 用于存储处理器可执行指令的存储器；

[0249] 其中,处理器被配置为：

[0250] 在从预设服务器获取到插件安装包后,获取所述插件安装包对应的解密密钥,其中,所述插件安装包中包括插件的签名文件和所有相关文件,所述解密密钥与所述签名文件的加密密钥相对应；

[0251] 使用所述解密密钥对所述加密后的第一数值进行解密,得到第一数值；

[0252] 将所述所有相关文件使用预设算法进行计算,得到第二数值；

[0253] 将所述第一数值和所述第二数值进行比较,得到比较结果；

[0254] 根据比较结果确定是否允许安装所述插件安装包。

[0255] 上述处理器还可被配置为：

[0256] 所述预设算法包括哈希算法。

[0257] 上述处理器还可被配置为：

[0258] 所述根据比较结果确定是否允许安装所述插件安装包,包括：

[0259] 在所述第一数值与所述第二数值相同时,确定所述插件的所有相关文件未被篡改,允许安装所述插件安装包；

[0260] 在所述第一数值与所述第二数值不同时,确定所述插件的所有相关文件被篡改,不允许安装所述插件安装包。

[0261] 上述处理器还可被配置为：

[0262] 确定所述插件的所有相关文件未被篡改之后，还包括：

[0263] 从所述预设服务器中获取所述插件安装包的属性信息，其中，所述属性信息包括下述信息中的一项或多项：插件安装包的名称、插件类型、插件支持的最低应用版本号；

[0264] 判断欲安装的所述插件安装包的应用的当前版本号；

[0265] 在所述当前版本号不低于所述最低应用版本号时，转向允许安装所述插件安装包的步骤；

[0266] 在所述当前版本低于所述最低应用版本号时，转向不允许安装所述插件安装包的步骤。

[0267] 上述处理器还可被配置为：

[0268] 所述在从预设服务器获取到插件安装包后，获取所述插件安装包对应的解密密钥，包括：

[0269] 在从所述预设服务器获取到插件安装包后，发送所述插件安装包对应的解密密钥的获取请求至所述预设服务器；

[0270] 接收所述预设服务器发送的与所述插件安装包对应的解密密钥；或者

[0271] 在从所述预设服务器获取到插件安装包后，对所述插件安装包进行解析获取其中包含的解密密钥。

[0272] 关于上述实施例中的装置，其中各个模块执行操作的具体方式已经在有关该方法的实施例中进行了详细描述，此处将不做详细阐述说明。

[0273] 图 20 是根据一示例性实施例示出的一种用于插件安装包上传装置或插件安装包安装的框图，该装置适用于终端设备。例如，装置 2000 可以是移动电话，计算机，数字广播终端，消息收发设备，游戏控制台，平板设备，医疗设备，健身设备，个人数字助理等。

[0274] 装置 2000 可以包括以下一个或多个组件：处理组件 2002，存储器 2004，电源组件 2006，多媒体组件 2008，音频组件 2010，输入/输出 (I/O) 的接口 2012，传感器组件 2014，以及通信组件 2016。

[0275] 处理组件 2002 通常控制装置 2000 的整体操作，诸如与显示，电话呼叫，数据通信，相机操作和记录操作相关联的操作。处理元件 2002 可以包括一个或多个处理器 2012 来执行指令，以完成上述的方法的全部或部分步骤。此外，处理组件 2002 可以包括一个或多个模块，便于处理组件 2002 和其他组件之间的交互。例如，处理部件 2002 可以包括多媒体模块，以方便多媒体组件 2008 和处理组件 2002 之间的交互。

[0276] 存储器 2004 被配置为存储各种类型的数据以支持在设备 2000 的操作。这些数据的示例包括用于在装置 2000 上操作的任何应用程序或方法的指令，联系人数据，电话簿数据，消息，图片，视频等。存储器 2004 可以由任何类型的易失性或非易失性存储设备或者它们的组合实现，如静态随机存取存储器 (SRAM)，电可擦除可编程只读存储器 (EEPROM)，可擦除可编程只读存储器 (EPROM)，可编程只读存储器 (PROM)，只读存储器 (ROM)，磁存储器，快闪存储器，磁盘或光盘。

[0277] 电力组件 2006 为装置 2000 的各种组件提供电力。电力组件 2006 可以包括电源管理系统，一个或多个电源，及其他与为装置 2000 生成、管理和分配电力相关联的组件。

[0278] 多媒体组件 2008 包括在所述装置 2000 和用户之间的提供一个输出接口的屏幕。

在一些实施例中,屏幕可以包括液晶显示器(LCD)和触摸面板(TP)。如果屏幕包括触摸面板,屏幕可以被实现为触摸屏,以接收来自用户的输入信号。触摸面板包括一个或多个触摸传感器以感测触摸、滑动和触摸面板上的手势。所述触摸传感器可以不仅感测触摸或滑动动作的边界,而且还检测与所述触摸或滑动操作相关的持续时间和压力。在一些实施例中,多媒体组件 2008 包括一个前置摄像头和 / 或后置摄像头。当设备 2000 处于操作模式,如拍摄模式或视频模式时,前置摄像头和 / 或后置摄像头可以接收外部的多媒体数据。每个前置摄像头和后置摄像头可以是一个固定的光学透镜系统或具有焦距和光学变焦能力。

[0279] 音频组件 2010 被配置为输出和 / 或输入音频信号。例如,音频组件 2010 包括一个麦克风(MIC),当装置 2000 处于操作模式,如呼叫模式、记录模式和语音识别模式时,麦克风被配置为接收外部音频信号。所接收的音频信号可以被进一步存储在存储器 2004 或经由通信组件 2016 发送。在一些实施例中,音频组件 2010 还包括一个扬声器,用于输出音频信号。

[0280] I/O 接口 2012 为处理组件 2002 和外围接口模块之间提供接口,上述外围接口模块可以是键盘,点击轮,按钮等。这些按钮可包括但不限于:主页按钮、音量按钮、启动按钮和锁定按钮。

[0281] 传感器组件 2014 包括一个或多个传感器,用于为装置 2000 提供各个方面的状态评估。例如,传感器组件 2014 可以检测到设备 2000 的打开 / 关闭状态,组件的相对定位,例如所述组件为装置 2000 的显示器和小键盘,传感器组件 2014 还可以检测装置 2000 或装置 2000 一个组件的位置改变,用户与装置 2000 接触的存在或不存在,装置 2000 方位或加速 / 减速和装置 2000 的温度变化。传感器组件 2014 可以包括接近传感器,被配置用来在没有任何的物理接触时检测附近物体的存在。传感器组件 2014 还可以包括光传感器,如 CMOS 或 CCD 图像传感器,用于在成像应用中使用。在一些实施例中,该传感器组件 2014 还可以包括加速度传感器,陀螺仪传感器,磁传感器,压力传感器或温度传感器。

[0282] 通信组件 2016 被配置为便于装置 2000 和其他设备之间有线或无线方式的通信。装置 2000 可以接入基于通信标准的无线网络,如 WiFi, 2G 或 3G, 或它们的组合。在一个示例性实施例中,通信部件 2016 经由广播信道接收来自外部广播管理系统的广播信号或广播相关信息。在一个示例性实施例中,所述通信部件 2016 还包括近场通信(NFC)模块,以促进短程通信。例如,在 NFC 模块可基于射频识别(RFID)技术,红外数据协会(IrDA)技术,超宽带(UWB)技术,蓝牙(BT)技术和其他技术来实现。

[0283] 在示例性实施例中,装置 2000 可以被一个或多个应用专用集成电路(ASIC)、数字信号处理器(DSP)、数字信号处理设备(DSPD)、可编程逻辑器件(PLD)、现场可编程门阵列(FPGA)、控制器、微控制器、微处理器或其他电子元件实现,用于执行上述方法。

[0284] 在示例性实施例中,还提供了一种包括指令的非临时性计算机可读存储介质,例如包括指令的存储器 2004,上述指令可由装置 2000 的处理器 2020 执行以完成上述方法。例如,所述非临时性计算机可读存储介质可以是 ROM、随机存取存储器(RAM)、CD-ROM、磁带、软盘和光数据存储设备等。

[0285] 一种非临时性计算机可读存储介质,当所述存储介质中的指令由装置 2000 的处理器执行时,使得装置 2000 能够执行上述插件安装包上传的方法,所述方法包括:

[0286] 接收插件的所有相关文件;

- [0287] 将所述所有相关文件使用预设算法进行计算,得到第一数值;
- [0288] 使用加密密钥对所述第一数值进行加密,将所述加密后的第一数值写入签名文件;
- [0289] 将签名文件和所述所有相关文件进行打包,生成插件安装包;
- [0290] 上传所述插件安装包至预设服务器。
- [0291] 在一个实施例中,所述预设算法包括哈希算法。
- [0292] 在一个实施例中,所述方法还包括:
- [0293] 接收输入的密钥生成命令;
- [0294] 根据所述密钥生成命令,生成所述加密密钥和与所述加密密钥对应的解密密钥。
- [0295] 在一个实施例中,所述方法还包括:
- [0296] 上传所述解密密钥至所述预设服务器,所述预设服务器用于将所述插件安装包和所述解密密钥进行对应存储;或者
- [0297] 生成插件安装包,包括:
- [0298] 将所述解密密钥、签名文件、所述所有相关文件进行打包生成插件安装包。
- [0299] 在一个实施例中,所述方法还包括:
- [0300] 上传所述插件安装包的属性信息至所述预设服务器,其中,所述属性信息中包括下述一项或多项:
- [0301] 插件安装包的名称、插件类型、插件所支持的最低应用版本。
- [0302] 一种非临时性计算机可读存储介质,当所述存储介质中的指令由装置 2000 的处理器执行时,使得装置 2000 能够执行上述插件安装包安装方法,所述方法包括:
- [0303] 在从预设服务器获取到插件安装包后,获取所述插件安装包对应的解密密钥,其中,所述插件安装包中包括插件的签名文件和所有相关文件,所述解密密钥与所述签名文件的加密密钥相对应;
- [0304] 使用所述解密密钥对所述加密后的第一数值进行解密,得到第一数值;
- [0305] 将所述所有相关文件使用预设算法进行计算,得到第二数值;
- [0306] 将所述第一数值和所述第二数值进行比较,得到比较结果;
- [0307] 根据比较结果确定是否允许安装所述插件安装包。
- [0308] 在一个实施例中,所述预设算法包括哈希算法。
- [0309] 在一个实施例中,所述根据比较结果确定是否允许安装所述插件安装包,包括:
- [0310] 在所述第一数值与所述第二数值相同时,确定所述插件的所有相关文件未被篡改,允许安装所述插件安装包;
- [0311] 在所述第一数值与所述第二数值不同时,确定所述插件的所有相关文件被篡改,不允许安装所述插件安装包。
- [0312] 在一个实施例中,确定所述插件的所有相关文件未被篡改之后,还包括:
- [0313] 从所述预设服务器中获取所述插件安装包的属性信息,其中,所述属性信息包括下述信息中的一项或多项:插件安装包的名称、插件类型、插件支持的最低应用版本号;
- [0314] 判断欲安装的所述插件安装包的应用的当前版本号;
- [0315] 在所述当前版本号不低于所述最低应用版本号时,转向允许安装所述插件安装包的步骤;

[0316] 在所述当前版本低于所述最低应用版本号时,转向不允许安装所述插件安装包的步骤。

[0317] 在一个实施例中,所述在从预设服务器获取到插件安装包后,获取所述插件安装包对应的解密密钥,包括:

[0318] 在从所述预设服务器获取到插件安装包后,发送所述插件安装包对应的解密密钥的获取请求至所述预设服务器;

[0319] 接收所述预设服务器发送的与所述插件安装包对应的解密密钥;或者

[0320] 在从所述预设服务器获取到插件安装包后,对所述插件安装包进行解析获取其中包含的解密密钥。

[0321] 本领域技术人员在考虑说明书及实践这里公开的公开后,将容易想到本公开的其它实施方案。本申请旨在涵盖本公开的任何变型、用途或者适应性变化,这些变型、用途或者适应性变化遵循本公开的一般性原理并包括本公开未公开的本技术领域中的公知常识或惯用技术手段。说明书和实施例仅被视为示例性的,本公开的真正范围和精神由下面的权利要求指出。

[0322] 应当理解的是,本公开并不局限于上面已经描述并在附图中示出的精确结构,并且可以在不脱离其范围进行各种修改和改变。本公开的范围仅由所附的权利要求来限制。

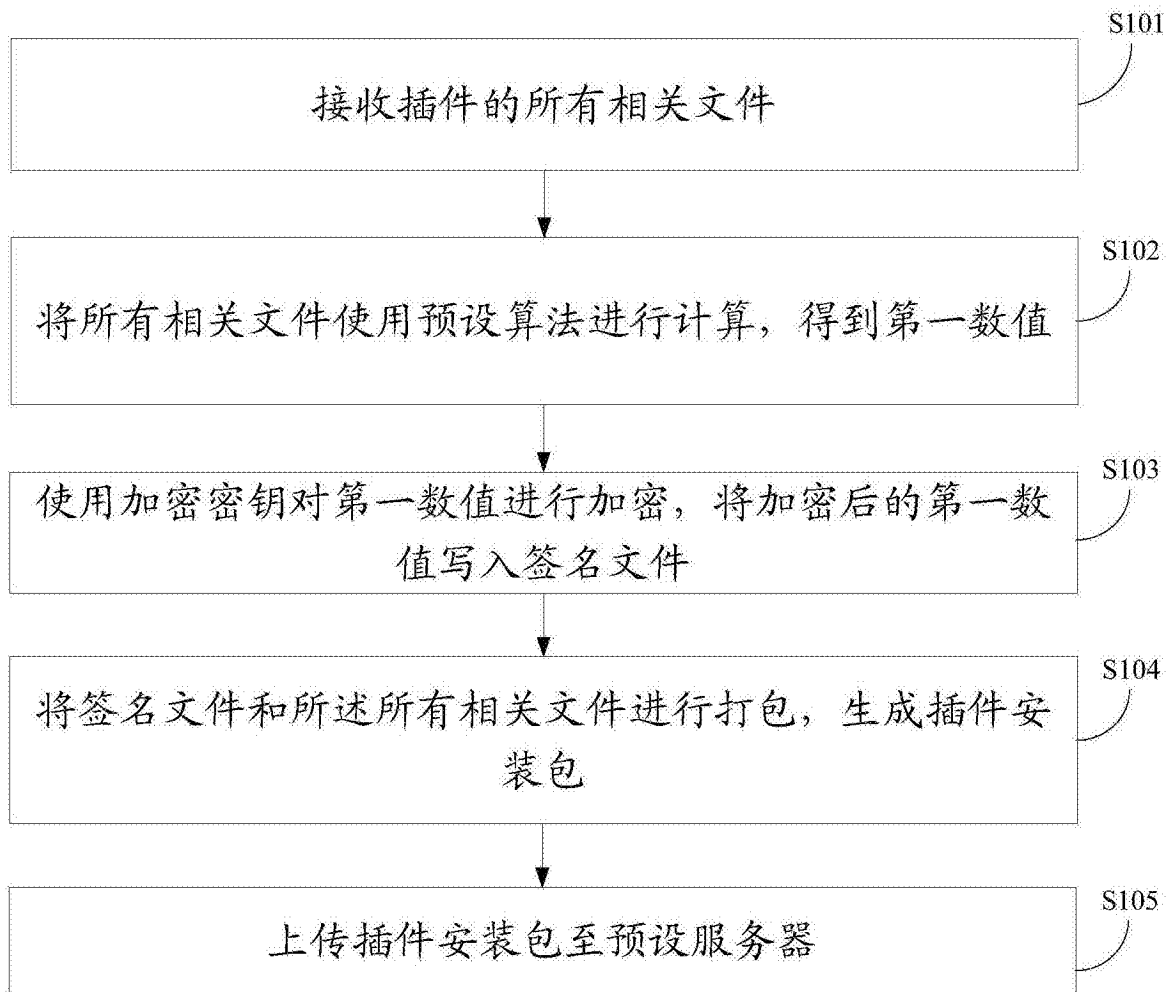


图 1

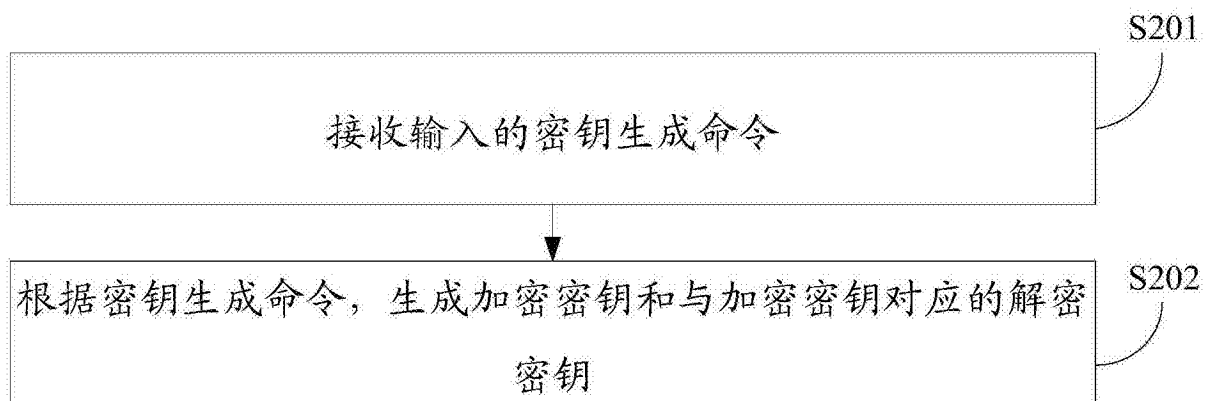


图 2

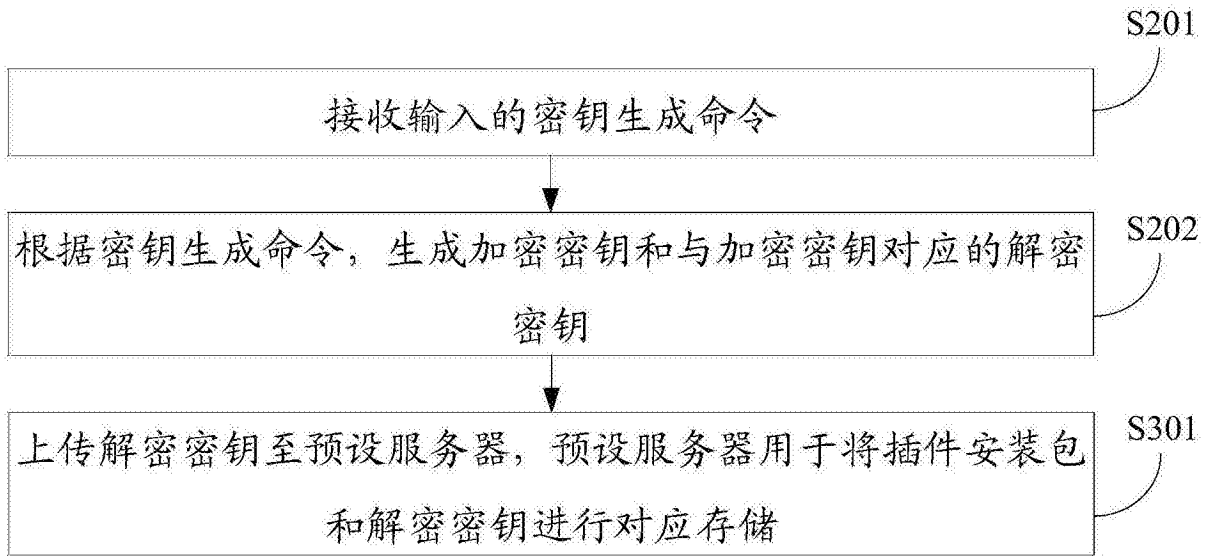


图 3

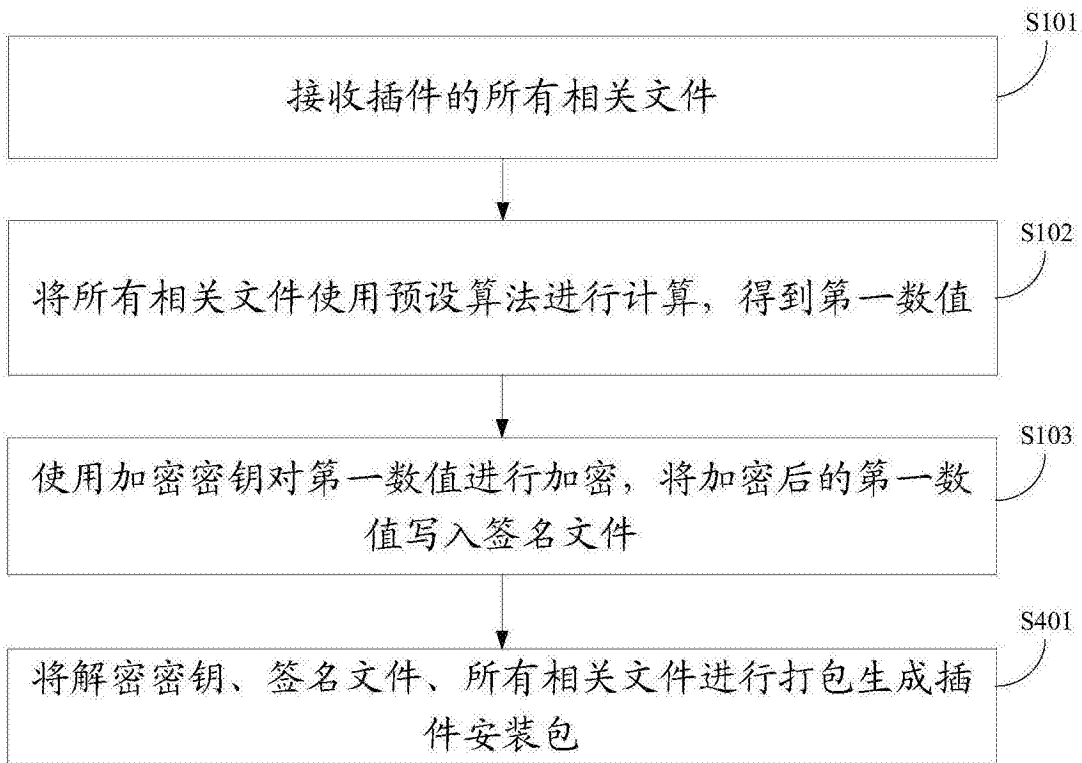


图 4

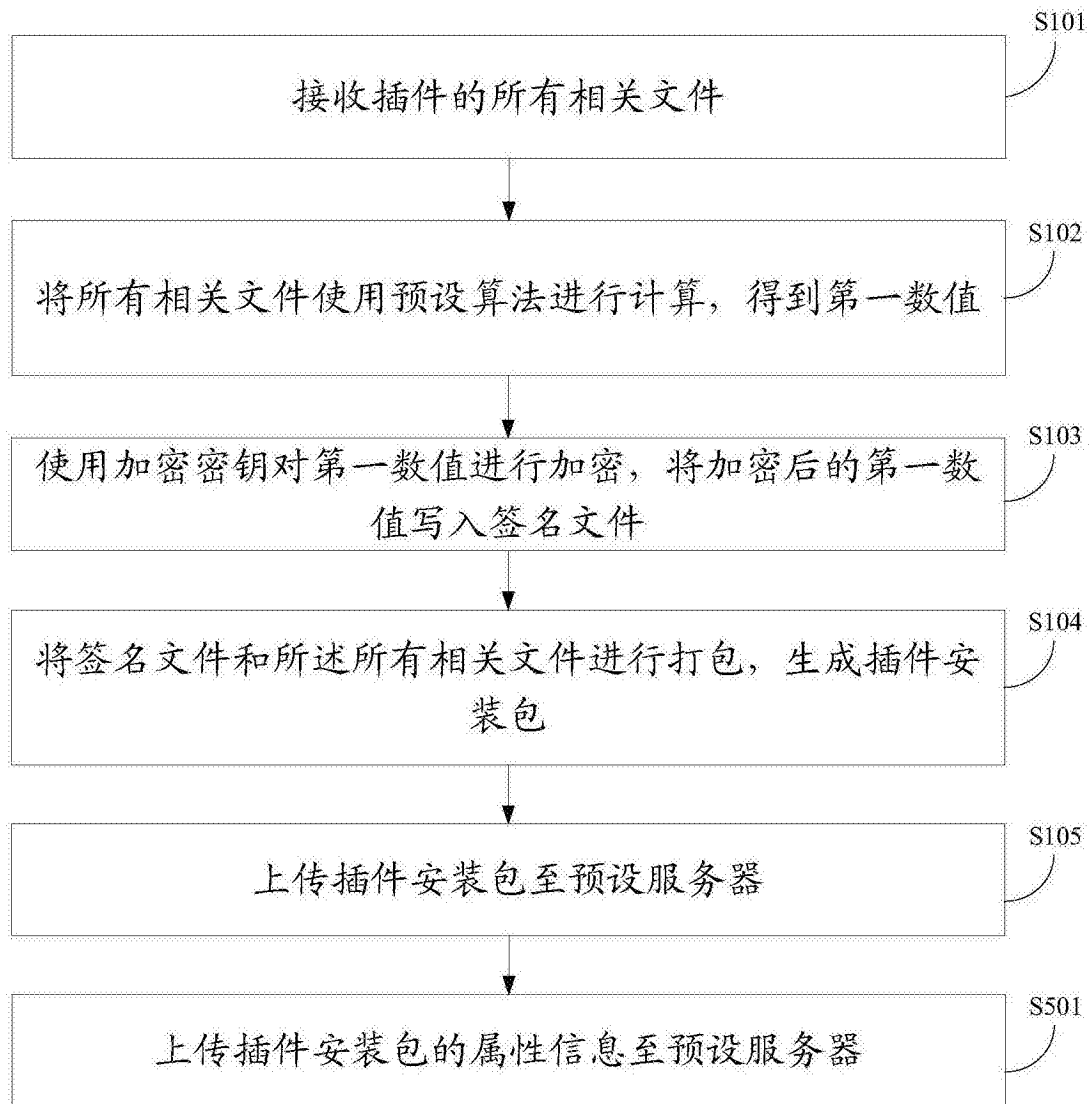


图 5A

插件管理

插件安装包名称 _____

插件类型 _____

最低应用版本 _____

图 5B

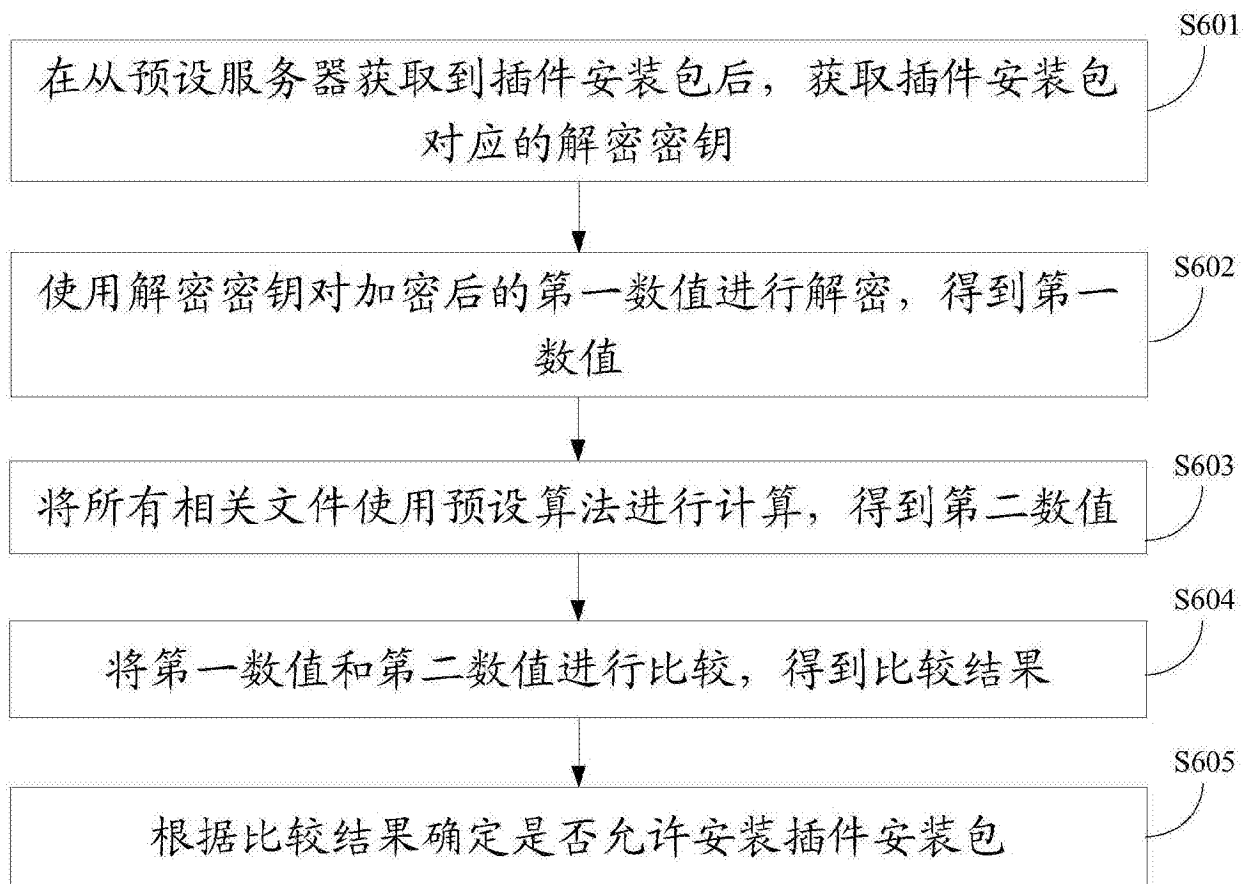


图 6

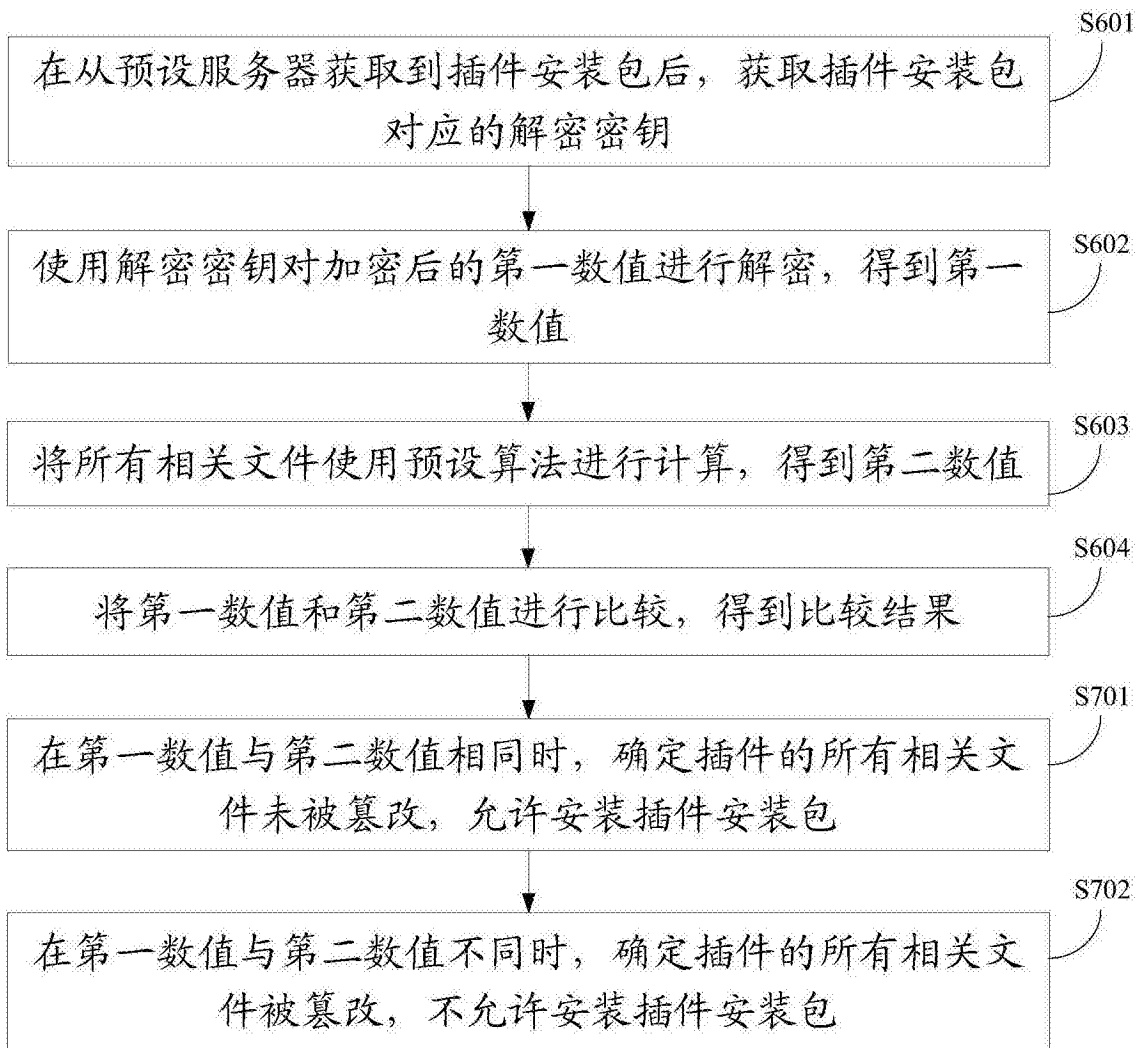


图 7

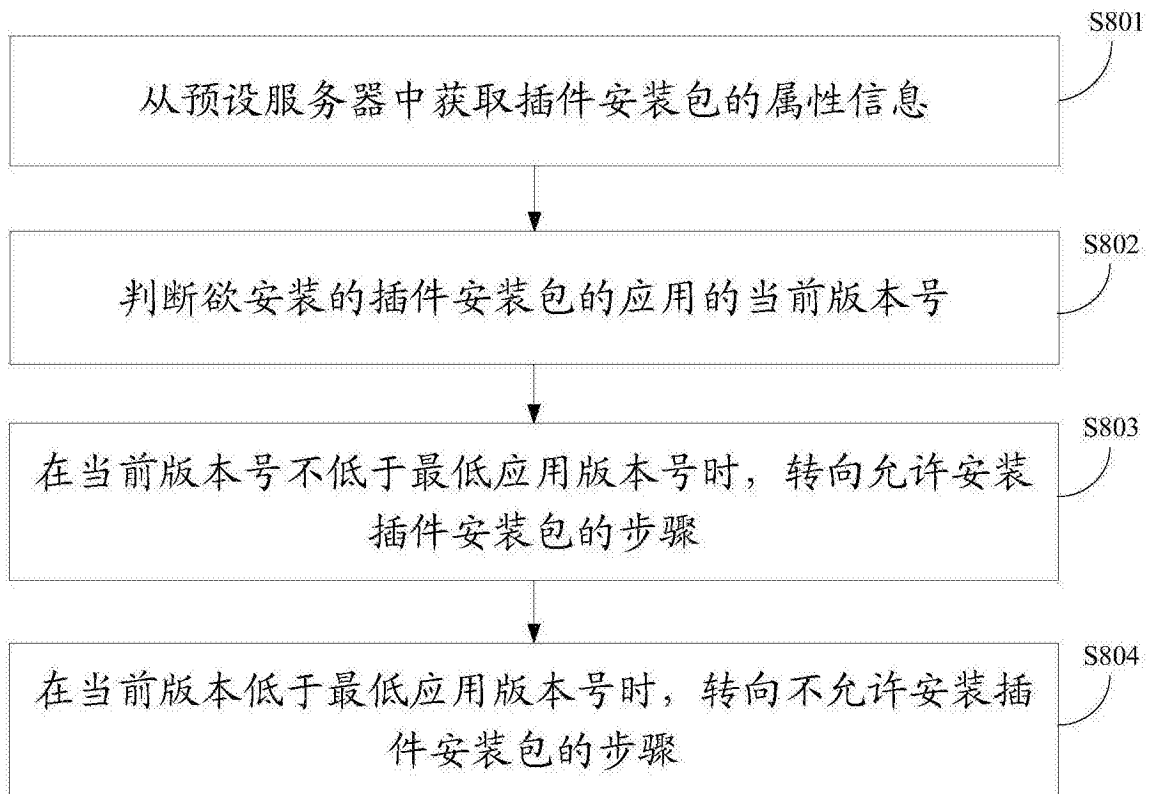


图 8

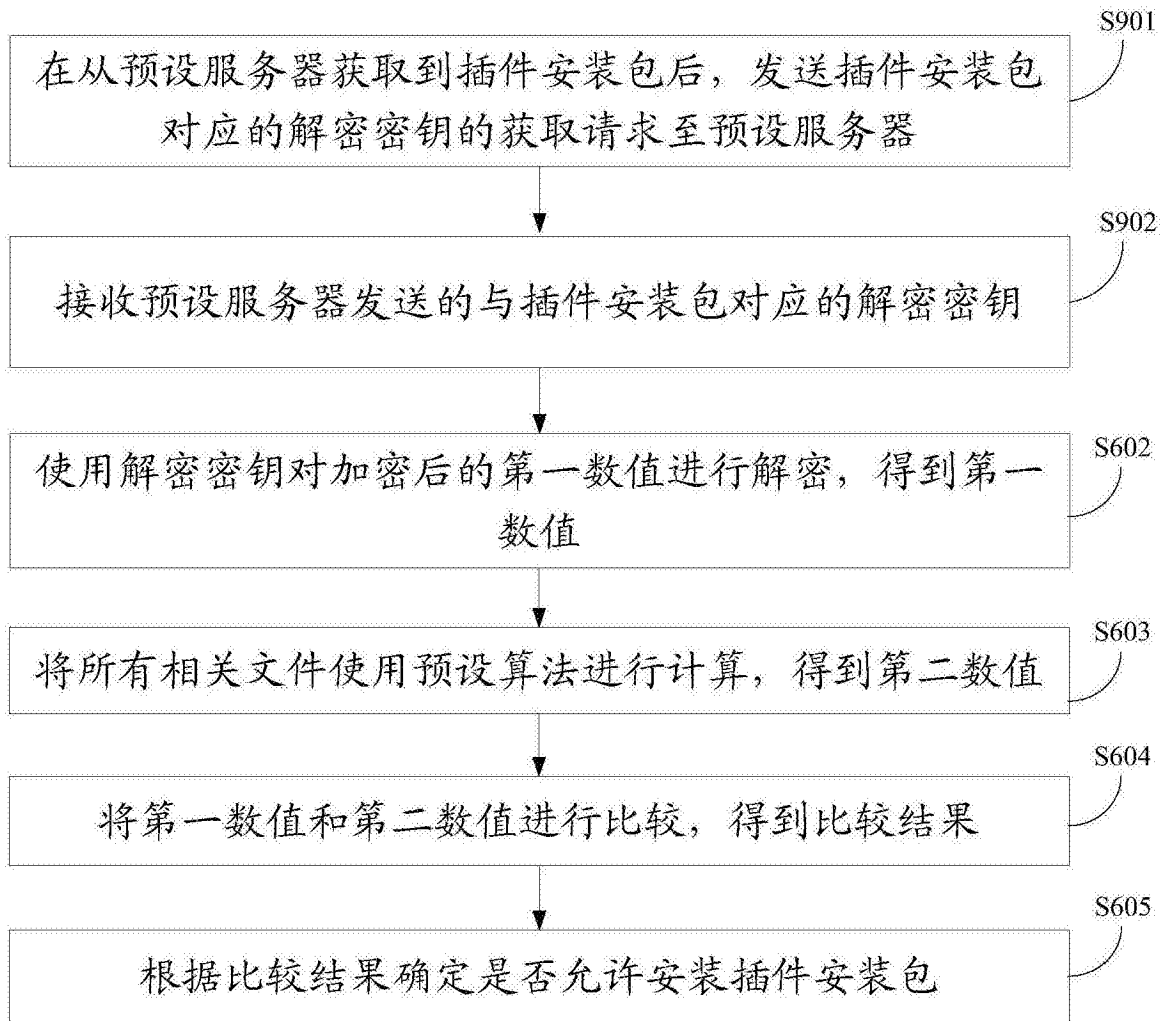


图9

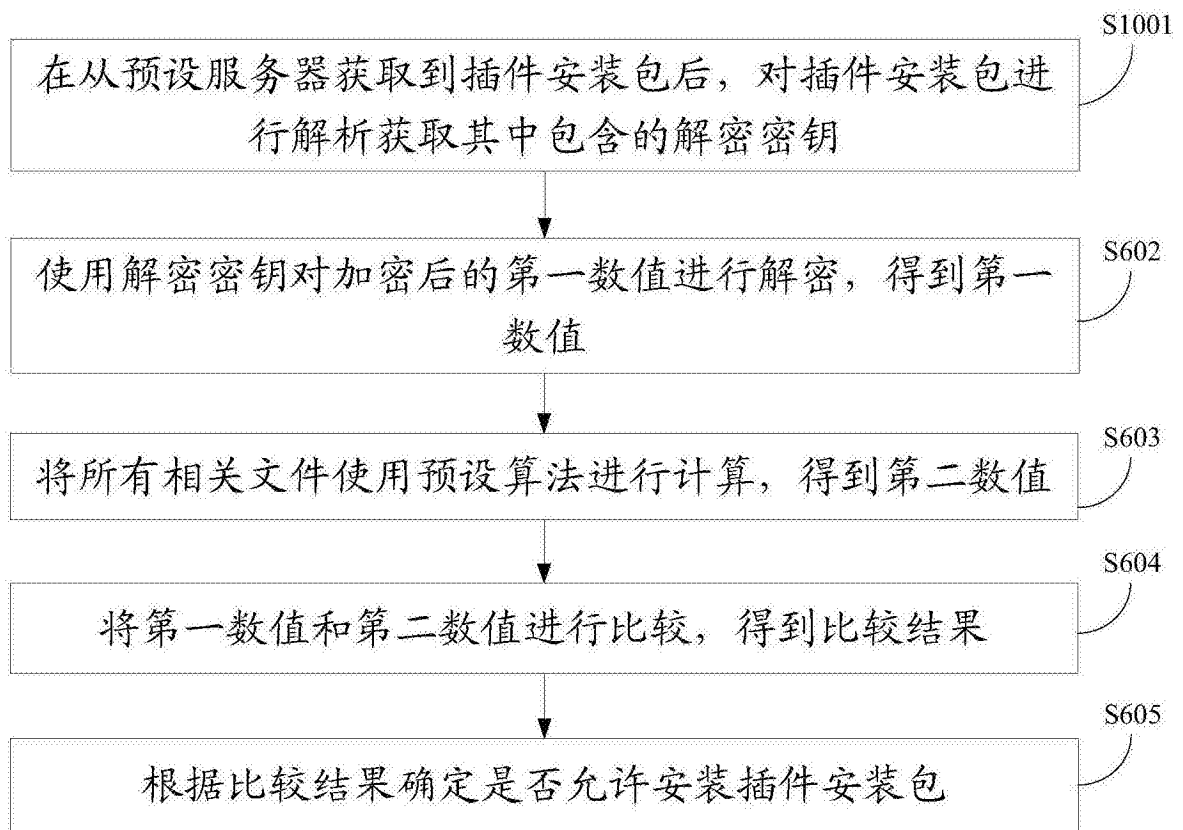


图 10

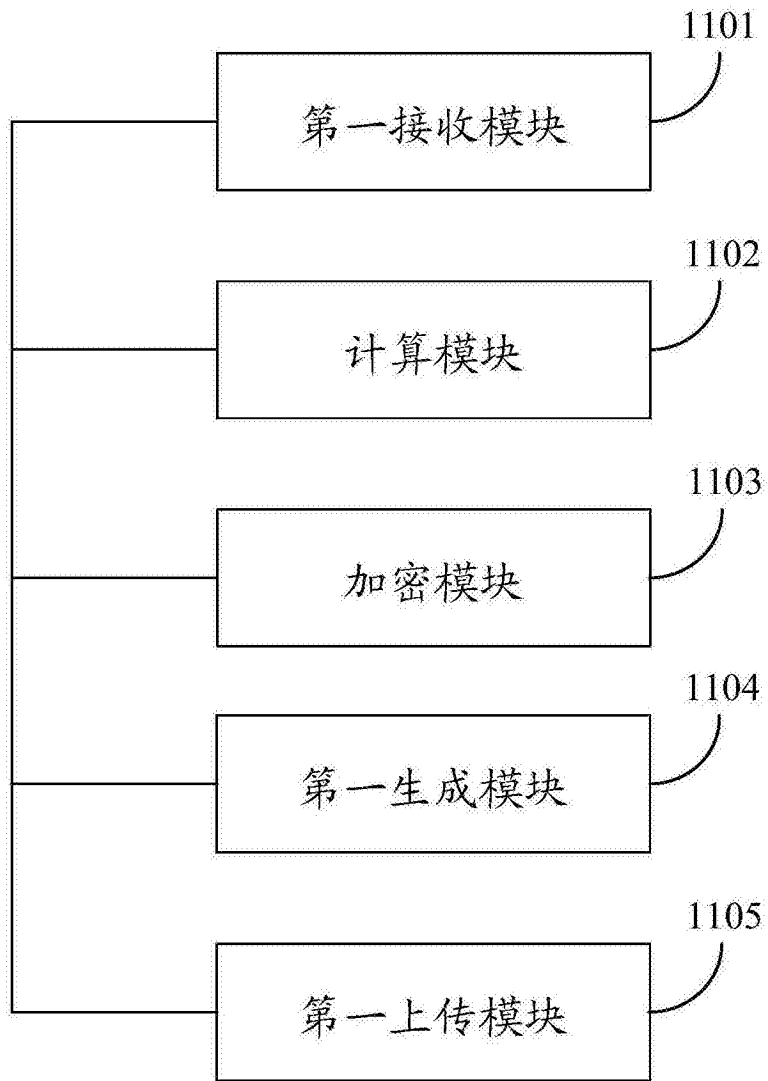


图 11

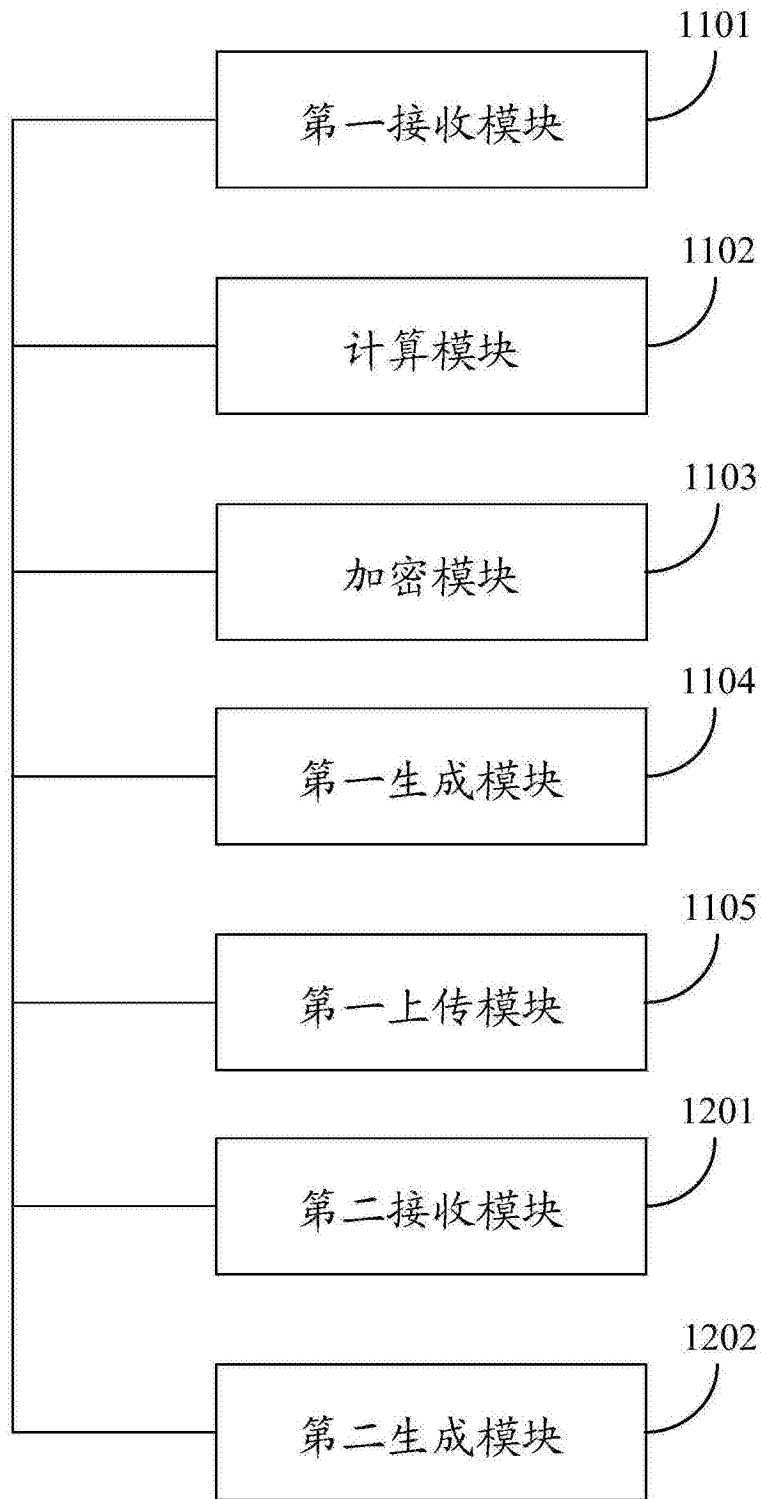


图 12

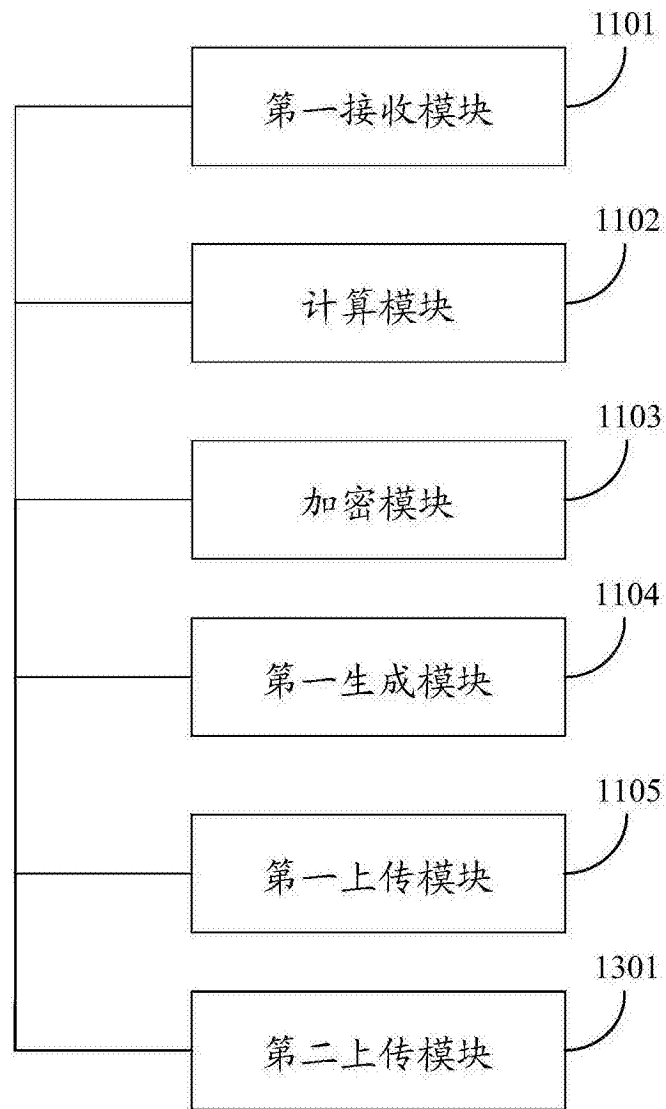


图 13

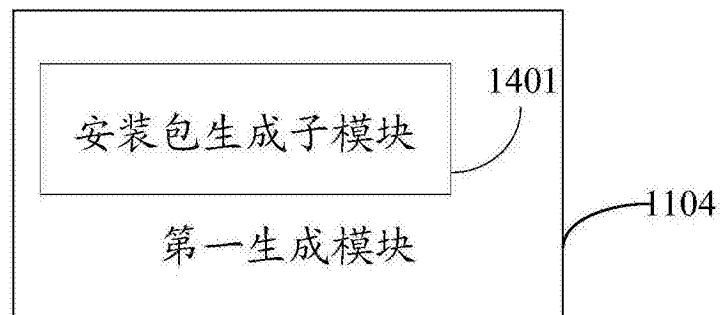


图 14

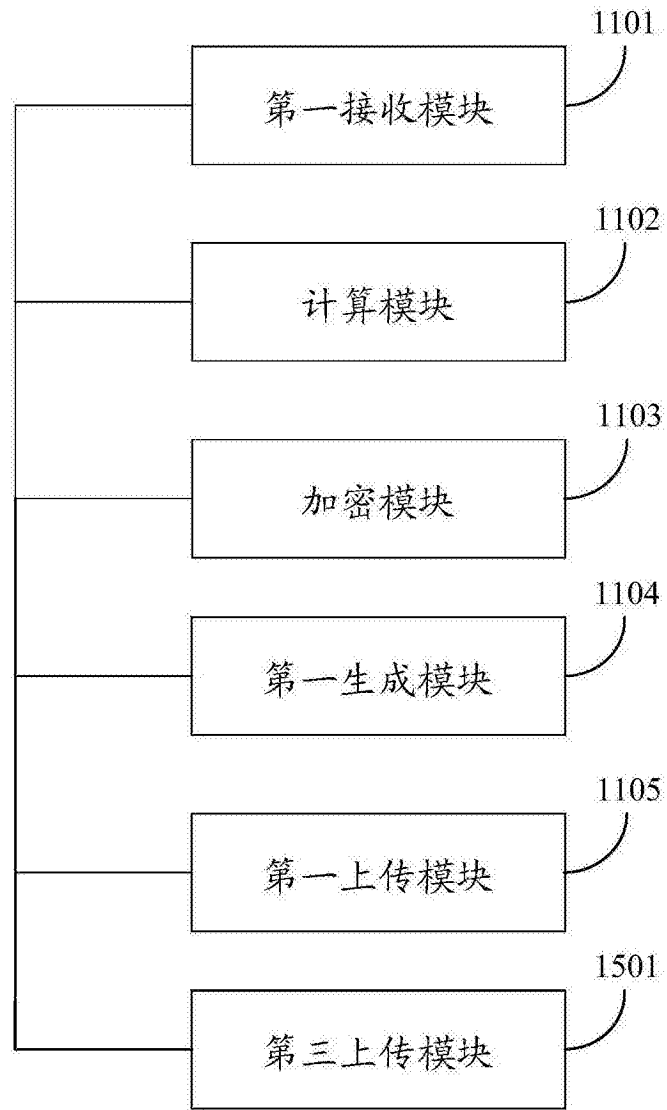


图 15

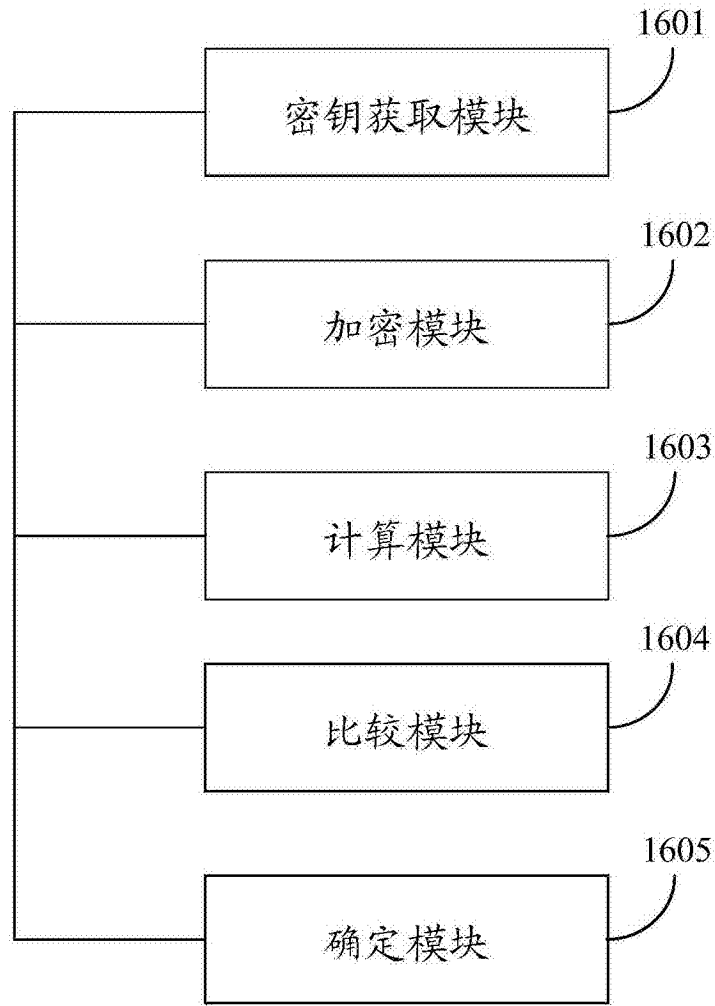


图 16

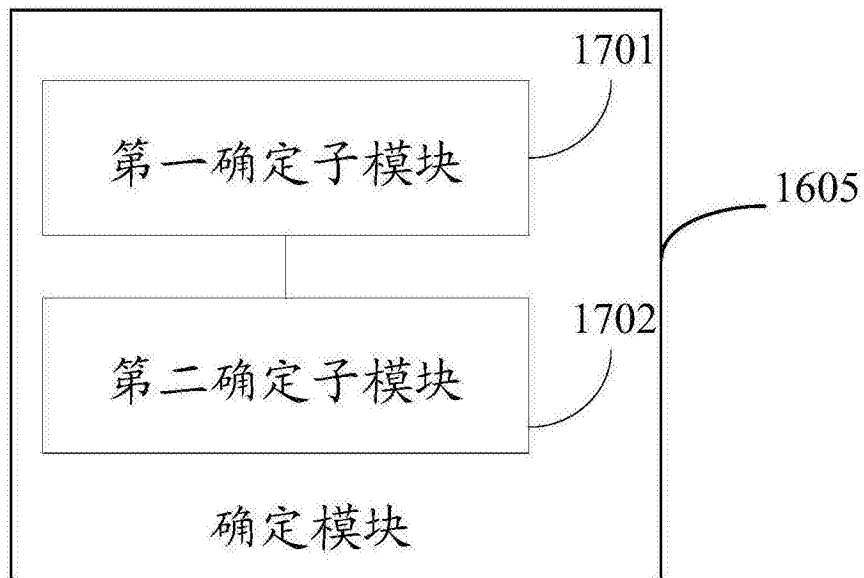


图 17

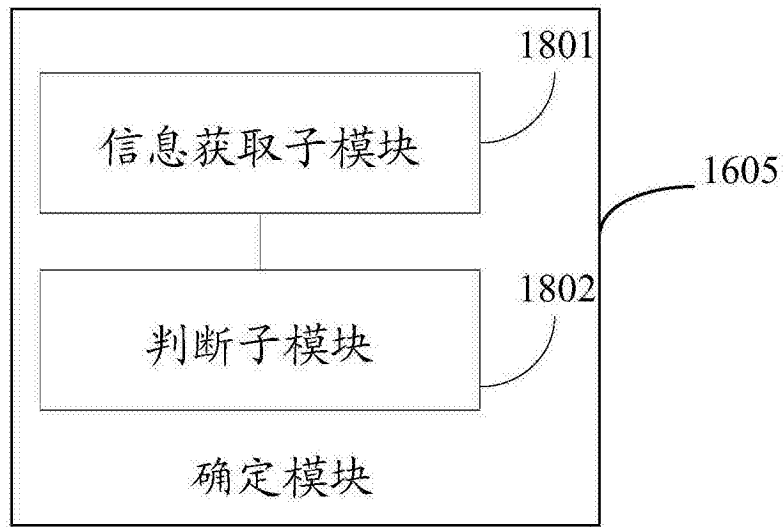


图 18

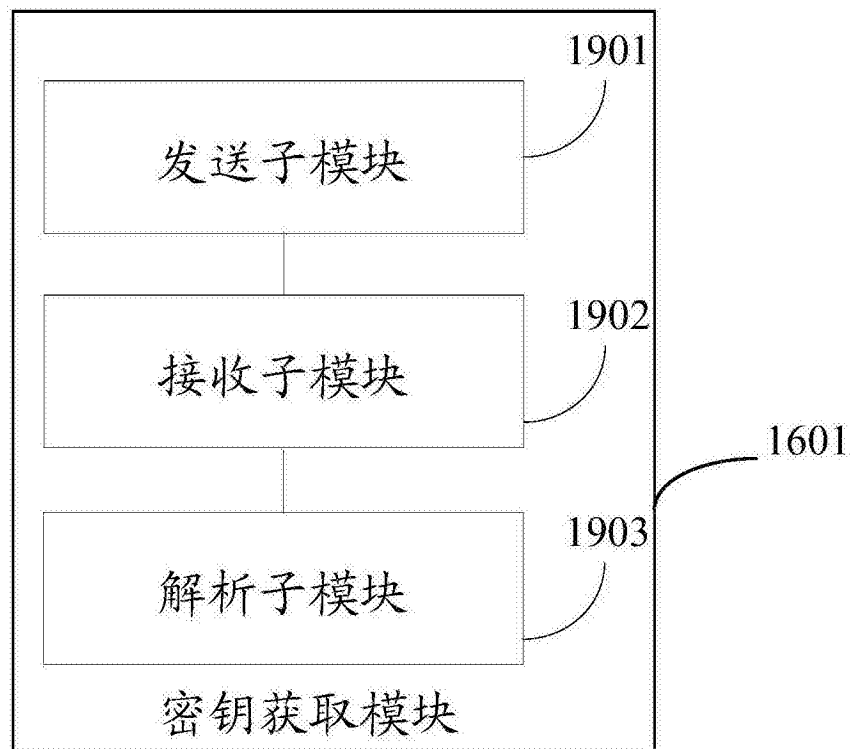


图 19

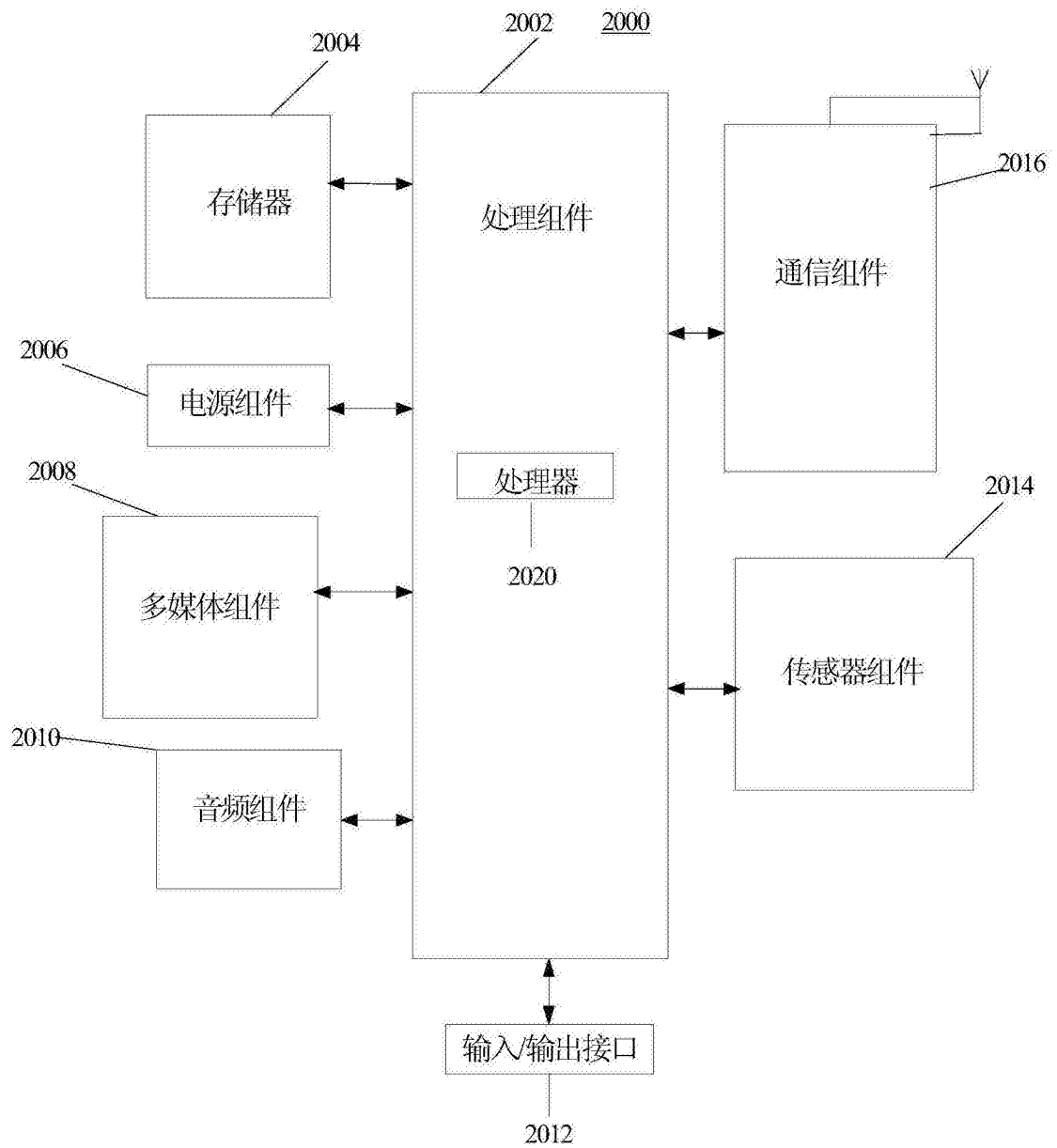


图 20