(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 101924794 A
(43)申请公布日 2010.12.22

(21)申请号 201010259423.7

(22)申请日 2010.08.18

(71)申请人 厦门雅迅网络股份有限公司
地址 361009 福建省厦门市软件产业基地观日路46号

(72)发明人 王松辉 杨一麟 时宜 王国清

(74)专利代理机构 厦门市新华专利商标代理有限公司 35203
代理人 朱凌

(51)Int.Cl.
H04L 29/08(2006.01)
H04L 29/06(2006.01)
H04L 9/30(2006.01)
G06F 21/00(2006.01)

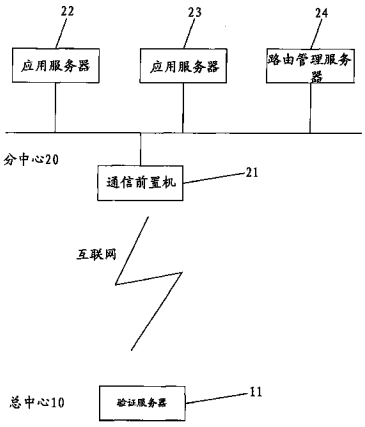
权利要求书 1 页 说明书 4 页 附图 3 页

(54)发明名称

一种基于互联网实时监视软件运行总量的方法

(57)摘要

本发明一种基于互联网实时监视软件运行总量的方法，在分中心的计算机上安装应用程序时生成初始密钥文件，并多处隐藏以不同扩展名存在的密钥文件，接着，应用程序初始启动时进行自身校验，若多处隐藏的密钥文件出现数量短少的情形，或者密钥文件数量正确，但该密钥文件中有关信息不一致，则判断该应用程序已被非法移植；最后，总中心的验证服务器通过互联网定时收集和更新各分中心内各个计算机的本地密钥文件，对各分中心的计算机的运行总量进行监视，判断分中心的各个计算机是否实时在线，从而达致有效控制软件运行总量之目的。



CN 101924794 A

1. 一种基于互联网实时监视软件运行总量的方法，其特征在于包括一总中心和至少一个分中心，该总中心设置一与各分中心实时保持通信联络的验证服务器；

在各分中心的计算机上安装应用程序时，通过安装包里的一个附加执行程序，收集运行该安装包的计算机的机器码，并随机生成一个密钥，安装包利用这个密钥对包括分中心标识、密钥生成时间及计算机的机器码的数据进行加密，再将这些加密后的数据连同密钥一起，采用公钥进行二次加密生成一个密钥文件；最后安装包将生成的密钥文件分别复制存储重命名为多种不同扩展名的文件并设置隐藏属性；

然后，应用程序初始启动时进行自身校验，首先应用程序判断本地隐藏的多处密钥文件是否都存在，若出现文件缺少的情况，则判断应用程序遭到非法移植；如果都存在，则分别采用公钥解密该多个密钥文件，提取该文件中各自的分中心标识、密钥生成时间和机器码的信息并进行比较，如果不一致，则判断应用程序遭到非法移植；同时将本地时间和密钥生成时间进行判断，如果超过预设值，则判断该计算机恶意断开与总中心验证服务器之间的联系，停止运行应用程序；

最后，分中心的各个应用服务器分别定时读取所在计算机上的多个隐藏的密钥文件，并发送给总中心的验证服务器，该验证服务器根据收到的密钥文件提取分中心标识和机器码信息，计算出各分中心运行中的计算机的总量，如果数量大于规定的合法数量值，又或所在的网络地址 IP 不合法，则验证服务器回传一个失败信息给对应的分中心内全部的应用服务器，则该各个应用服务器在数据传输时进行不规律的丢弃数据包；若分中心运行的计算机总量在规定的数值范围内，则验证服务器根据当前时间重新生成密钥文件并发送回对应的分中心内全部的应用服务器，该各个应用服务器分别更新所在计算机的多处隐藏的密钥文件。

2. 根据权利要求 1 所述的一种基于互联网实时监视软件运行总量的方法，其特征在于：所述的多处隐藏的密钥文件为三处，该密钥文件的扩展名分别为 .rom 文件、.dll 文件和 .pnf 文件。

3. 根据权利要求 1 所述的一种基于互联网实时监视软件运行总量的方法，其特征在于：所述的密钥文件分别复制重命名为多种不同扩展名的文件到 System32 目录和 inf 目录下。

4. 根据权利要求 1 所述的一种基于互联网实时监视软件运行总量的方法，其特征在于：所述的机器码包括硬盘序列号、网卡序列号以及 CPU 序列号的信息。

一种基于互联网实时监视软件运行总量的方法

技术领域

[0001] 本发明涉及一种基于互联网实时监视软件运行总量的方法。

背景技术

[0002] 现代服务体系的一个重要特征就是分工合作。随着应用的完善，业务的不断拓展，一个完整的应用服务提供系统会越来越庞大，单靠一个独立的运营服务商是很难做好系统运营的，往往需要跟第三方合作运营方能实现。这样，就会造成应用系统资源（包括数据、应用程序等）的外泄。如果系统没有很好的防拷贝、防盗用技术，就会不可避免地出现在合法范围之外该应用系统资源被随意重复使用的情形，必然会对应用服务提供系统的发布厂商造成一定的经济损失。

发明内容

[0003] 本发明目的在于提供一种基于互联网实时监视软件运行总量的方法，能有效地控制软件的运行总量，以防止出现第三方合作商将应用系统资源外泄以及应用系统资源被随意重复使用的情况。

[0004] 一种基于互联网实时监视软件运行总量的方法，包括一总中心和至少一个分中心，该总中心设置一与各分中心实时保持通信联络的验证服务器；

[0005] 在各分中心的计算机上安装应用程序时，通过安装包里的一个附加执行程序，收集运行该安装包的计算机的机器码，并随机生成一个密钥，安装包利用这个密钥对包括分中心标识、密钥生成时间及计算机的机器码的数据进行加密，再将这些加密后的数据连同密钥一起，采用公钥进行二次加密生成一个密钥文件；最后安装包将生成的密钥文件分别复制存储重命名为多种不同扩展名的文件并设置隐藏属性；

[0006] 然后，应用程序初始启动时进行自身校验，首先应用程序判断本地隐藏的多处密钥文件是否都存在，若出现文件缺少的情况，则判断应用程序遭到非法移植；如果都存在，则分别采用公钥解密该多个密钥文件，提取该文件中各自的分中心标识、密钥生成时间和机器码的信息并进行比较，如果不一致，则判断应用程序遭到非法移植；同时将本地时间和密钥生成时间进行判断，如果超过预设值，则判断该计算机恶意断开与总中心验证服务器之间的联系，停止运行应用程序；

[0007] 最后，分中心的各个应用服务器分别定时读取所在计算机上的多个隐藏的密钥文件，并发送给总中心的验证服务器，该验证服务器根据收到的密钥文件提取分中心标识和机器码信息，计算出各分中心运行中的计算机的总量，如果数量大于规定的合法数量值，又或所在的网络地址 IP 不合法，则验证服务器回传一个失败信息给对应的分中心内全部的应用服务器，则该各个应用服务器在数据传输时进行不规律的丢弃数据包；若分中心运行的计算机总量在规定的数值范围内，则验证服务器根据当前时间重新生成密钥文件并发送回对应的分中心内全部的应用服务器，该各个应用服务器分别更新所在计算机的多处隐藏的密钥文件。

[0008] 所述的多处隐藏的密钥文件为三处，该密钥文件的扩展名分别为 .rom 文件、.dll 文件和 .pnf 文件。

[0009] 所述的密钥文件分别复制重命名为多种不同扩展名的文件到 System32 目录和 inf 目录下。

[0010] 所述的机器码包括硬盘序列号、网卡序列号以及 CPU 序列号的信息。

[0011] 首先，本发明在分中心的计算机上安装应用程序时生成初始密钥文件，并多处隐藏以不同扩展名存在的密钥文件，所述的密钥文件包含分中心标识、密钥生成时间和安装该应用程序的计算机的机器码信息；接着，应用程序初始启动时进行自身校验，该多处隐藏的密钥文件若出现数量短少的情形，则判断该应用程序已被非法移植；若该多处隐藏的密钥文件数量正确，则解密该多个密钥文件，分别提取和比较该密钥文件中有关分中心标识、密钥生成时间和安装该应用程序的计算机的机器码的信息，若信息不一致，则判断该应用程序已被非法移植；最后是总中心的验证服务器通过互联网对各分中心的计算机的运行总量进行监视，实现软件运行总量的控制，该总中心的验证服务器定时收集和更新各分中心内各个计算机的本地密钥文件，通过提取密钥文件中有关分中心标识、密钥生成时间和安装该应用程序的计算机的机器码信息，计算出各分中心运行中的计算机总量，以及将本地时间与密钥生成时间进行比较，判断分中心的各个计算机是否实时在线，从而达致有效控制软件运行总量之目的。

附图说明

[0012] 图 1 为本发明中应用服务提供系统的架构示意图；

[0013] 图 2 为本发明中密钥文件的格式表；

[0014] 图 3 为本发明中应用程序安装流程示意图。

[0015] 以下结合附图和具体实施例对本发明作进一步详述。

具体实施方式

[0016] 如图 1 所示，为一种基于互联网的应用服务提供系统，该系统在本实施例中至少由两个中心构成，即总中心 10 和分中心 20，其中总中心 10 代表了应用服务提供商和开发商，主要包括一验证服务器 11；该分中心 20 代表了第三方合作商运营的系统，分中心 20 包括提供服务的应用服务器 22、23 以及系统平台软件设备路由管理服务器 24，还有一负责和外部通信用的通信前置机 21，该总中心 10 和分中心 20 拥有各自的局域网。所述的分中心 20 可以一个及一个以上，该应用服务器数量可以根据需要设置，不局限两个。每个分中心 20 都会有一个通信前置机 21，该通信前置机 21 在启动时，会登录部署在远程总中心 10 的授权验证服务器 11，并且一直与验证服务器 11 保持顺畅通信。

[0017] 本发明一种基于互联网实时监视软件运行总量的方法，首先，在分中心 20 的计算机上安装应用程序，安装流程如图 3 所示。

[0018] 步骤 101，安装应用程序的时候，通过安装包里的一个附加执行程序，收集运行该安装包的计算机的机器码，该机器码至少包括硬盘序列号、网卡序列号以及 CPU 序列号等信息，并随机生成一个密钥，安装包利用这个密钥对包括分中心标识、密钥生成时间及计算机的机器码等数据进行加密，再将这些加密后的数据连同密钥一起，采用公钥进行二次加

密生成一个密钥文件。

[0019] 本实施例中的密钥文件格式如图 2 所示，该密钥文件格式中包括分中心标识（该分中心标识为唯一标识）、密钥生成时间、安装有该软件的计算机的机器码（包含硬盘序列号、网卡序列号以及 CPU 序列号）。

[0020] 步骤 102，安装包在安装后期，将步骤 101 生成的密钥文件分别复制重命名为三种不同格式的文件，分别为 :. rom 文件、. dll 文件和 . pnf 文件。

[0021] 步骤 103，将步骤 102 生成的三个不同格式的文件，分别复制到 System32 目录和 inf 目录下，并设置隐藏属性。

[0022] 然后，应用程序初始启动时进行自身校验，以防出现应用程序被非法移植的情形：

[0023] 应用程序初始启动后，判断三处隐藏的密钥文件是否都存在，如果出现文件缺少的情况，则认为应用程序遭到非法移植，要求重新安装程序；如果都存在，则分别采用公钥解密三个密钥文件，取出三个文件中各自的中心标识、生成时间和机器码进行比较，如果不一致，则认为应用程序被非法移植，要求重新安装程序；同时将本地时间和密钥生成时间进行判断，如果超过预设值，表示该计算机恶意断开与总中心验证服务器 11 之间的联系，则停止运行应用程序。

[0024] 最后，总中心 10 的验证服务器 11 通过互联网对分中心 20 计算机的运行总量进行实时监视，进而控制软件的运行总量，具体包括以下几个步骤：

[0025] 步骤 201、分中心 20 的应用服务器 22、23 分别定时读取所在计算机上的三个隐藏的密钥文件，并发送给路由管理服务器 24；

[0026] 步骤 202、该路由管理服务器 24 每隔一定时间搜集所述的三种隐藏的密钥文件发送给通信前置机 21，该通信前置机 21 将收到的信息发往验证服务器 11；

[0027] 步骤 203、该验证服务器 11 根据收到密钥文件，提取有关分中心标识、密钥生成时间和安装有应用程序的计算机的机器码等信息，计算出分中心 20 内运行中的计算机的总量，如果数量大于规定的合法数量值，又或所在的网络地址 IP 不合法，则执行步骤 205，否则执行步骤 204；

[0028] 步骤 204、该验证服务器 11 根据当前时间重新生成密钥文件并发送给通信前置机 21，经由该通信前置机 21 转发给路由管理服务器 24 后，再由该路由管理服务器 24 转发给分中心 20 内全部的应用服务器 22、23，该各个应用服务器分别负责更新所在计算机的三处密钥文件；

[0029] 步骤 205、该验证服务器 11 回传一个失败信息给通信前置机 21，经路由管理服务器 24 转发给对应分中心 20 内全部的应用服务器 22、23，该各个应用服务器 22、23 在数据传输时进行不规律的丢弃数据包。

[0030] 本发明的重点在于：首先，在分中心的计算机上安装应用程序时生成初始密钥文件，并多处隐藏以不同扩展名存在的密钥文件，所述的密钥文件包含分中心标识、密钥生成时间和安装该应用程序的计算机的机器码信息；接着，应用程序初始启动时进行自身校验，若多处隐藏的密钥文件出现数量短少的情形，则判断该应用程序已被非法移植；若该多处隐藏的密钥文件数量正确，则解密该多个密钥文件，分别提取和比较该密钥文件中有关分中心标识、密钥生成时间和安装该应用程序的计算机的机器码的信息，若信息不一致，则判

断该应用程序已被非法移植；最后是总中心的验证服务器通过互联网对各分中心的计算机的运行总量进行监视，实现软件运行总量的控制，该总中心的验证服务器定时收集和更新各分中心内各个计算机的本地密钥文件，通过提取密钥文件中有关分中心标识、密钥生成时间和安装该应用程序的计算机的机器码信息，计算出各分中心运行中的计算机总量，以及将本地时间与密钥生成时间进行比较，判断分中心的各个计算机是否实时在线，从而达致有效控制软件运行总量之目的。

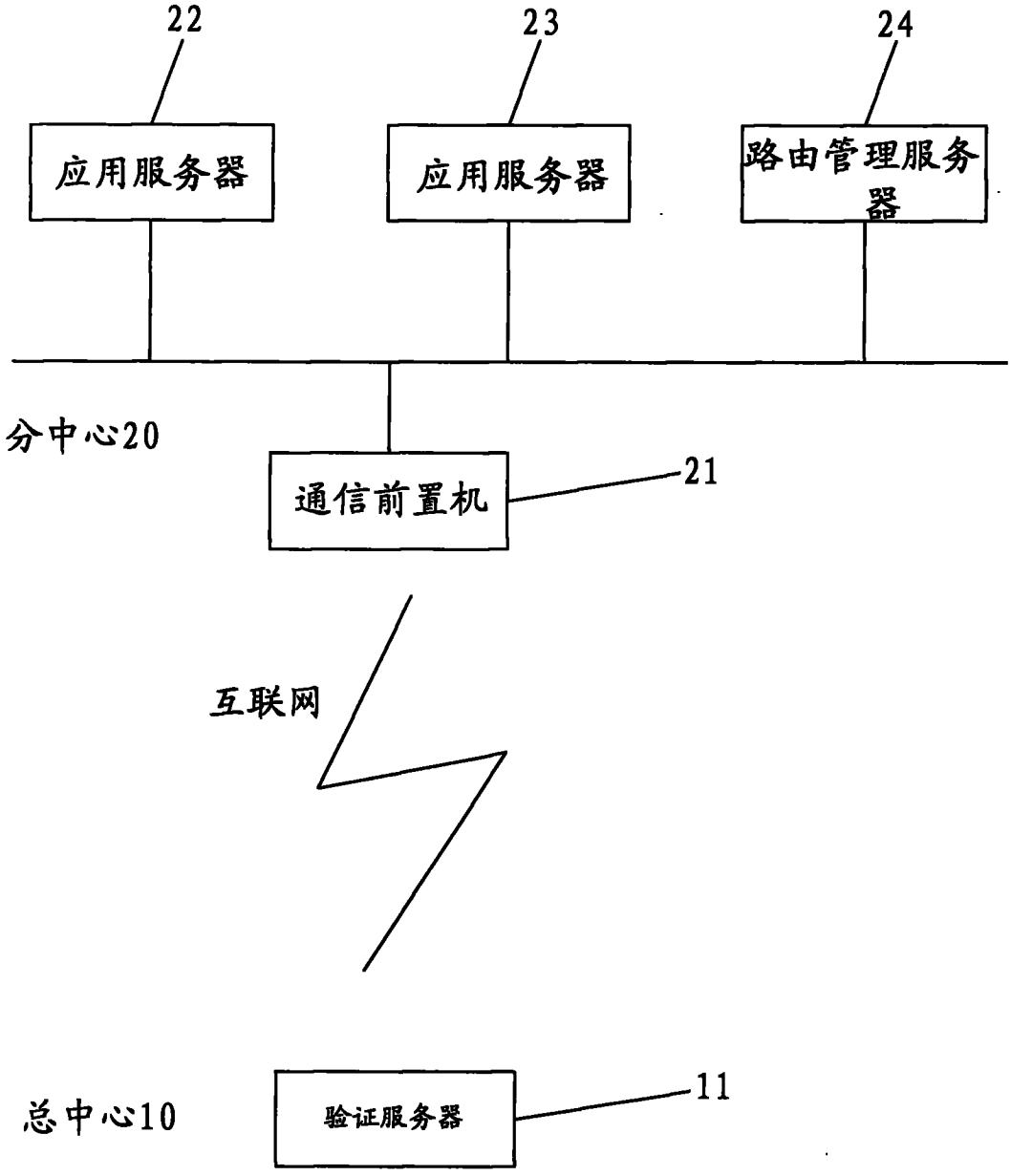

22
23
24
应用服务器
应用服务器
路由管理服务器
分中心20
通信前置机
21
互联网
总中心10
验证服务器
11

图 1

字段描述	长度	含义
密钥类型	1	决定何种加密方式
密钥	128	一个随机的16字节或者8字节字符
中心标识	4	用于匹配分中心的信息
花码	7	没有意义的字节
密钥生成年	1	密钥生成年
花码	13	没有意义的字节
密钥生成月	1	密钥生成月
花码	5	没有意义的字节
密钥生成日	1	密钥生成日
花码	12	没有意义的字节
机器码是否存在	1	机器码是否存在
机器码或者花码	48	当有机器码时：包含硬盘ID、网卡序列号以及CPU序列号

图 2

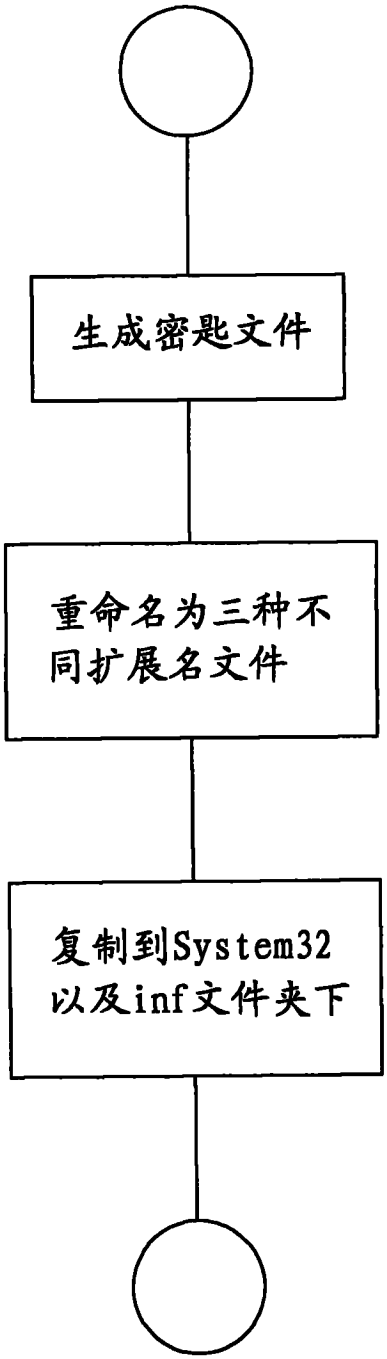


图 3