(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 108140099 B
(45)授权公告日 2019.11.19

(21)申请号 201680059104.7
(22)申请日 2016.09.30
(65)同一申请的已公布的文献号
申请公布号 CN 108140099 A
(43)申请公布日 2018.06.08
(30)优先权数据
62/236,435 2015.10.02 US
(85)PCT国际申请进入国家阶段日
2018.03.29
(86)PCT国际申请的申请数据
PCT/US2016/054895 2016.09.30
(87)PCT国际申请的公布数据
WO2017/059306 EN 2017.04.06
(73)专利权人 谷歌有限责任公司
地址 美国加利福尼亚州
(72)发明人 迈克尔·伯罗斯 希马宾度·普查拉亚·达乌德 杰定·罗黑尔 安库尔·塔利
(74)专利代理机构 上海华诚知识产权代理有限公司 31300
代理人 肖华
(51)Int.Cl.
*G06F 21/64*(2006.01)
(56)对比文件
US 2015/163206 A1,2015.06.11,
US 2004/088646 A1,2004.05.06,
CN 1897518 A,2007.01.17,
CN 101964789 A,2011.02.02,
审查员 冯慧萍

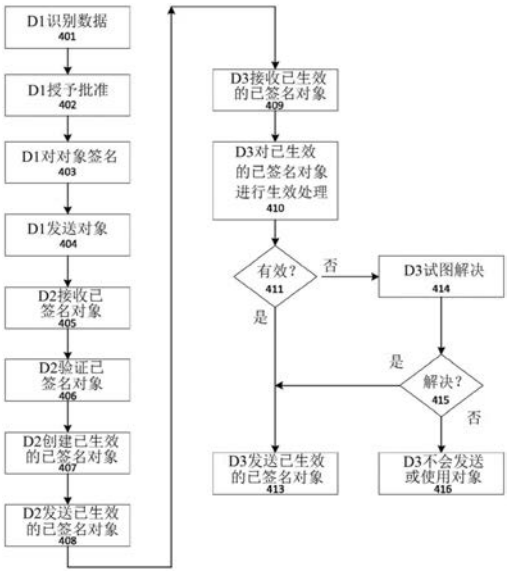
权利要求书4页 说明书14页 附图5页

(54)发明名称

在二进制数据同步协议中交换的更新的签名

(57)摘要

在分布式系统中,数据被在三个以上的电子设备之间共享。第一设备生成包括数据的对象并对包括数据的对象签名。第二设备接收已签名对象并确定已签名对象是否有效。如果有效,第二设备将生成已生效的已签名对象并将其发送到第三设备。第三设备将通过确定对象是否包括第一和第二设备两者的有效签名来使对象生效。




CN 108140099 B

1.一种在分布式系统中的电子设备之间共享对象的方法,其特征在于,所述方法包括:

由与同步群组相关的第一电子设备:

生成对象,

对所述对象签名以创建已签名对象,以及

将所述已签名对象传送到与同步群组相关的第二电子设备;

由所述第二电子设备:

接收所述已签名对象,

确定所述已签名对象是否有效,

在确定所述已签名对象有效之后,创建已生效的已签名对象,并将所述已生效的已签名对象传送到与同步群组相关的第三电子设备;

由所述第三电子设备:

从所述第二设备接收所述已生效的已签名对象,

确定所述已生效的已签名对象是否包括所述第一电子设备的有效签名和所述第二电子设备的有效签名,以及

如果所述已生效的已签名对象包括所述第一电子设备的所述有效签名和所述第二电子设备的所述有效签名,则将所述已生效的已签名对象发送给与所述同步群组相关的附加设备,否则不发送或不使用所述已生效的已签名对象;和

由所述第二电子设备或所述第三电子设备:

接收包括所述已签名对象或所述已生效的已签名对象的条目,

确定用于所述条目的查找密钥是否以与存储在数据存储中的对应条目的加密散列相匹配的值结束,其中,所述值包括所述第一电子设备的批准模式、受信任的重新签名者的批准模式、序列号和所述第一电子设备的签名,以及

只有在用于所述条目的查找密钥以所述值结束时才接受所述条目,否则拒绝所述条目。

2.如权利要求1所述的方法,其特征在于,生成对象并对所述对象签名以创建已签名对象包括:由所述第一电子设备:

向所述第一电子设备的存储服务授予批准;

用所述数据和所述批准创建所述对象;以及

将签名应用于所述对象以生成所述已签名对象。

3.如权利要求1所述的方法,其特征在于,所述确定所述已签名对象是否有效包括,由所述第二电子设备:

验证与所述已签名对象相关的远程过程调用的远程端是否被用于所述数据的访问控制表授权;

验证所述远程过程调用的所述远程端包括所述第一电子设备的公钥;以及

使用所述第一电子设备的所述公钥来验证所述已签名对象的签名。

4.如权利要求3所述的方法,其特征在于,所述确定所述已签名对象是否有效进一步包括,由所述第二电子设备:

从所述已签名对象中提取一个或多个批准名称;以及

验证提取的所述一个或多个批准名称满足所述访问控制表。

5.如权利要求1所述的方法,其特征在于,创建已生效的已签名对象包括,由所述第二电子设备,创建已生效的已签名对象以包括:

所述已签名对象;

从所述已签名对象提取的一个或多个批准名称;

所述第二电子设备的公钥;以及

所述第二电子设备的签名。

6.如权利要求1所述的方法,其特征在于:

确定所述已生效的已签名对象是否包括所述第一电子设备的所述有效签名和所述第二电子设备的所述有效签名包括:由所述第三电子设备来确定所述已生效的已签名对象包括所述第一电子设备的所述有效签名而不包括所述第二电子设备的所述有效签名;以及

所述方法进一步包括,由所述第三电子设备,要求所述第二电子设备提供具有所述第二电子设备的所述有效签名的更新的已生效的已签名对象。

7.如权利要求1所述的方法,其特征在于:

确定所述已生效的已签名对象是否包括所述第一电子设备的所述有效签名和所述第二电子设备的所述有效签名包括,由所述第三电子设备来确定所述已生效的已签名对象包括所述第二电子设备的所述有效签名而不包括所述第一电子设备的所述有效签名;以及

所述方法进一步包括,由所述第三电子设备,忽略所述已生效的已签名对象,或者要求所述第二电子设备为所述已生效的已签名对象获取所述第一电子设备的签名。

8.如权利要求1所述的方法,进一步包括,由所述第二电子设备或所述第三电子设备:

接收包括多个更新的对象的批次;

确定所述批次中的任意行是否由于所述行的本地版本与远程版本之间的差异存在冲突;以及

对于任何被确定存在冲突的行,将所述行添加到闭包中。

9.如权利要求1所述的方法,其特征在于,所述值还包括公钥。

10.如权利要求1所述的方法,进一步包括,由所述第二电子设备或所述第三电子设备:

接收包括所述已签名对象或所述已生效的已签名对象的条目;

确定用于所述条目的查找密钥是否包括受信任的重新签名者的批准模式;和

只有在用于所述条目的查找密钥包括所述受信任的重新签名者的所述批准模式时才接受所述条目,否则拒绝所述条目。

11.一种电子设备的系统,其特征在于,所述系统包括:

硬件处理器;和

存储设备;

其中所述系统还包括程序指令,所述程序指令被配置为通过以下方式使得所述系统中的第二设备的处理器共享对象:

从所述系统的第一电子设备接收所述对象作为已签名对象,

确定所述已签名对象是否有效,

在确定所述已签名对象有效后,创建已生效的已签名对象,并将所述已生效的已签名对象传送到与同步群组相关的第三电子设备;以及

所述系统还包括程序指令,所述程序指令被配置为通过以下方式使得所述系统中的所

述第三电子设备的处理器共享所述对象：

从所述第二设备接收所述已生效的已签名对象，

确定所述已生效的已签名对象是否包括所述第一电子设备的有效签名和所述第二电子设备的有效签名，以及

如果所述已生效的已签名对象包括所述第一电子设备的所述有效签名和所述第二电子设备的所述有效签名，则将所述已生效的已签名对象发送到与所述同步群组相关的附加设备，否则不发送或不使用所述已生效的已签名对象；以及

所述系统还包括附加的程序指令，所述附加的程序指令被配置为使所述第二电子设备或所述第三电子设备：

接收包括所述已签名对象或所述已生效的已签名对象的条目，

确定用于所述条目的查找密钥是否以与存储在数据存储中的对应条目的加密散列相匹配的值结束，其中，所述值包括所述第一电子设备的批准模式、受信任的重新签名者的批准模式、序列号和所述第一电子设备的签名，以及

只有在用于所述条目的查找密钥以所述值结束时才接受所述条目，否则拒绝所述条目。

12.如权利要求11所述的系统，其特征在于，所述系统还包括程序指令，所述程序指令被配置为通过以下方式使所述第一电子设备生成所述对象并对所述对象签名，以创建所述已签名对象：

向所述第一电子设备的存储服务授予批准；

用所述数据和所述批准创建所述对象；以及

将签名应用于所述对象以生成所述已签名对象。

13.如权利要求11所述的系统，其特征在于，被配置为使所述第二电子设备确定所述已签名对象是否有效的所述程序指令包括以下程序指令：

验证与所述已签名对象相关的远程过程调用的远程端是否被用于所述数据的访问控制表授权；

验证所述远程过程调用的所述远程端包括所述第一电子设备的公钥；以及

使用所述第一电子设备的所述公钥来验证所述已签名对象的签名。

14.如权利要求13所述的系统，其特征在于，被配置为使所述第二电子设备确定所述已签名对象是否有效的所述程序指令进一步包括以下指令：

从所述已签名对象中提取一个或多个批准名称；以及

验证提取的所述一个或多个批准名称满足所述访问控制表。

15.如权利要求11所述的系统，其特征在于，创建已生效的已签名对象的所述程序指令包括程序指令，该程序指令被配置为使所述第二电子设备创建所述已生效的已签名对象以包括：

所述已签名对象；

从所述已签名对象提取的一个或多个批准名称；

所述第二电子设备的公钥；以及

所述第二电子设备的签名。

16.如权利要求11所述的系统，其特征在于：

确定所述已生效的已签名对象是否包括所述第一电子设备的所述有效签名和所述第二电子设备的所述有效签名的所述程序指令,包括程序指令使所述第三电子设备:

确定所述已生效的已签名对象包括所述第一电子设备的所述有效签名但不包括所述第二电子设备的所述有效签名;以及

要求所述第二电子设备提供具有所述第二电子设备的所述有效签名的更新的已生效的已签名对象。

17.如权利要求11所述的系统,其特征在于:

使所述第三电子设备确定所述已生效的已签名对象是否包括所述第一电子设备的所述有效签名和所述第二电子设备的所述有效签名的所述程序指令,包括程序指令使所述第三电子设备:

确定所述已生效的已签名对象包括所述第二电子设备的所述有效签名但不包括所述第一电子设备的所述有效签名;以及

忽略所述已生效的已签名对象,或者要求所述第二电子设备来为所述已生效的已签名对象获取所述第一电子设备的签名。

18.如权利要求11所述的系统,进一步包括附加的程序指令,所述附加的程序指令被配置为使所述第二电子设备或所述第三电子设备:

接收包括多个更新的对象的批次;

确定所述批次中的任意行是否由于所述行的本地版本与远程版本之间的差异存在冲突;以及

对于任何被确定存在冲突的行,将所述行添加到闭包中。

19.如权利要求11所述的系统,其特征在于,所述值还包括公钥和序列号。

20.如权利要求11所述的系统,进一步包括被附加的程序指令,所述附加的程序指令被配置为使所述第二电子设备或所述第三电子设备:

接收包括所述已签名对象或所述已生效的已签名对象的条目;

确定用于所述条目的查找密钥是否包括受信任的重新签名者的批准模式;和

只有在用于所述条目的查找密钥包括所述受信任的重新签名者的所述批准模式时才接受所述条目,否则拒绝所述条目。

# 在二进制数据同步协议中交换的更新的签名

[0001] 相关的申请和优先权要求

[0002] 本专利文件要求2015年10月2日提交的申请号为62/236,435的美国临时专利的优先权,其公开内容通过引用被全部结合到本文档中。

## 背景技术

[0003] 本公开描述了用于对两个以上电子设备之间的通信进行识别、认证和授权的机制。

[0004] 当诸如移动电子设备和服务器之类的电子设备参与经由网络的通信,并且特别是当它们共享对象并且多个实体具有更新对象的权限时,如果设备不明白授权的更新何时完成,可能会出现错误。

[0005] 本文档描述了旨在改进电子设备的分布式系统中的认证和/或授权的方法和设备。

## 发明内容

[0006] 在各个方面中,一种系统实施控制在分布式系统中的实体之间的对象的共享更新的方法。例如,同步群组可以包括多个电子设备。群组的第一电子设备生成对象,对对象签名并将已签名对象传送给群组的第二电子设备。第二电子设备接收已签名对象,确定已签名对象是否有效,以及在确定已签名对象有效之后,生成已生效的已签名对象,并将已生效的已签名对象传送给群组的第三电子设备。第三电子设备从第二设备接收已生效的已签名对象,确定已生效的已签名对象是否包括第一设备和第二设备两者的有效签名,并且只有当其在已生效的已签名对象中找到有效签名时,其才进一步分发或使用更新。

[0007] 可选地,当生成对象并对对象签名以创建已签名对象时,第一电子设备可以向第一电子设备的存储服务授予批准(blessing),用数据和批准创建对象,并且将签名应用于对象以生成已签名对象。

[0008] 可选地,当确定已签名对象是否有效时,第二电子设备可以:(i)验证与已签名对象相关的远程过程调用的远程端是否被用于数据的访问控制表授权;(ii)验证远程过程调用的远程端包括第一电子设备的公钥;和(iii)使用第一电子设备的公钥来验证已签名对象的签名。

[0009] 可选地,当确定已签名对象是否有效时,第二电子设备可以从已签名对象的批准中提取一个或多个批准名称,并且第二电子设备可以验证提取的一个或多个批准名称满足访问控制表。

[0010] 可选地,当创建已生效的已签名对象时,第二电子设备创建已生效的已签名对象以包括:(i)已签名对象;(ii)从已签名对象提取的一个或多个批准名称;(iii)第二电子设备的公钥;和(iv)第二电子设备的签名。

[0011] 可选地,当确定已生效的已签名对象是否包括第一电子设备的有效签名和第二电子设备的有效签名时,第三电子设备可能确定已生效的已签名对象包括第一电子设备的有

效签名而不包括第二电子设备的有效签名。如果是这样的话，则第三电子设备可以要求第二电子设备提供具有第二电子设备的有效签名更新的已生效的已签名对象。

[0012] 可选地，当确定已生效的已签名对象是否包括第一电子设备的有效签名和第二电子设备的有效签名时，第三电子设备可以确定已生效的已签名对象包括第二电子设备的有效签名而不包括第一电子设备的有效签名。如果是这样的话，则第三电子设备可以忽略已生效的已签名对象，或者要求第二电子设备为已生效的已签名对象获取第一电子设备的有效签名。

[0013] 可选地，第二电子设备或第三电子设备可以：接收包括多个更新对象的批次；确定批次中的任意行是否由于行的本地版本与远程版本之间的差异而存在冲突；并将任何被确定存在冲突的行添加到闭包(closure)中。

[0014] 可选地，第二电子设备或第三电子设备可以：(i)接收包括已签名对象或已生效的已签名对象的条目(entry)；(ii)确定用于条目的查找密钥是否以与存储在数据存储中的对应条目的加密散列(hash)匹配的值结束；(iii)只有在用于条目的查找密钥以该值结束时才接受该条目，否则拒绝该条目。在一些实施例中，值可以包括公钥和序列号。在一些其他实施例中，值可以包括：(a)第一设备的批准模式；(b)受信任的重新签名者的批准模式；(c)序列号；和(d)第一设备的签名。

[0015] 可选地，第二电子设备或第三电子设备可以：(i)接收包括已签名对象或已生效的已签名对象的条目；(ii)确定用于条目的查找密钥是否包括受信任的重新签名者的批准模式；(iii)只有在用于条目的查找密钥包括受信任的重新签名者的批准模式时才接受该条目，否则拒绝该条目。

[0016] 在另一方面，为了使分布式系统中的电子设备之间共享的对象生效，至少包括第一电子设备、第二电子设备和第三电子设备的同步群组的第三电子设备将实施一种方法，该方法包括：接收对象，以及通过确定对象是否包括第一电子设备的有效签名和第二电子设备的有效签名来确定接收的对象是否是已生效的已签名对象。如果接收的对象包括第一电子设备的有效签名和第二电子设备的有效签名，则第三电子设备将发送接收的对象到与同步群组相关的附加设备，否则它不发送或不使用接收的对象。

[0017] 可选地，在前一段落的系统中，当确定已生效的已签名对象是否包括第一电子设备的有效签名和第二电子设备的有效签名时，第三电子设备将会确定已生效的已签名对象包括第一电子设备的有效签名而不包括第二电子设备的有效签名。如果是这样的话，则第三电子设备可以要求第二电子设备提供具有第二电子设备的有效签名的更新的经过验证的签名的对象。

[0018] 可选地，在该方面，当确定已生效的已签名对象是否包括第一电子设备的有效签名和第二电子设备的有效签名时，第三电子设备可以确定已生效的已签名对象包括第二电子设备的有效签名而不包括第一电子设备的有效签名。如果是这样的话，第三电子设备可以要求第二电子设备来使已生效的已签名对象中的第一电子设备的签名生效。

[0019] 在另一个方面，至少包括第一电子设备、第二电子设备和第三电子设备的同步群组的第二电子设备，可以通过接收已签名对象来使在设备之间共享的已签名对象生效，并确定已签名对象是否有效。在确定已签名对象有效之后，第二电子设备可以创建包括第二电子设备的签名的已生效的已签名对象。第二电子设备可以将已生效的已签名对象传送到

与同步群组相关的第三电子设备。

[0020] 可选地，在该方面，当确定已签名对象是否是有效的时，包括，第二电子设备可以：(i)验证与已签名对象相关的远程过程调用的远程端是否被用于数据的访问控制表授权；(ii)验证远程过程调用的远程端包括第一电子设备的公钥；和(iii)使用第一电子设备的公钥来验证已签名对象的签名。

[0021] 可选地，在该方面，当确定已签名对象是否有效时，第二电子设备可以从已签名对象的批准中提取一个或多个批准名称。第二电子设备还可以验证提取的一个或多个批准名称满足访问控制表。

[0022] 可选地，在该方面，第二电子设备可以创建已生效的已签名对象以包括：(i)已签名对象；(ii)从已签名对象提取的一个或多个批准名称；(iii)第二电子设备的公钥；和(iv)第二电子设备的签名。

## 附图说明

[0023] 图1示出了根据各种实施例的包括各种客户端设备和服务器的分布式系统的示例。

[0024] 图2示出了已签名对象的元素示例，该已签名对象具有在分布式系统的设备之间将被分享的数据。

[0025] 图3示出已生效的已签名对象的元素示例。

[0026] 图4示出分布式系统中的对象验证处理示例。

[0027] 图5示出当对象更新被分批接收时，解决冲突的示例处理。

[0028] 图6是根据各种实施例的可以被用于包含或实施程序指令以及与其他设备的通信的硬件示例的框图。

## 具体实施方式

[0029] 与本公开相关的术语包括：

[0030] “电子设备”或“计算设备”指的是包括处理器和存储器的设备。每个设备可以有它自己的处理器和/或存储器，或者处理器和/或存储器可以与例如虚拟机或容器布置中的其他设备共享。存储器可以包含或接收编程指令，当编程指令由处理器执行时，使得电子设备根据编程指令执行一个或多个操作。电子设备的示例包括：个人计算机，服务器，主机，虚拟机，容器，游戏系统，电视机，以及诸如智能手机、个人数字助理、照相机、平板电脑、膝上型计算机、媒体播放器等的移动电子设备。在客户端-服务器布置中，客户端设备和服务器各自是电子设备，其中服务器包含客户端设备经由一个或多个通信网络中的一个或多个通信链路访问的指令和/或数据。在虚拟机布置中，服务器可以是电子设备，并且每个虚拟机或容器也可以被认为是电子设备。在下面的讨论中，为了简洁起见，客户端设备、服务器设备、虚拟机或容器可以简称为“设备”。

[0031] 在本文档中，术语“处理器”和“处理设备”指的是被配置为执行编程指令的电子设备的硬件组件。除非特别声明，否则单数术语“处理器”或“处理设备”意在将单个处理设备实施例和其中多个处理设备一起或共同执行处理的实施例都包括。

[0032] 在本文档中，术语“存储器”、“存储器设备”、“数据存储”、“数据存储设施”等每一

个都指非临时性设备，在其上计算机可读数据、编程指令或两者被存储。除非特别说明，否则术语“存储器”、“存储器设备”、“数据存储”、“数据存储设施”等意在包括单个设备实施例、其中的多个存储器设备一起或共同存储一组数据或指令以及这些设备中的各个分区的实施例。

[0033] “对象”是服务器控制对已授权客户端设备的访问，和可以由一个或多个已授权客户端设备更新的进程、数据集或其他资源、功能或事务。

[0034] 识别和/或验证设备或其用户的处理可以被称为“认证”。识别一个或多个设备或用户可以执行或使用什么动作或服务的处理可以被称为“授权”。

[0035] “主体(principal)”是(或在用户为特定用户正在使用的电子设备的情况下)发布请求的实体(即，电子设备)。主体具有身份并可以通过名称和/或密钥被识别。

[0036] “批准(blessing)”是针对主体的一组密码或其他电子呈现的证书。它证明一个实体支配某些权力的能力。可选地，批准可以包含一个或多个人类可读的名称，可以被称为“批准名称”。为简便起见，本文档可以使用简略术语“批准”来同样指代基于上下文含义明确的批准名称。

[0037] “人类可读名称”是能够被人自然地阅读和理解的数据的表示，例如由英语或其他人类语言的字符构成的一个表示，可选地具有一个或多个符号，该符号具有被人类所理解的含义。示例包括个人姓名、包含表示“at”的@符号的电子邮件地址等。

[0038] 术语“同步群组”意在指一组设备，该组设备允许彼此之间同步一组数据，和/或向数据存储设施同步一组数据。

[0039] “访问控制表”或“ACL”是列表、表格、或者识别附加于特定对象或对象集的权限的其他数据结构。权限可能涉及单个设备或多个设备。

[0040] 如本文中所使用的，除非上下文另有明确规定，否则单数形式“一个”“个”“该”包括复数形式。除非另外定义，否则本文使用的所有技术和科学术语具有与本领域普通技术人员通常的理解相同的含义。

[0041] 如本文中所使用的，术语“包括”意味着“包括，而不局限于”。

[0042] 图1示出分布式系统的各种实施例，在该分布式系统中，一个或多个客户端设备12a……12n经由一个或多个通信网络20，与一个或多个如服务器16a…16n的其他电子设备通信，一个或多个通信网络20诸如无线电话网络、因特网、内联网、局域网、广域网、其他类型的网络、或这些网络的任意组合其他电子设备通信诸之类。服务器16a…16n的任何一个可以被用于使一个或多个对象可用于已授权客户端设备。此外，一个或多个服务器(例如16b)可以作为证书服务器或另外地存储诸如用于诸如访问控制表之类的结构中的任何客户端设备12a...12n的加密密钥的访问证书。任何服务器16n还可以是存储一个或多个群组的例如群成员列表之类的细节的群服务器。此外，任何客户端设备(例如12a)都可以将访问各种对象的权力委托给一个或多个其他客户端设备(例如12b)。

[0043] 在本文档中，客户端设备可以被分成同步群组，在同步群组之间能够同步对象或对象集。任何特定的设备可以与一个或多个同步群组相关。每个同步群组中的设备将会存储识别是同步群组的成员的其他设备的ACL或其他数据结构。(注意：在这里，术语“设备”可以指实际的电子设备标识符，或指使用特定电子设备的用户的标识符。)

[0044] 本文档描述了对于在同步协议期间为可以被设备之间交换的对象，对对象签名并

验证对象签名的方案。本文档中描述的过程可以使设备之间交换的对象呈现完整性、真实性和一致性。完整性的特征意味着系统能够确定对象是否由于该对象被其作者首次传输而被篡改。真实性的特征意味着系统能够确定创作(author)对象的初始设备的批准名称。一致性的特征意味着如果对象被一个系统认为有效,随后它能够被实施本文档中描述的处理的其他系统认为有效。

[0045] 本文档还描述了设备或服务可以确定对象是否有效的若干动作。在不限制本公开的情况下,一种可以发生有效性的确定的方法涉及:(i)检查被签名的散列,以确认散列与数据匹配;(ii)通过一系列证书确认签名与第一设备绑定,所有证书都有与其各自内容相匹配的散列;(iii)确认在证书串的第一证书之后签名的每个连续证书由在该证书串中前面的证书中提及的公钥签名;(iv)确认第一证书被第二设备的可信根集中列出的密钥签名;并且串中的所有密钥、签名和证书都没有过期,并且所有的密钥都没有被撤销。

[0046] 本文描述了对对象(它可以被称为例如二进制大对象或“blobs”,虽然对象不一定是二进制大对象)签名时可能涉及的步骤。本文档还描述了用于处理已签名对象以对其进行认证并确定其完整性和真实性的步骤。Blob的示例在图2中被示出,其中已签名的blob(通过“sb”表示)201是具有签名205、一个或多个批准204以及数据203的元素的数据结构。数据203是正在被同步的对象(即,数据),并且它可以包括诸如用于重播保护的序列号和同步相关元数据的信息。$Blessings_{author-sb}$是绑定到创作对象的系统或设备的批准204。创作系统通常从设备获取这些批准,代表它正在同步数据。系统利用其私钥($S_{author-sb}$)生成签名205。在下面的讨论中,对于已签名对象“sb”,讨论可以使用短语数据(sb)、批准(sb)和签名(sb)来指代例如在图2中示出的已签名对象的数据、批准和签名组件。下面的讨论也可以使用术语AuthorPublicKey(sb)来指代批准的公钥(sb),这也是已签名对象的作者的公钥。

[0047] 在这个系统中,实体(即设备)在其更新对象时对其更新进行签名。这允许第一设备X更新一些数据,并将其传递给第二设备Y。设备Y可以被允许查看数据,但不能修改它。设备Y接着将其传递给第三设备Z。设备Z从设备Y接收更新,但不允许其进行更新,但是Z可以分辨该更新最初是由具有权限的设备X创建的。因此,设备Z可以应用更新,知道未违反安全性。

[0048] 在更新传播期间,如果实体更新对象的权力被撤销,则可能会出现问题。例如,如果在上述情况下,在更新被转移到设备Y之后但是在更新从Y转移到设备Z之前如果设备X被报告被盗,则设备Z可以接收由设备X生成的已授权的更新。然而,当设备Z不明真相地检查更新是否被授权时,它可能获悉设备X的证书不再有效(因为该设备被报告被盗了),因此它会拒绝更新。由于设备Y接受了更新,这会导致冲突,并且各种设备上的数据可能会有分歧。

[0049] 在本实施例中,系统将会使用通过设备Y增加的第二签名,以使更新生效和允许更新继续传播通过同步群组。设备Y可以由此创建已生效的已签名对象。图3示出了已生效的已签名对象301的数据结构示例,已生效的已签名对象301包括初始的已签名对象201。已生效的已签名对象301还包括$BlessingNames_{author-sb}$,它们是由校验者(validator)确定的已签名对象201的作者的批准名称303。校验者可以通过验证在已签名对象201最初由验证者的系统接收的安全性内容中封装在签名的对象201内的批准来获取这些。已生效的已签名对象301还包括$P_{validator-sb}$,它是校验者系统的公钥,以及通过校验者系统使用其私钥($S_{validator-sb}$)生成的签名305。在下面的讨论中,对于已签名对象“sb”,讨论可以使用

phrases data(vsb)、BlessingNames(vsb)、validatorPublicKey(vsb)和signature(vsb)来指代如图3中所示的已生效的已签名对象的数据、批准名称公钥和签名组件。

[0050] 下面是来自上文描述的实施例的实现实例的一些描述性评述：

[0051] 系统利用公开密钥签名对对象更新进行签名，并允许这些签名在其他节点被检查。它有两个操作:Sign()和Check()。它们以DataWithSignature类型进行操作，该类型包括数据(Data)(要被签名的数据)，随后返回项(Item)的向量，这些Item是任意数据项。该功能可以对跨同步群组的数据同步有用，因为一旦签名被同步群组的至少一个成员接受，其可以允许在跨同步群组传播期间中签名保持有效，即使在此期间初始密钥或其批准无效。

[0052] 可能有三种类型的参与者:(1)创建更新并用Sign()操作对其进行签名的“作者”;(2)一个或多个“校验者”，每个“校验者”都直接从作者处接收更改，并应用Check()操作来使其生效;以及(3)零个以上“检查者”，每个检查者都从校验者或另一个检查者处接收更改，并应用Check()检查它。

[0053] 校验者检查由创建者提供的签名和批准，然后校验者附加其自己的签名，创建已生效的已签名对象，证明作者的签名在校验者看到它时是好的的事实。

[0054] 检查者检查作者和校验者两者的签名，但是对于签名有效性使用比校验者弱的检查。特别地，对于密钥期满它使用显著的宽限期，以便如果密钥或批准在更改被准许后但是被完全地传播之前被撤销，由校验者准许到同步群组的更改有机会传播到同步群组中的所有的节点。目的可以是宽限期被选择为大于同步群组的直径(在时间上测量)。一种保证的方式是坚持成员至少每T时间单位与中央服务器同步，并使宽限期为2T或T的某个其他倍数或函数。中央服务器可重新对数据签名以允许新成员拾取该数据。

[0055] 被写入系统的更新可以非常小(可能为几十个字节)，但公钥签名或验证可能需要毫秒的数量级(实际上非常大)。检查者执行两项这种验证。为了解决这个并减少性能问题，系统可以批量进行对象更新，以便将一次签名检查应用到多个更新。因此DataWithSignature中的Data可以是Item的向量，而不是单个Item。

[0056] 然而，系统不会总是想要将所有的更新放到相同的批次中。例如，作者和校验者可以共享具有不同成员资格的两个不同的同步群组。在这种情况下，作者可能会将一个同步群组的批次与其他同步群组的批次分开，尽管作者的批准和校验者身份对于所有批次都是相同的。因此，系统可以将作者的批准数据和校验者的密钥数据与签名批次本身分开，以便即使发送了几批更新，批准数据和校验者数据能够被一次处理。

[0057] ValidationCache可以被用于分别保存这些数据，并且允许其仅被发送一次，而不是每个签名一次。

[0058] 作为最后一个例子，如果作者向校验者发送一批(例如10个)更新，并且随后校验者与仅允许查看一半更新的检查者同步;或许ACL阻止其查看其他的。这要求即使批次中的某些更新被移除，批次上的签名仍然有效。这可以通过Item类型实现，Item类型是一种数据编码方案联合类型，其包含更新的编组形式的字节，或者(如果更新不能被发送)(可以用SumByteVectorWithLength()计算的)数据的SHA-256(或其他)散列。

[0059] 图4是示出实施上述实施例的处理的流程图。第一设备(D1)执行应用程序，该应用程序识别数据401以同步同步群组内的一些数据，这将向第一设备的存储服务(例如Syncbase)授予批准402，该第一设备是密钥值存储系统，该存储系统处理结构化数据和/或

对象，例如blobs。第一设备的存储服务代理将会使用其私钥创建带有数据、授予的批准和签名的已签名对象403。已签名对象可以是新对象或之前创建的对象的更新。第一设备的存储服务代理随后将已签名对象发送到一个或多个其他设备404，以将创建的已签名对象与同步群组中的其他设备同步。

[0060] 第二设备(D2)的存储服务代理将接收已签名对象405并且通过使已签名对象406生效，即通过确定已签名对象是否有效，来充当校验者。它可以通过以下方式使已签名对象406生效：(i)验证与已签名对象一起到达的远程过程调用(RPC)的远程端由相关的同步ACL授权；(ii)验证RPC的远程端具有作者的公钥authorPublicKey(sb)，以确保已签名对象的作者是使RPC与它同步的作者；(iii)使用authorPublicKey(sb)验证已签名对象的签名(sb)的签名；(iv)在已签名对象被接收的远程过程调用的上下文中验证批准(sb)，并提取批准的批准名称(由$blessingNames_{author-sb}$表示)；和(v)验证$blessingNames_{author-sb}$是否满足数据(sb)的相关ACL。在验证已签名对象406之后，第二设备能够确定已签名对象具有完整性并且是由通过相关的ACL授权的存储服务代理创作。

[0061] 在验证已签名对象之后，第二设备将创建已生效的已签名对象407，并通过将已生效的已签名对象传送到同步群组中的一个或多个附加设备408，将已生效的已签名对象传播。为了创建已生效的已签名对象407，第二设备将使用其私钥在这些项上包括已签名对象、$BlessingNames_{author-sb}$、其自己的公钥以及签名signature(vsb)。

[0062] 当同步群组中的第三设备接收到已生效的已签名对象时409，第三设备可以通过确定已生效的已签名对象是否包括作者(第一设备)和第二设备两者的有效签名，来对接收到的对象进行生效处理410。第三设备可以通过以下方式来实现这个：(i)验证与已生效的已签名对象一起到达的远程过程调用(RPC)的远程端是由相关的同步ACL授权的；(ii)验证validatorPublicKey(vsb)不等于authorPublicKey(sb)，以确保该对象不是由原作者生效的；(iii)检查以确保validatorPublicKey(vsb)和authorPublicKey(sb)未超过任何到期阈值加上任何可适用的宽限期，例如通过参考从同步群组的服务器获取的证书撤销列表(注意：如上，此举可以由第二个设备完成作为生效批准的一部分)；(iv)使用authorPublicKey(sb)验证已生效的已签名对象内的已签名对象上的第一设备的签名；(v)验证BlessingNames(vsb)是封装在批准(sb)中的批准名字的子集；以及(vi)验证BlessingNames(vsb)满足对象中的数据或相关ACL。

[0063] 如果第三设备确定已生效的已签名对象是有效的(411是)，随后其可以进一步将对象发送到同步群组中的其他设备，可选地，可以通过创建带有其自己的签名的已生效的已签名对象的更新版本来发送。如果第三设备不能确定已生效的已签名对象是有效的(411否)，然后其可以试图解决生效问题414。例如，如果第一设备(signature(sb))的签名不是有效的，第三设备可以要求第二设备来获取第一设备的签名，否则其可以忽略已生效的已签名对象。如果第二设备的签名(signature(vsb))不是有效的，第三设备可能需要第二设备提供带有有效签名(vsb)的被更新的已生效的已签名对象。如果第三设备能够解决生效问题(415是)，第三设备将会只进一步传播更新。否则(415否)，其不会进一步分发或使用对象416。

[0064] 在一些实施例中，一个或多个设备和它们安装的应用程序可能分批地执行对对象的更新和/或生成对象，这可以在数据存储中本地应用。一批包括原子方式(atomically)写

入同步系统而行之间有一些恒定关系的一行或多行对象。如果行有时被两个设备独立修改，以致每个设备在没有信息表明其他设备也修改了该行时修改了该行，则认为该行发生冲突。当设备中的任何一个或其他设备意识到两次修改时，在跨设备同步期间，冲突被检测到。如果设备的权限允许，发现冲突的该设备将尝试通过在修改之间进行选择，或者通过将它们以某种方式组合到一个新的修改中来解决冲突。冲突与批次相互作用，因为系统必须解决冲突，同时允许批次原子级地出现。如果批次中的某一行存在冲突，则整个批次可能会被视为存在冲突。

[0065] 另外，如果批次B1中的行P与批次B2存在冲突，而批次B1中的行Q与批次B3存在冲突，那么三个批次{B1，B2，B3}的群组一起存在冲突。冲突解决算法创建冲突在一起的批次的闭包，在闭包外部没有批次与内部的批次存在冲突。

[0066] 系统可以将整个闭包流作为单个冲突，调用到正在使用存储服务解析的设备应用程序。

[0067] 存在冲突的行可以有两个版本：(1)本地(设备本地已知的版本)和(2)远程(通过同步协议从其他设备获取的版本)。

[0068] 为了查找批次的闭包，系统可以实施例如图5所示的处理。系统将审查每行以确定其是否存在冲突501。如果对于该行，有本地版本和不同的远程版本，则该行处于冲突，并且那些版本可以是本地批次和远程批次的一部分。当系统识别出行处于冲突时502，系统将尝试为其创建闭包。如果处于冲突的行已经被添加到先前创建的闭包(503是)，则系统可以跳过它(即，对该行不进行任何进一步操作)并移动到下一行504。如果它没有被添加到之前的闭包，系统可以对该行创建闭包505。对行创建闭包505涉及向闭包添加行、查找该行属于的本地和远程批次、以及将所有属于这些批次的行添加到该闭包。如果新行本身存在冲突，那么系统可以通过向闭包添加更多批次来递归处理上述这些行中的每一行。如果已经看到批次，则系统可以跳过处理它。最后，作为输入给出的行集合将被分组为不相交的闭包，每个闭包包含彼此冲突的但不与闭包之外的任何批次冲突的批次行。

[0069] 一旦冲突批次的闭包被识别，设备必须确定其是否被允许解决该冲突。如果这种解决需要设备向行写入新值并且它没有这样做的权限，则设备可能不被允许解决该冲突。如果设备具有解决该冲突的权限，存储服务代理的参数或客户端设置将指示要使用哪个冲突解决算法。算法可以是简单的，例如拾取带有最近时间戳的更新。在更复杂的实施例中，系统可以向应用程序提供相关的更新批次(连同在任何改变之前的数据值)，应用程序可以使用依赖于应用程序的规则来选择将什么写入行。如果设备不被允许解决冲突，其反而可以将所有的各种更改保留在存储器中，但不会使更改对应用程序可见。更改将最终(通过该设备或其他)传播到一些具有解决冲突的权限的设备。解决的值随后被传播到其他设备。同步群组可以被布置以便至少一个设备(例如服务器设备)具有解决冲突的权限。

[0070] 举例来说，考虑下面的情形：

[0071] (a)第一设备D1将一组行{X，Y}写入作为批量B1的一部分。

[0072] (b)第一设备D1将一组行{Z，A}写入作为批量B2的一部分。

[0073] (c)第一设备D1将一组行{p，q}写入作为批量B3的一部分。

[0074] (d)第二设备D2将行{Y，Z}写入作为批次B4的一部分。

[0075] (e)第二个设备写入行{p，q}作为批次B5的一部分。

[0076] 假设设备D1是从设备D2接收更新的本地设备。处于冲突的行将是Y、Z、p和q。在这个例子中，行X和A没有冲突。该系统将为这种情况创建两个闭包，导致如下两个单独的冲突解决调用：

[0077] (1)由于B1和B4在Y上相互冲突，B2和B4在Z上冲突，所以形成以下闭包：{X(B1)，Y(B1,B4}，Z(B2,B4)，A(B4)}。

[0078] 由于B3和B5在p和q上彼此冲突，因此形成以下闭包：{p(B3,B5)，q(B3,B5)}。

[0079] 在进一步的实施例中，系统可以包括新对象的限制写入或根据指定协议更新现有对象的处理。这种处理可以使系统能够使用较不详细的ACL，或者在一些实施例中根本不使用ACL。在上述处理中，签名至少可作为两个目的用：(1)面对潜在恶意存储服务代理，允许同步群组内的正确的存储服务代理实现的收敛；以及(2)允许应用程序控制哪些设备可以写入更新，以及更新可以被写入到哪里。以下讨论中描述的处理实现第三个目的：允许应用程序发现什么设备已写入更新，以及该更新被写入在哪里。

[0080] 在该选项中，密钥可以被用于处理收敛。同步群组的读写成员设备可以对他们同步到其他设备的对象签名，并且只读成员可以不更改地发送这些签名。签名可以有比与同步群组相关的时间直径长的超时。时间直径可以是数据从作者处获取到同步群组的任何其他设备成员需要花费的最大时间。

[0081] 在例如上述的那些系统中，可以添加对受限的权限的支持而不需要附加的签名。这可以通过向存储服务代理中的每个条目添加查找密钥并使用以下任意选项来完成：(1)不可变条目；(2)未经认证的单写入者条目；(3)经过认证的单写入者条目；(4)访问控制条目。系统没有必要包括所有的这些选项或这些选项中的任何一个选项；他们中的任何一个可以或可以不在任何系统设计中被实施。

[0082] 不可变条目。

[0083] 第一选项，不可变条目，是能够被任何设备写入一次的值，但是决不能被改变或删除。用此选项，如果系统具有以“$hash＝<hash_of_value>$”结尾的查找密钥，则系统会将条目(即存储服务代理程序中的对象)视为不可变的，其中“<hash_of_value>”是存储在条目中的值的加密散列。系统会拒绝删除这种条目，并且拒绝将其覆盖，除非该值有指定的散列，这样保证该值不变。这些检查会通过系统中的每个设备执行，以便试图改变或移除对象的恶意系统不被信任。

[0084] 在一些实施例中，用于解决冲突的程序可以被布置为通过从值中任意拾取一个，来使得系统立即解决这些条目上的所有冲突，因为所有合法值将是相等。可选地，如果要存储的值明显小于散列，则系统可以将该值而不是它的散列存储在查找密钥中。

[0085] 用该选项，想要写入条目(<lookup_key>，<value>)的应用程序范例将写入(<lookup_key_prefix>$hash＝<hash_of_value>$，<value>)作为替代。想要在不知道散列的情况下查找值的客户端将扫描数据集以查找任何此类查找密钥。

[0086] 用该选项，未被授权的设备将不能覆盖或删除合法的条目。

[0087] 未经认证的，单写入者条目。

[0088] 第二选项，未经认证的单写入者条目，是能够被给予写访问权的同步群组成员写入，但是另外未经认证的值。条目的原写入者能够根据需要将其多次写入。该技术通过添加额外的条件建立在不可变条目上，额外的条件允许被条目的第一写入者进行的更改。

[0089] 用该选项，系统可以接受查找密钥以“$hash＝<hash_of_value>$”结尾的条目，其中“<hash_of_value>”是存储在系统已存储的条目的对应版本中的第一值的加密散列。第一值具有至少两个字段(field)：

[0090] (1)authkey:公钥;和

[0091] (2)squence:序列号，其中0是初始值。

[0092] 系统会拒绝删除这种条目，并且允许其将仅被具有三个字段的值覆盖，字段带有下面的三个特性：

[0093] (1)authkey:与上述第一值相同的公钥。

[0094] (2)squence:大于被覆盖的值中的“sequence”的值。

[0095] (3)signature:除“signature”之外的所有字段的签名，用与“authkey”对应的密钥签名。

[0096] 如果更新的条目不满足以上特性，系统可以拒绝该条目。可选地，条目的若干版本可以存储在系统的各个位置。如果是这样的话，随后当无用集合(garbage collection)被应用于条目的有向无环图(DAG，directed acyclic graph)时，系统可以只保存DAG中的条目的版本，其值与查找密钥中嵌入的散列匹配，并且其可以抛弃其他条目。这允许新加入的成员来验证较新的值已经被绑定到查找密钥的“authkey”密钥签名。

[0097] 如果在条目的DAG中发现冲突，系统可以选择具有最高序列号的版本并抛弃带有低序列号的条目。

[0098] 因此，在这个选项中，想要在给定的查找密钥处写入一系列值的应用程序首先创建公钥对。该应用程序会将其第一值的散列附加到查找密钥中，如在“不可变条目”方案中。接着，随后的值将使用秘密密钥进行签名。

[0099] 如果多个设备(可能与同一个人或组织相关)想要写入相同的查找密钥，那么设备将共享与“authkey”密钥对应的秘密密钥。在该实施例中，只有这种设备能够解决冲突。

[0100] 这些条目不能被删除，但是为达到类似的效果，写入者会用使读取者忽略的最小的条目覆盖该值。

[0101] 经过认证的单写入条目。

[0102] 第三选项，经过认证的单写入者条目，是能够被任何一个具有写访问权的同步群组成员写入并指定批准模式的值。任何具有支配匹配该模式的批准的写访问权的成员同样可以写入。一旦设定，批准模式可能不会改变。此技术通过使用批准名称和证书链而不是未经认证的公钥，建立在第二选项(未经认证的单写入者条目)上。

[0103] 在该选项中，系统将识别查找密钥以“$hash＝<hash_of_value>$”结尾的条目，其中“<hash_of_value>”是存储在条目中的第一值的加密散列。第一值包括至少四个字段：

[0104] (1)author:原作者的批准模式。

[0105] (2)re_signer:受信任的重新签名者的批准模式，可以为原作者签名。

[0106] (3)sequence:序列号，其中0是初始值。

[0107] (4)signature:除“signature”以外的所有字段的签名，其签有一个与“作者”匹配的签名批准。

[0108] 系统可以拒绝接受不具有这种字段的条目，并且其可以允许使用只具有以下字段和特性的值覆盖现有条目：

[0109] (1)author:与上述先前的值的“author”字段中相同的批准模式。

[0110] (2)re_signer:与先前的值的“re_signer”字段相同的批准模式。

[0111] (3)sequence:二者之一:

[0112] a.大于被覆盖的值中的“sequence”的值,如果签名是由“author”签名,或

[0113] b.与先前的“sequence”相同的值,提供除具有与先前值相同的值并且签名是由

[0114] “re_signer”签名的“signature”之外的所有字段。

[0115] (3)signature:除“signature”之外的所有字段的签名,用与“author”或“re_signer”匹配的签名批准进行签名。

[0116] 当无用集合被应用于条目的DAG时,系统可以抛弃除值与嵌入在查找密钥中的散列匹配的那一个之外的条目版本。这允许新加入的存储服务代理来验证较新的值已被绑定到查找密钥的“author”签名。

[0117] 如果在条目的DAG中发现冲突,系统可以选择具有最高序列号的条目。在具有相同序列号的两个值之间,系统可以选择由“re_signer”签名的值而不是由“author”签名的值;另外这些值可能是等同的。

[0118] 该选项中存在指定的“re_signer”以允许密钥翻转。其将对由作者签名的值进行重新签名,并定期对它之前签名的值重新签名,以防止签名、批准或密钥中的任何一个过期。任何处理长寿命数据的应用程序都可以使用这种重新签名者,并且可以为该角色指定同步群组的服务器成员。应用程序可以为服务器提供重新签名的批准,告诉服务器更新其的签名的频率,以及可选地还指定哪个前缀应当被重新签名以避免服务器不得不搜索整个集合。

[0119] 如果条目的签名已过期,系统可以拒绝带有已过期或在数据将到达同步群组的其它成员之前将要过期的另外的有效的“signature”字段的Put()调用。当通过同步协议接收值时,系统仍然可以接受另外的有效但过期的“signature”字段。这可能有助于阻止已过期的签名的引入,但如果创建了带有过期签名的条目,则多个同步群组成员可以看到同样过期条目的一致视图。无用集合代码可以包含保存最后未过期的值(如果有一个的话)的指令,来帮助重新签名者修复该情况。冲突解决代码还可以包括规则,该规则越过到期条目优先选择未过期条目。可选地,如果他们满足指定的标准,重新签名者可以对到期的“author”签名重新签名或重新签名者可以移除过期的条目或用更早的未过期条目替换它们。

[0120] 用该选项,如在“不可变条目”方案中的,想要在给定的查找密钥中写入一系列值的设备,可以被要求将其第一值的散列附加到查找密钥,并且其可以被要求使用与“author”匹配的签名批准进行签名。所有的后续值可以随后被用这种批准签名,或由重新签名者重新签名。

[0121] 假设写入者和重新签名者不允许签名过期,读取条目的设备将能够分辨哪个批准(模式)负责写入。

[0122] 这些条目不能被删除,但是为达到类似的效果,写入者会用使读取者忽略的最小的条目覆盖该值。

[0123] 该选项可以不必限制哪个设备对任何给定查找密钥执行第一次写入。但是它将会有助于确保后续写入相同的查找密钥是由初始作者授权的。

[0124] 可选地,只有初始作者可以被允许解决冲突。系统还可以给予重新签名者那个能

力。

[0125] 可选地，“author”和“re_signer”字段中的任一个或两者可以保存批准模式的列表，而不是单个批准模式。同样可选地，重新签名者可以被允许改变“author”字段，以适应谁可以写入的更改。

[0126] 可选地，“author”和“re_signer”字段可以指代ACL群组名称。

[0127] 如上所述，重新签名者会是同步群组的读/写成员。一些实施例可以允许重新签名者是同步群组的只读成员。如果是这样的话，系统会放松关于成员如何从只读成员接受值的规则。重新签名者随后会只对与先前写入另一个实体的值相同的值重新签名。

[0128] 访问控制条目。

[0129] 第四种选择，经过认证的单写入者条目，是只能由一些同步群组成员写入的值，这些同步群组成员具有对集合的写访问权并支配与条目的查找密钥的批准模式匹配的批准。通过在查找密钥中包含批准模式而不是在查找密钥中包含散列，该技术建立在经过认证的单写入者条目上。

[0130] 在这个选项中，系统可能只接受其查找密钥包含“$writer＝<blessing_pattern>$”的条目(包括原始条目和对现有条目的更新)。系统可以拒绝删除这些条目的现有版本，并且可以只允许其被使用具有以下字段和特性的值来写入：

[0131] (1)re_signer:受信任的重新签名者的批准模式，重新签名者可以为原作者签名。

[0132] (2)sequence:以下之一：

[0133] a.0，如果这是初始值，

[0134] b.大于被覆盖的值中的“sequence”的值，如果签名是由“author”签名，或

[0135] c.是与前述“sequence”相同的值，如果签名是由“re_signer”签名，提供除具有与先前值相同的值的“signature”之外的所有字段。

[0136] (3)signature:除“signature”之外的所有字段的签名，用与<blessing_pattern>或“re_signer”匹配的签名批准进行签名。

[0137] 如果在条目的DAG中发现冲突，系统可以选择具有最高序列号的版本，其他条件相同。在具有相同序列号的两个值之间，系统可以选择由“re_signer”签名的值，而不是由“author”签名的值，其他条件相同。

[0138] 指定的“re_signer”能够使密钥翻转。其可以对由作者签名的值进行重新签名，并定期对它之前签名的值重新签名，以防止签名、批准或密钥中的任何一个过期。任何处理长寿命数据的应用程序都可以使用这种重新签名者，并且该应用程序可以为该角色指定同步群组的服务器成员;应用程序可以为服务器提供重新签名批准，告诉服务器更新其签名的频率，以及可选地还指定应该对哪些前缀进行重新签名，以避免服务器不得不搜索整个集合。

[0139] 如果条目的签名已过期，系统可以拒绝带有已过期或在数据将到达同步群组的其它成员之前将要过期的另外的有效的“signature”字段的Put()调用。当通过同步协议接收值时，系统仍然可以接受其他有效但已过期的“signature”字段。这可能有助于阻止已过期的签名的引入，但如果创建了带有过期签名的条目，则多个同步群组成员可以看到同样过期条目的一致视图。无用集合代码可以包含保存最后未过期的值(如果有一个的话)的指令，来帮助重新签名者修改该情况。冲突解决代码还可以包括规则，该规则越过到期条目优

先选择未过期条目。可选地，如果过期的“author”签名满足指定的标准，重新签名者可以对过期的“author”签名进行重新签名，或者重新签名者可以移除过期的条目或用更早的未过期的条目替换它们。

[0140] 用该选项，想要在给定查找密钥处写入一系列值的设备，将把批准模式放入查找密钥中，并且它将用与该模式匹配的签名批准来对该对象签名。所有的后续值可以由设备(或重新签名者)用这种批准签名。

[0141] 如果写入者和重新签名者不允许签名过期，则对象的读取者将能够分辨哪个批准(模式)对写入负责，并且将知道没有其他成员可能在相关查找密钥处写入值。

[0142] 可选地，这些条目可以不必被删除，但是为达到类似的效果，写入者可以用使读取者忽略的最小的条目覆盖该值。

[0143] 在该选项中，<blessing_pattern>和“re_signer”字段中的任一个或两者可以保存批准模式的列表，而不是单个批准模式。<blessing_pattern>和“re_signer”字段还可以指代ACL群组名称。

[0144] 在该实施例中，只有匹配<blessing_pattern>的成员可被允许解决冲突。可选地，重新签名者也可以被给予这种能力。

[0145] 可选地，“re_signer”模式可以被编码在查找密钥中，而不是在字段中。

[0146] 可选地，DAG无用集合算法可以通过“re_signer”保留由后继签名者通过<blessing_pattern>签名的每个节点。如果完成，这可能有助于(通过编写新数据)验证“re_signer”没有超越它的权力，而作者的密钥不受影响。

[0147] 如描述的，重新签名者会是同步群组的读/写成员。在一些实施例中，重新签名者可以是只读成员，在只读成员中系统会放松关于成员如何从只读成员接受值的规则。这可以通过允许重新签名者只对与其他人写入的先前的值相同的值重新签名来完成。

[0148] 图6描绘了可用于包含或实施程序指令的硬件的框图，例如服务器或其他电子设备。总线600被用作信息高速公路，使其他示出的硬件组件互相连接。处理器(CPU)605是系统的中央处理设备，进行执行程序所需的逻辑操作和计算。单独或与图6中公开的一个或多个其他元件结合的CPU 605，是在本公开内容中使用的处理设备、计算设备或处理器这种术语示例。只读存储器(ROM)610和随机存取存储器(RAM)615构成存储器设备的示例。

[0149] 控制器620与一个或多个可选非暂时性计算机可读存储介质(即，存储器设备625)连接至总线600。这些存储介质可以包括，例如，外部或内部DVD驱动、CD ROM驱动、硬盘驱动、闪存、USB驱动等。如前面表示的，这些各种驱动和控制器都是可选设备。

[0150] 用于提供接口并执行与一个或多个数据集相关的任何查询或分析的程序指令、软件或交互模块可以存储在ROM 610和/或RAM 615中。可选地，程序指令可以被存储在上文讨论的存储介质625上。

[0151] 可选的显示界面630可以允许来自总线600的信息以音频、视觉、图形或字母数字格式显示在显示器635上。使用各种通信元件640，例如通信端口或天线，与诸如打印设备之类的外部设备的通信可以发生。通信元件640可以可通信地连接到通信网络，例如因特网或内联网。

[0152] 硬件还可以包括接口645，其允许从诸如键盘650或其他输入设备655接收数据，其他输入设备655诸如鼠标、触摸板、触摸屏、遥控器、指向设备、视频输入设备和/或音频输入

设备。

[0153] 以上公开的特征和功能，以及替换物，可以被结合成许多其他不同的系统或应用程序。本领域技术人员可以实现的各种目前无法预料的或未预料到的替换、修改、变化或改进，其中的每一个也旨在被本公开的实施例所涵盖。

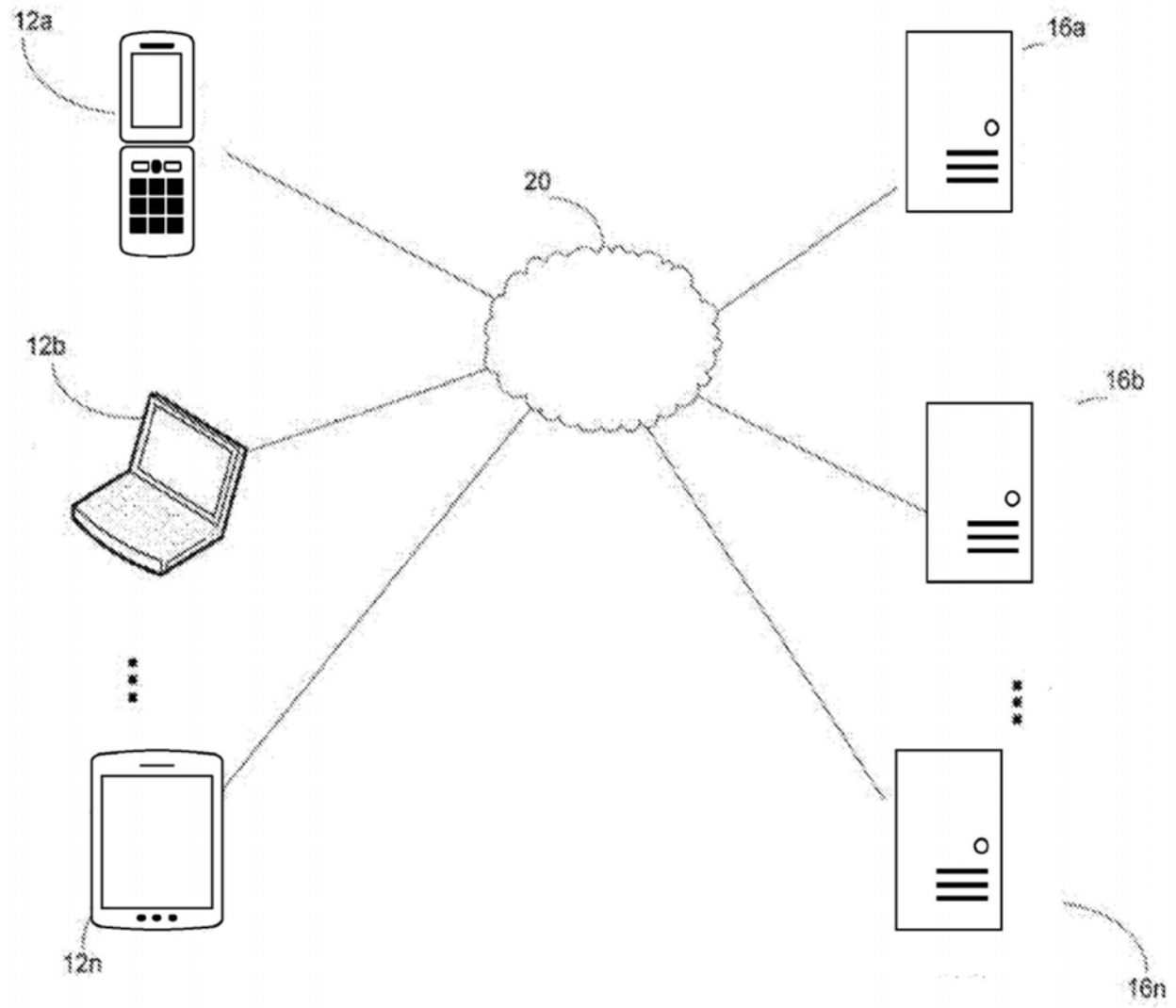

12a
16a
20
12b
16b
12n
16n

图1

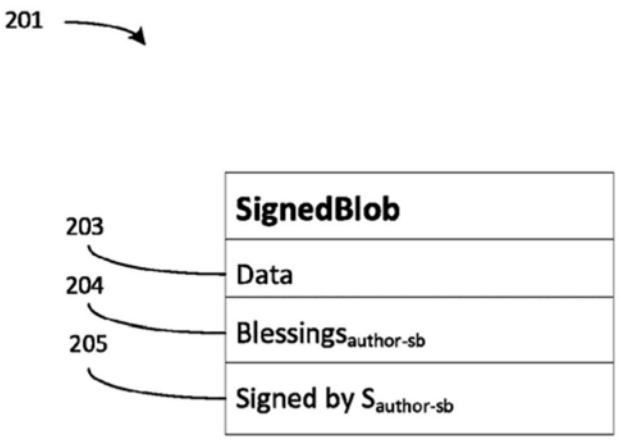

201
SignedBlob
203
Data
204
Blessings author-sb
205
Signed by S author-sb

图2

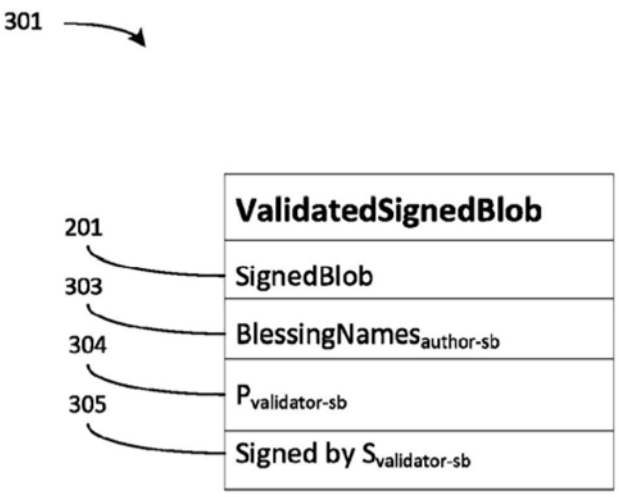

301
ValidatedSignedBlob
201
SignedBlob
303
BlessingNames author-sb
304
P validator-sb
305
Signed by S validator-sb

图3

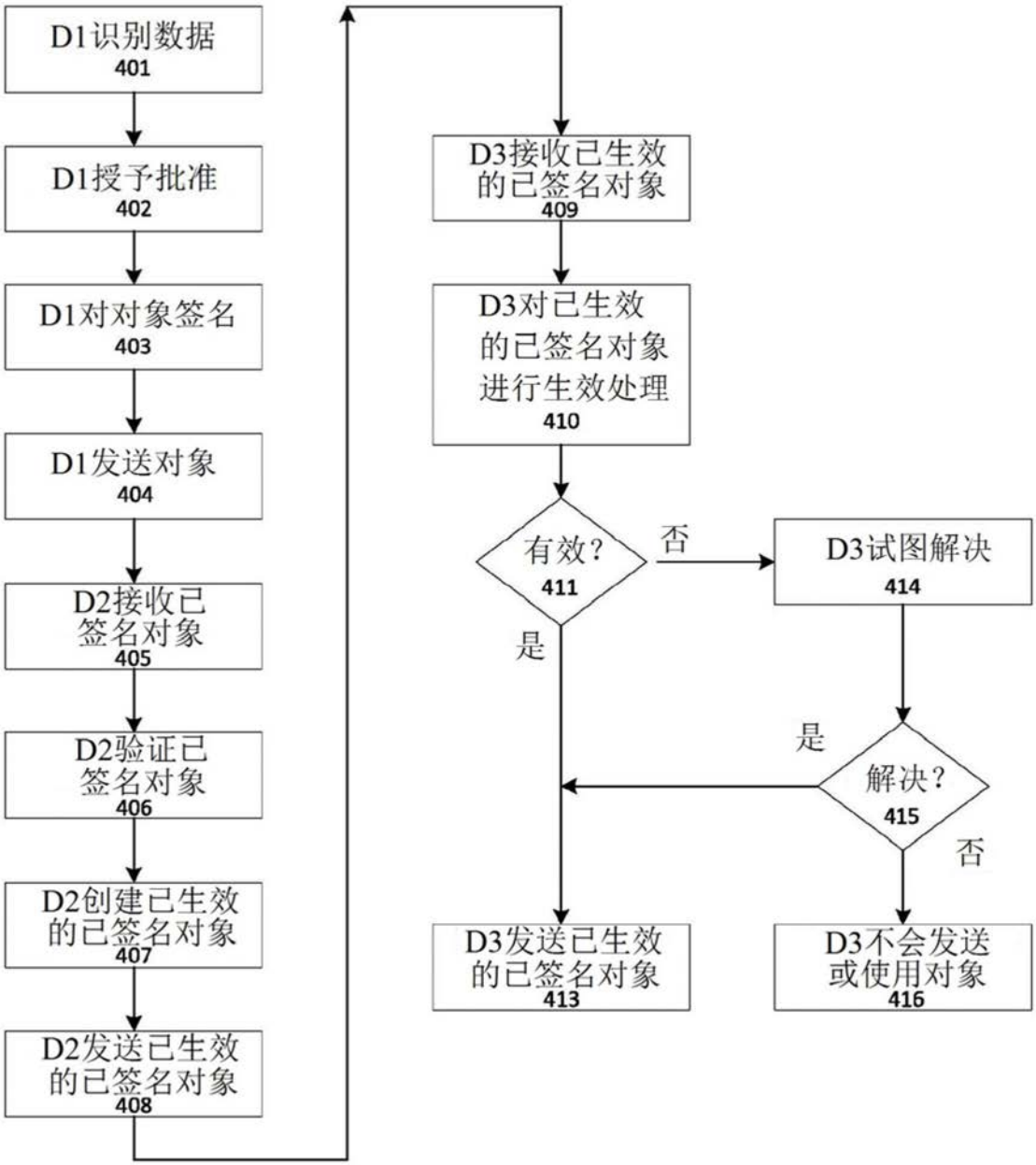

D1识别数据
401
D1授予批准
402
D1对对象签名
403
D1发送对象
404
D2接收已签名对象
405
D2验证已签名对象
406
D2创建已生效的已签名对象
407
D2发送已生效的已签名对象
408
D3接收已生效的已签名对象
409
D3对已生效的已签名对象进行生效处理
410
有效?
411
否
D3试图解决
414
是
是
解决?
415
否
D3发送已生效的已签名对象
413
D3不会发送或使用对象
416

图4

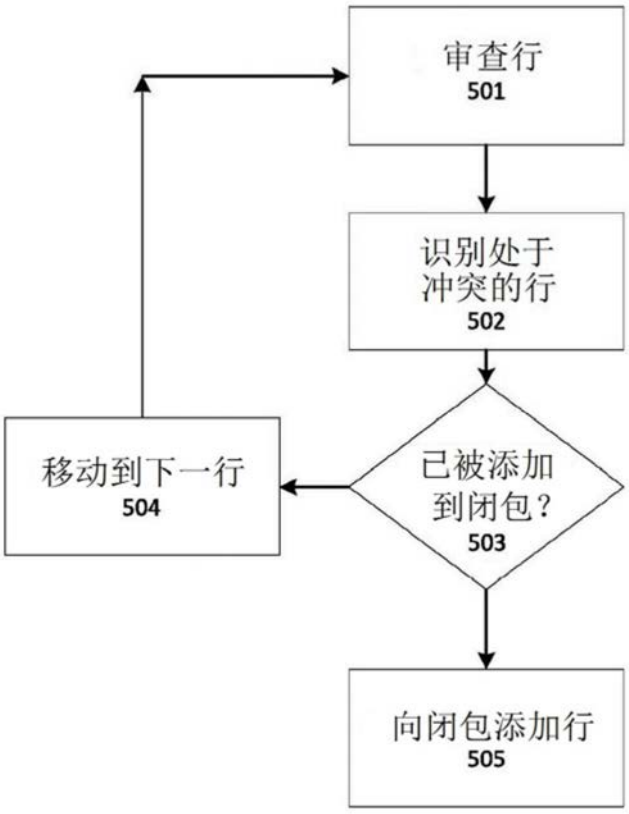



图5

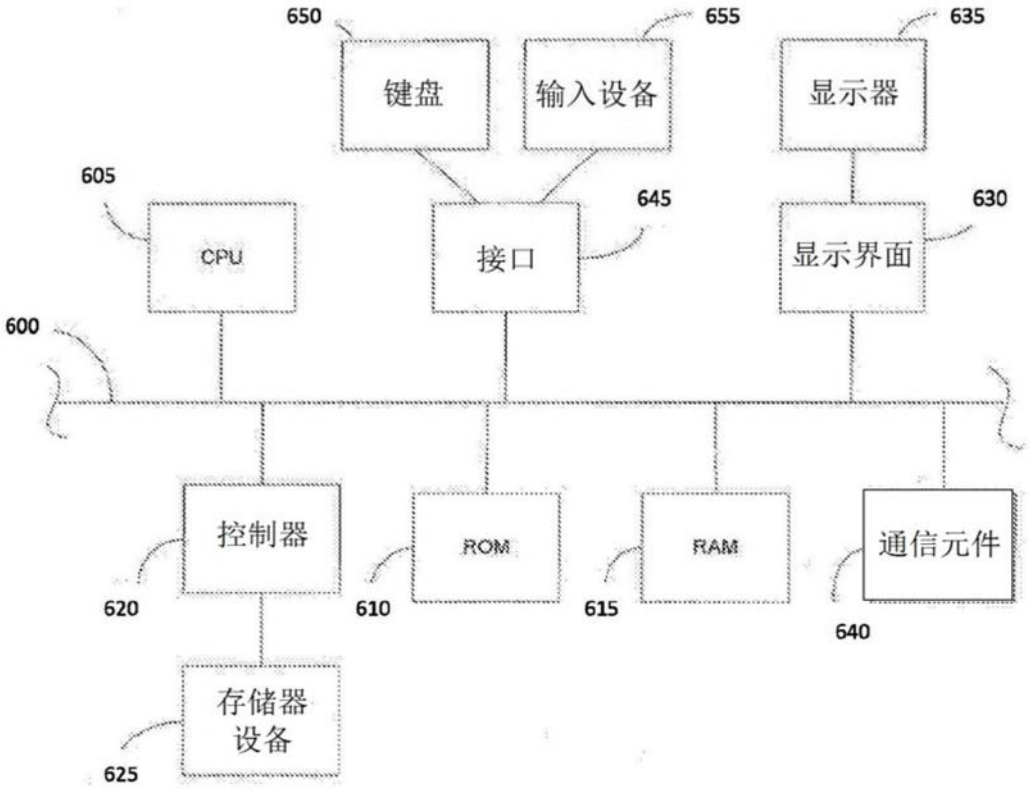

650
键盘
655
输入设备
635
显示器
605
CPU
645
接口
630
显示界面
600
控制器
620
ROM
610
RAM
615
通信元件
640
存储器
设备
625

图6