(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 101855860 A
(43)申请公布日 2010.10.06

(21)申请号 200880115595.8

(22)申请日 2008.09.12

(30)优先权数据
60/993,756 2007.09.14 US

(85)PCT申请进入国家阶段日
2010.05.11

(86)PCT申请的申请数据
PCT/US2008/010677 2008.09.12

(87)PCT申请的公布数据
WO2009/035674 EN 2009.03.19

(71)申请人 安全第一公司
地址 美国加利福尼亚

(72)发明人 R·L·奥尔西尼 M·S·奥黑尔
R·达文波特

(74)专利代理机构 中国国际贸易促进委员会专利商标事务所 11038
代理人 马浩

(51)Int.Cl.
*H04L 9/00*(2006.01)

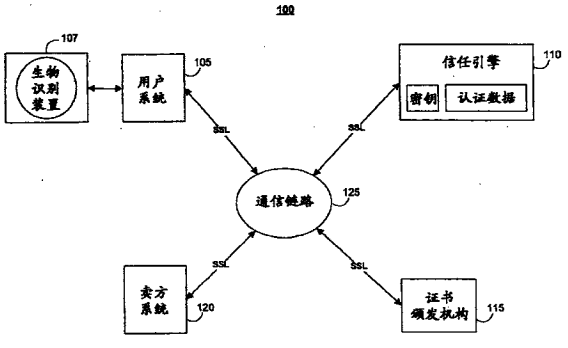
权利要求书 1 页 说明书 64 页 附图 41 页

(54)发明名称

用于管理加密密钥的系统和方法

(57)摘要

提供了一种用于管理加密密钥的公共接口。管理加密密钥的请求可以以第一接口格式被接收，被转换为公共接口格式，然后远离第一接口被执行。返回参数然后可以从公共接口格式转换为与第一接口兼容的格式并被安全地传送至第一接口。加密密钥可以与安全数据解析器联合使用，其中安全数据解析器通过将数据集中的数据随机分配成两个或更多个份来保护数据。




CN 101855860 A

1. 一种用于管理加密密钥的方法，所述方法包括：

从第一接口接收管理远离所述第一接口存储的至少一个加密密钥的请求；

将所述请求从第一接口格式转换为公共接口格式；

通过至少校验所述请求是从授权源发起的来认证所述请求；以及

执行转换后的具有所述公共接口格式的请求。

2. 根据权利要求 1 所述的方法，其中，执行转换后的请求包括取回所述至少一个加密密钥。

3. 根据权利要求 1 所述的方法，其中，执行转换后的请求包括生成所述至少一个加密密钥。

4. 根据权利要求 1 所述的方法，其中，执行转换后的请求包括删除所述至少一个加密密钥。

5. 根据权利要求 1 所述的方法，其中，执行转换后的请求包括在密钥存储器中存储所述至少一个加密密钥。

6. 根据权利要求 1 所述的方法，其中，执行转换后的请求包括在可移动介质上存储所述至少一个加密密钥。

7. 根据权利要求 1 所述的方法，进一步包括使用所述至少一个加密密钥来保护数据集，其中保护所述数据集包括：

使用所述至少一个加密密钥来加密所述数据集；

生成随机或伪随机值；

至少部分基于所述随机或伪随机值，将所述数据集中的加密数据分配成两份或更多份；以及

将所述两份或更多份分开存储在至少一个数据仓库中。

8. 根据权利要求 7 所述的方法，其中，将所述两份或更多份分开存储在至少一个数据仓库中包括将所述两份或更多份存储在至少两个地理上分开的数据仓库中。

9. 根据权利要求 1 所述的方法，进一步包括将所执行的请求的至少一个返回参数从公共接口格式转换为第一接口格式。

10. 根据权利要求 9 所述的方法，其中，所执行的请求的所述至少一个返回参数包括至少一个加密密钥。

11. 根据权利要求 9 所述的方法，进一步包括通过安全通信通道将所述至少一个返回参数传输至所述第一接口。

12. 根据权利要求 1 所述的方法，其中，认证所述请求包括执行认证协议或加密握手。

13. 根据权利要求 1 所述的方法，其中，认证所述请求包括校验与所述请求相关联的加密签名。

14. 根据权利要求 1 所述的方法，其中，认证所述请求包括验证认证令牌。

15. 根据权利要求 14 所述的方法，其中，验证认证令牌包括实施与所述认证令牌相关联的截止日期或截止时间。

# 用于管理加密密钥的系统和方法

[0001] 相关申请的交叉引用

[0002] 本申请要求 2007 年 9 月 14 日提交的美国临时申请 No.60/993,756 的优先权，其通过引用全部结合于此。

## 技术领域

[0003] 本发明一般地涉及用于保护数据以免于未授权访问或使用的系统。更具体地，本发明涉及用于支持加密密钥的公共接口。

## 背景技术

[0004] 在当今社会，个人和企业在计算机系统上和通过计算机系统执行不断增加的活动量。这些计算机系统，包括专属的和非专属的计算机网络，通常存储、存档和传输各种类型的敏感信息。因此，存在不断增加的用于确保通过这些系统存储和传输的数据不被读取或泄露 (compromise) 的需求。

[0005] 用于保护计算机系统的一种通常解决方案是提供登录和口令功能。然而，口令管理已经被证明是花费非常大的，大量的求助台呼叫都与口令问题有关。此外，口令提供了很少的安全性，这是因为它们通常被存储在易受到通过例如暴力攻击而进行的不适当访问的文件中。

[0006] 用于保护计算机系统的另一方案是提供加密基础结构。密码术通常指的是通过将数据变换或加密成为不可读格式来保护数据。只有那些拥有加密密钥的人才能够将数据解密成可用格式。密码术被用来识别用户（例如认证），以允许访问特权（例如授权）从而创建数字证书和签名等。一种普遍的密码术系统是使用两个密钥的公钥系统，公钥被每个人知道，而私钥仅被其个人或企业所有者知道。通常，使用一个密钥加密的数据用另一密钥解密，并且任一密钥都不可由另一密钥重建。

[0007] 遗憾的是，即使上述典型的公钥加密系统在安全上仍然高度依赖用户。例如，加密系统例如通过用户浏览器向用户发布私钥。没有经验的用户然后通常将该私钥存储在可被其他人通过开放的计算机系统（例如，因特网）访问的硬盘驱动器上。另一方面，用户可能为包含其私钥的文件选择较差的名字，例如，“密钥”。上述和其他动作的结果是使得密钥或多个密钥易被泄露。

[0008] 除了上述泄露，用户还可能将他或她的私钥保存在配置有存档或备份系统的计算机系统上，潜在地导致私钥的副本经过多个计算机存储装置或其他系统。这种安全漏洞通常被称作“密钥迁移 (keymigration)”。类似于密钥迁移，许多应用最多通过简单的登录和口令访问提供对用户的私钥的访问。如上所述，登录和口令访问往往无法提供充分的安全性。

[0009] 用于增加上述加密系统的安全性的一个解决方案是将生物识别 (biometrics) 包括作为认证或授权的一部分。生物识别通常包括可测量的身体特征，诸如能够由自动化系统通过例如图案匹配或者指纹图案或语音图案识别而检查的指纹或语音。在这样的系统

中，用户的生物识别信息和 / 或密钥可以被存储在移动计算装置（例如，智能卡、膝上型电脑、个人数字助理、或移动电话）上，从而使得生物识别信息或密钥能够在移动环境下使用。

[0010] 上述的移动生物识别加密系统仍然遭受各种缺点。例如，移动用户可能丢失或损坏智能卡或便携式计算装置，从而使他或她对可能重要的数据的访问被完全切断。或者，恶意的人可能偷取移动用户的智能卡或便携式计算装置并使用它来有效地偷取移动用户的数字证书。另一方面，便携式计算装置可以连接到诸如因特网之类的开放系统，并且，类似于口令，存储生物识别信息的文件可能由于用户对于安全性的疏忽或恶意入侵者而易被泄露。

[0011] 此外，有许多方式来安全地创建、存储和管理个人加密密钥。例如，一些应用可以将用户的加密密钥存储在密钥存储器或其他数据结构中。在用户的密钥存储器中的加密密钥可以被多种应用访问。然而一些应用可能与其他应用不兼容，或可能损害用户的加密密钥的安全性，例如，将一个或多个密钥暴露至讹误或未授权或不安全的访问。

## 发明内容

[0012] 因此，提供一种密码系统，其安全性是用户无关的，同时仍支持移动用户。

[0013] 此外，还提供了一种公共接口，例如应用程序接口 (API)，其能够支持对多个加密密钥提供者的多个接口，并将从这些密钥提供者获取的加密密钥提交给安全解析器引擎 (secure parser engine)，该安全解析器引擎用于例如保护用于存储或传输的数据。这样的安全解析器引擎在 Orsini 等人的美国专利 No. 7,391,865、2005 年 10 月 25 日提交的美国专利申请 No. 11/258,839 以及 2006 年 11 月 20 日提交的美国专利申请 No. 11/602,667 中有更详细的描述，所有这些都通过引用全文结合于此。

[0014] 因此，本发明的一个方面是提供一种用于保护实际上任何类型的数据免于未授权访问或使用的方法。该方法包括将要保护的数据解析、拆分和 / 或分离成为两个或更多个部或部分的一个或多个步骤。该方法还包括加密要保护的数据。数据的加密可以在数据的第一次解析、拆分和 / 或分离之前或之后执行。此外，对于数据的一个或多个部分可以重复加密步骤。类似地，对于数据的一个或多个部分可以重复解析、拆分和 / 或分离步骤。该方法还可选地包括存储已经在一个位置或在多个位置处加密的解析、拆分和 / 或分离的数据。该方法还可选地包括将被保护的数据重建或重新组装成其原始形式以供授权的访问或使用。该方法可以结合到任何能够执行该方法的期望步骤的计算机、服务器、引擎等的操作中。

[0015] 本发明的另一方面提供了一种用于实际保护任何类型的数据免于未授权访问或使用的系统。该系统包括数据拆分模块、加密处理模块以及可选的数据组装模块。在一个实施例中，该系统还包括一个或多个数据存储设备，其中可以存储安全数据。

[0016] 因此，本发明的一个方面是提供安全服务器或信任引擎，其具有服务器中心 (server-centric) 密钥，或换句话说，在服务器上存储加密密钥和用户认证数据。根据该实施例，用户访问信任引擎以执行认证和加密功能，所述功能诸如但不限于，认证，授权，数字签名，生成、存储和检索证书，加密，类似公证和类似委托书的动作，等等。

[0017] 本发明的另一方面是提供一种可靠的或可信任的认证处理。而且，在可信任的肯

定认证之后，可以采取大量的不同动作，从提供加密技术，到系统或装置授权和访问，到允许使用或控制一个或大量电子装置。

[0018] 本发明的另一方面是在加密密钥和认证数据不被丢失、偷取或泄露的环境中提供加密密钥和认证数据，从而有利地避免对连续重新发布和管理新密钥和认证数据的需求。根据本发明的另一方面，信任引擎允许用户为多个活动、供应商和 / 或认证请求使用一个密钥对。根据本发明的另一方面，信任引擎在服务器端执行加密处理的至少一个步骤，例如但不限于，加密、认证、或签名，从而允许客户或用户仅拥有很少的计算资源。

[0019] 根据本发明的另一方面，信任引擎包括一个或多个仓库，用于存储每个加密密钥和认证数据的各个部分。这些部分是通过数据拆分处理来创建的，禁止在没有来自一个仓库中多于一个位置或来自多个仓库的预定部分的情况下重建。根据另一实施例，多个仓库在地理上可以是远离的，从而在一个仓库处的不良员工或被泄密的系统不会提供对用户密钥或认证数据的访问。

[0020] 根据另一实施例，认证处理有利地允许信任引擎并行处理多个认证活动。根据另一实施例，信任引擎可以有利地跟踪失败的访问尝试，并由此限制恶意入侵者可能尝试破坏系统的次数。

[0021] 根据本发明的另一实施例，信任引擎可以包括多个实例，其中每个信任引擎可以预测并与其他信任引擎共享处理负荷。根据另一实施例，信任引擎可以包括用于轮询 (poll) 多个认证结果以确保多于一个系统认证了用户的冗余模块。

[0022] 因此，本发明的一个方面包括安全加密系统，其可以被远程访问，用于存储任何类型的数据，包括但不限于与多个用户相关联的多个私有加密密钥。该加密系统将多个用户中的每个用户与多个私有加密密钥中的一个或多个不同密钥相关联，并使用相关联的一个或多个不同密钥为每个用户执行加密功能而不释放 (release) 所述多个私有加密密钥给用户。该加密系统包括仓库系统，其具有存储要保护的数据（诸如多个私有加密密钥和多个注册认证数据）的至少一个服务器。每个注册认证数据识别多个用户之一，并且多个用户中的每个用户与多个私有加密密钥中的一个或多个不同密钥相关联。该加密系统还可以包括认证引擎，其将由多个用户中的一个用户接收的认证数据与对应于该用户并从仓库系统接收的注册认证数据进行比较，从而产生认证结果。该加密系统还可以包括加密引擎，其在认证结果表明正确识别了该多个用户中的该用户时，以该多个用户中的该用户的名义使用从仓库系统接收的相关联的一个或多个不同密钥执行加密功能。该加密系统还可以包括交易引擎，连接以将数据从多个用户路由至仓库服务器系统、认证引擎、以及加密引擎。

[0023] 本发明的另一方面包括安全加密系统，其可选地可远程访问。该加密系统包括仓库系统，其具有存储至少一个私钥和任何其他数据的至少一个服务器，其中其他数据例如但不限于多个注册认证数据，每个注册认证数据识别可能的多个用户之一。该加密系统还可以可选地包括认证引擎，其将用户接收到的认证数据与对应于该用户并从仓库系统接收的注册认证数据进行比较，从而产生认证结果。该加密系统还包括加密引擎，其在认证结果表明正确识别了用户时，以该用户的名义至少使用所述私钥来执行加密功能，所述私钥可以从仓库系统接收。该加密系统还可以可选地包括交易引擎，其连接以将数据从用户路由至其他引擎或系统，例如但不限于，仓库服务器系统、认证引擎、以及加密引擎。

[0024] 本发明的另一方面包括一种有助于加密功能的方法。该方法包括将多个用户中的

一个用户与存储在安全位置（诸如安全服务器）的多个私有加密密钥中的一个或多个密钥相关联。该方法还包括接收来自用户的认证数据，以及将该认证数据与对应于该用户的认证数据进行比较，从而校验该用户的身份。该方法还包括使用一个或多个密钥来执行加密功能而不向该用户释放该一个或多个密钥。

[0025] 本发明的另一方面包括认证系统，用于通过安全存储用户的注册认证数据来唯一识别用户。该认证系统包括一个或多个数据存储设备，其中每个数据存储设备包括计算机可存取存储介质，其存储注册认证数据的至少一个部分。该认证系统还包括与一个或多个数据存储设备通信的认证引擎。该认证引擎包括：数据拆分模块，其对注册认证数据进行操作以创建多个部分；数据组装模块，其处理来自至少一个数据存储设备的部分以组装注册认证数据；以及数据比较器模块，其从用户接收当前认证数据，并将该当前认证数据与所组装的注册认证数据进行比较以确定该用户是否已经被唯一识别。

[0026] 本发明的另一方面包括加密系统。该加密系统包括一个或多个数据存储设备，其中每个数据存储设备包括计算机可存取存储介质，其存储一个或多个加密密钥的至少一部分。该加密系统还包括加密引擎，其与数据存储设备通信。加密引擎还包括：数据拆分模块，其对加密密钥进行操作以创建多个部分；数据组装模块，其处理来自至少一个数据存储设备的部分以组装加密密钥；以及加密处理模块，其接收所组装的加密密钥并利用其执行加密功能。

[0027] 本发明的另一方面包括一种在地理上远程的安全数据存储设备中存储任何类型的数据（包括但不限于认证数据）的方法，从而保护数据不会由任何单独的数据存储设备合成。该方法包括在信任引擎接收数据，在信任引擎将数据与第一基本上随机的值结合以形成第一结合值，以及将数据与第二基本上随机的值结合以形成第二结合值。该方法包括创建第一基本上随机的值与第二结合值的第一配对，创建第一基本上随机的值与第二基本上随机的值的第二配对，以及将第一配对存储在第一安全数据存储设备中。该方法包括将第二配对存储在远离第一安全数据存储设备的第二安全数据存储设备中。

[0028] 本发明的另一方面包括一种存储任何类型的数据（包括但不限于认证数据）的方法，包括：接收数据，将该数据与第一位组（setof bits）结合以形成第二位组，以及将该数据与第三位组结合以形成第四位组。该方法还包括创建第一位组与第三位组的第一配对。该方法还包括创建第一位组与第四位组的第二配对，以及将第一配对和第二配对中的一个存储在第一计算机可存取存储介质中。该方法还包括将第一配对和第二配对中的另一个存储在第二计算机可存取存储介质中。

[0029] 本发明的另一方面包括一种将加密数据存储在地理上远离的安全数据存储设备中的方法，从而保护加密数据不会由任何单独的数据存储设备组成。该方法包括在信任引擎接收加密数据，在信任引擎将加密数据与第一基本上随机的值结合以形成第一结合值，以及将加密数据与第二基本上随机的值结合以形成第二结合值。该方法还包括创建第一基本上随机的值与第二结合值的第一配对，创建第一基本上随机的值与第二基本上随机的值的第二配对，以及将第一配对存储在第一安全数据存储设备中。该方法还包括将第二配对存储在远离第一安全数据存储设备的第二安全数据存储设备中。

[0030] 本发明的另一方面包括一种存储加密数据的方法，包括接收认证数据以及将加密数据与第一位组结合以形成第二位组。该方法还包括将加密数据与第三位组结合以形成第

四位组，创建第一位组和第三位组的第一配对，以及创建第一位组和第四位组的第二配对。该方法还包括在第一计算机可存取存储介质中存储第一配对和第二配对中的一个，以及在第二计算机可存取存储介质中存储第一配对和第二配对中的另一个。

[0031] 本发明的另一方面包括一种在加密系统中处理任何类型或形式的敏感数据的方法，其中敏感数据仅在授权用户使用该敏感数据的动作期间以可用形式存在。该方法还包括在软件模块中从第一计算机可存取存储介质接收基本上随机化的或加密的敏感数据，以及在该软件模块中从一个或多个其他计算机可存取存储介质接收可能是或不是敏感数据的基本上随机化的或加密的数据。该方法还包括在软件模块中处理基本上随机化的预加密敏感数据和可能是或不是敏感数据的基本上随机化的或加密的数据以组装敏感数据，以及在软件引擎中使用敏感数据执行动作。所述动作包括但不限于认证用户和执行加密功能之一。

[0032] 本发明的另一方面包括安全认证系统。该安全认证系统包括多个认证引擎。每个认证引擎接收被设计为以某个确信度唯一识别用户的注册认证数据。每个认证引擎接收当前认证数据以与注册认证数据相比较，并且每个认证引擎确定认证结果。该安全认证系统还包括冗余系统，其接收至少两个认证引擎的认证结果并确定用户是否已被唯一识别。

[0033] 本发明的另一方面包括在移动系统中的安全数据，借由该移动系统，数据能够在根据本发明被保护的不同部分中传输，从而被泄露的任何一个部分都不会提供足够数据来恢复原始数据。这可以应用于任何数据传输，无论其是有线、无线或物理的。

[0034] 本发明的另一方面包括将本发明的安全数据解析器集成到存储或传输数据的任何适当系统中。例如，电子邮件系统、RAID 系统、视频广播系统、数据库系统、或任何其他适当系统可以具有以任何适当等级集成的安全数据解析器。

[0035] 本发明的另一方面包括使用任何适当的解析和拆分算法来生成数据份 (share)。随机、伪随机、确定的或其任意组合都可以被采用来解析和拆分数据。

## 附图说明

[0036] 下面结合附图具体描述本发明，附图用于示出本发明，但不用于限制本发明，其中：

[0037] 图 1 示出了根据本发明的实施例的多个方面的加密系统的框图；

[0038] 图 2 示出了根据本发明的实施例的多个方面，图 1 的信任引擎的框图；

[0039] 图 3 示出了根据本发明的实施例的多个方面，图 2 的交易引擎的框图；

[0040] 图 4 示出了根据本发明的实施例的多个方面，图 2 的仓库的框图；

[0041] 图 5 示出了根据本发明的实施例的多个方面，图 2 的认证引擎的框图；

[0042] 图 6 示出了根据本发明的实施例的多个方面，图 2 的加密引擎的框图；

[0043] 图 7 示出了根据本发明的另一实施例的多个方面的仓库系统的框图；

[0044] 图 8 示出了根据本发明的实施例的多个方面的数据拆分处理的流程图；

[0045] 图 9，面 A 示出了根据本发明的实施例的多个方面的注册处理的数据流；

[0046] 图 9，面 B 示出了根据本发明的实施例的多个方面的互用性 (interoperability) 处理的流程图；

[0047] 图 10 示出了根据本发明的实施例的多个方面的认证处理的数据流；

[0048] 图 11 示出了根据本发明的实施例的多个方面的签名处理的数据流；

[0049] 图 12 示出了根据本发明的另一实施例的多个方面的数据流和加密 / 解密处理；

[0050] 图 13 示出了根据本发明的另一实施例的多个方面的信任引擎系统的简化框图；

[0051] 图 14 示出了根据本发明的另一实施例的多个方面的信任引擎系统的简化框图；

[0052] 图 15 示出了根据本发明的实施例的多个方面，图 14 的冗余模块的框图；

[0053] 图 16 示出了根据本发明的一个方面，评估认证的处理；

[0054] 图 17 示出了根据在本发明的图 16 中所示的一个方面，为认证赋值的处理；

[0055] 图 18 示出了在图 17 中所示的本发明的一个方面中执行信任裁决的处理；

[0056] 图 19 示出了根据本发明的实施例的多个方面，在用户和卖方之间的范例交易，其中，初始基于网页的联系引向由双方签署的销售合同；

[0057] 图 20 示出了具有为用户系统提供安全性功能的加密服务提供者模块的范例用户系统；

[0058] 图 21 示出了解析、拆分和 / 或分离被加密的数据以及将加密主密钥与数据一起存储的处理；

[0059] 图 22 示出了解析、拆分和 / 或分离被加密的数据以及将加密主密钥与数据分开存储的处理；

[0060] 图 23 示出了解析、拆分和 / 或分离被加密的数据以及将加密主密钥与数据一起存储的中间密钥处理；

[0061] 图 24 示出了解析、拆分和 / 或分离被加密的数据以及将加密主密钥与数据分开存储的中间密钥处理；

[0062] 图 25 示出了对于小工作组，本发明的加密方法和系统的使用；

[0063] 图 26 是采用根据本发明一个实施例的安全数据解析器的示例性物理令牌安全系统的框图；

[0064] 图 27 是根据本发明的一个实施例，将安全数据解析器集成在系统中的示例性布置的框图；

[0065] 图 28 是根据本发明一个实施例的在移动系统中的示例性数据的框图；

[0066] 图 29 是根据本发明一个实施例的在移动系统中的另一示例性数据的框图；

[0067] 图 30-32 是根据本发明一个实施例的具有集成的安全数据解析器的示例性系统的框图；

[0068] 图 33 是根据本发明的一个实施例的用于解析和拆分数据的示例性处理的处理流程图；

[0069] 图 34 是根据本发明的一个实施例的用于将多个部分的数据恢复成原始数据的示例性处理的处理流程图；

[0070] 图 35 是根据本发明的一个实施例的用于以位级拆分数据的示例性处理的处理流程图；

[0071] 图 36 是根据本发明的一个实施例的示例性步骤和特征的处理流程图，其可以具有任何适当添加、删除或修改，或以任何适当组合的形式被使用；

[0072] 图 37 是根据本发明的一个实施例的示例性步骤和特征的处理流程图，其可以具有任何适当添加、删除或修改，或以任何适当组合的形式被使用；

[0073] 图 38 是根据本发明的一个实施例，份中的密钥和数据成分的存储的简化框图，其可以具有任何适当添加、删除或修改，或以任何适当组合的形式被使用；

[0074] 图 39 是根据本发明的一个实施例，使用工作组密钥的份中的密钥和数据成分的存储的简化框图，其可以具有任何适当添加、删除或修改，或以任何适当组合的形式被使用；

[0075] 图 40A 和 40B 是用于移动数据的头生成和数据拆分的简化和示意性处理流程图，其可以具有任何适当添加、删除或修改，或以任何适当组合的形式被使用；

[0076] 图 41 是根据本发明的一个实施例的示例性份格式的简化框图，其可以具有任何适当添加、删除或修改，或以任何适当组合的形式被使用；

[0077] 图 42 是根据本发明的一个实施例的用于管理加密密钥的示例性步骤和特征的处理流程图，其可以具有任何适当添加、删除或修改，或以任何适当组合的形式被使用。

## 具体实施方式

[0078] 本发明的一个方面提供一种加密系统，其中一个或多个安全服务器或信任引擎存储加密密钥和用户认证数据。用户通过对信任引擎的网络访问来访问传统加密系统的功能，然而，信任引擎不释放实际密钥和其他认证数据，因此，密钥和数据保持安全。密钥和认证数据的这种服务器中心存储提供用户无关的安全性、便携性、可用性以及简单性。

[0079] 因为用户能够确信或信任该加密系统来执行用户和文档认证以及其他加密功能，所以广泛的各种功能可以被结合到该系统中。例如，信任引擎提供者可以通过例如认证协定参与者、以参与者的名义或者为参与者数字签署协定、以及存储由每个参与者数字签署的协定的记录，来确保防备协定抵赖 (repudiation)。此外，该加密系统可以监视协定并基于例如价格、用户、卖方、地理位置、使用地点等来确定应用不同程度的认证。

[0080] 为了便于完全理解本发明，具体实施方式的余下部分参考附图描述本发明，其中通篇，类似的部件使用类似的标号来表示。

[0081] 图 1 示出了根据本发明的实施例的多个方面的加密系统 100 的框图。如图 1 所示，加密系统 100 包括通过通信链路 125 进行通信的用户系统 105、信任引擎 110、证书颁发机构 (certificateauthority)115、以及卖方系统 120。

[0082] 根据本发明的一个实施例，用户系统 105 包括传统通用目的计算机，其具有一个或多个微处理器，例如基于 Intel 的处理器。此外，用户系统 105 包括适当的操作系统，例如，能够包括图形或窗口的操作系统，诸如 Windows、Unix、Linux 等等。如图 1 所示，用户系统 105 可以包括生物识别装置 107。生物识别装置 107 可以有利地获取用户的生物识别信息并将获取的生物识别信息传递至信任引擎 110。根据本发明的一个实施例，生物识别装置可以有利地包括具有类似于在 1997 年 9 月 5 日提交的标题为“RELIEF OBJECTIMAGE GENERAOR”的美国专利申请 No.08/926,277、2000 年 4 月 26 日提交的标题为“IMAGING DEVICE FOR A RELIEF OBJECTAND SYSTEM AND METHOD OF USING THE IMAGE DEVICE”的美国专利申请 No.09/558,634、1999 年 11 月 5 日提交的标题为“RELIEF OBJECT SENSOR ADAPTOR”的美国专利申请 No.09/435,011、以及 2000 年 1 月 5 日提交的标题为“PLANAROPTICAL IMAGE SENSOR AND SYSTEM FOR GENERATINGAN ELECTRONIC IMAGE OF A RELIEF OBJECT FORFINGERPRINT READING”的美国专利申请 No.09/477,943 中公开的那些

属性和特征的装置，所有这些专利由当前受让人所有，以及全部通过引用结合于此。

[0083] 此外，用户系统 105 可以通过传统业务提供商，例如，拨号、数字订户线路 (DSL)、有线调制解调器、光纤连接等等，连接到通信链路 125。根据另一实施例，用户系统 105 通过诸如局域网或广域网之类的网络连接而连接通信链路 125。根据一个实施例，操作系统包括 TCP/IP 栈，其处理在通信链路 125 上传递的所有输入和输出消息业务。

[0084] 尽管参考上述实施例描述了用户系统 105，但是本发明不因此而被限制。相反，本领域技术人员由此处的披露应该意识到用户系统 105 的大量可替换实施例，包括能够发送信息至另一计算机系统或从另一计算机系统接收信息的几乎任何计算装置。例如，用户系统 105 可以包括但不限于，能够与通信链路 125 交互的计算机工作站、交互式电视、交互式信息站、个人移动计算装置（诸如数字助理、移动电话、膝上型电脑等等）、无线通信装置、智能卡、嵌入式计算装置等等。在这样的可替换系统中，操作系统很可能不同，并且适于特定装置。然而，根据一个实施例，操作系统有利地继续提供与通信链路建立通信所需的适当的通信协议。

[0085] 图 1 示出了信任引擎 110。根据一个实施例，该信任引擎 110 包括用于访问和存储敏感信息的一个或多个安全服务器，敏感信息可以是任何类型或形式的数据，诸如但不限于文本、音频、视频、用户认证数据以及公共和私有加密密钥。根据一个实施例，认证数据包括被设计为唯一识别加密系统 100 的用户的数据。例如，认证数据可以包括用户标识号、一个或多个生物识别信息、以及由信任引擎 100 或用户生成但由用户在注册时初始回答的一系列问题和答案。上述问题可以包括诸如出生地、地址、周年纪念之类的人口统计数据，诸如母亲的娘家姓、最喜欢的冰激凌之类的个人数据，或被设计为唯一识别用户的其他数据。信任引擎 110 将与当前交易相关联的用户的认证数据与较早时候（例如注册期间）提供的认证数据进行比较。信任引擎 110 可以有利地要求用户在每次交易时产生认证数据，或者，信任引擎 110 可以有利地允许用户周期性地产生认证数据，例如在一串交易的开始或在登录到特定卖方网站时。

[0086] 根据用户产生生物识别数据的实施例，用户向生物识别装置 107 提供身体特征，诸如但不限于脸部扫描、手扫描、耳朵扫描、虹膜扫描、视网膜扫描、血管模式、DNA、指纹、笔迹、或语音。生物识别装置有利地产生身体特征的电子图案或生物识别信息。电子图案通过用户系统 105 被传递至信任引擎 110 用于注册或认证目的。

[0087] 一旦用户产生适当的认证数据并且信任引擎 110 确定认证数据（当前认证数据）与在注册时提供的认证数据（注册认证数据）之间的肯定匹配，则信任引擎 110 向用户提供完全的加密功能。例如，被正确认证的用户可以有利地使用信任引擎 110 来执行散列、数字签名、加密和解密（通常一起仅称为加密）、创建或分发数字证书、等等。然而，用在加密功能中的私有加密密钥在信任引擎 110 之外不可用，因此保证加密密钥的完整性。

[0088] 根据一个实施例，信任引擎 110 生成并存储加密密钥。根据另一实施例，至少一个加密密钥与每个用户相关联。此外，当加密密钥包括公钥技术时，与用户相关的每个私钥在信任引擎 110 中生成，而不从其释放。因此，只要用户已经访问了信任引擎 110，用户就可以使用他或她的私钥或公钥执行加密功能。这样的远程访问有利地允许用户保留完全的移动性，以及通过实际中的任何因特网连接（例如蜂窝和卫星电话、信息站、膝上型电脑、宾馆房间等）访问加密功能。

[0089] 根据另一实施例，信任引擎 110 使用为信任引擎 110 生成的密钥对来执行加密功能。根据该实施例，信任引擎 110 首先认证该用户，然后在用户已经正确产生与注册认证数据匹配的认证数据之后，信任引擎 110 使用其本身的加密密钥对，以被认证的用户的名义执行加密功能。

[0090] 本领域的技术人员从此处的公开应意识到，加密密钥可以有利地包括一些或所有对称密钥、公钥和私钥。此外，本领域的技术人员从此处的公开应意识到，上述密钥可以使用诸如 RSA、ELGAMAL 等商业技术提供的多种算法来实现。

[0091] 图 1 还示出了证书颁发机构 115。根据一个实施例，证书颁发机构 115 可以有利地包括发布数字证书的可信的第三方组织或公司，例如，VeriSign，Baltimore，Entrust 等等。信任引擎 110 可以有利地通过一个或多个传统数字证书协议（例如 PKCS10）向证书颁发机构 115 发送数字证书请求。作为响应，证书颁发机构 115 将以多种不同协议中的一种或多种发布数字证书，例如 PKCS7。根据本发明的一个实施例，信任引擎 110 从多个或所有著名的证书颁发机构 115 请求数字证书，从而信任引擎 110 能够使用与任何请求方的证书标准相对应的数字证书。

[0092] 根据另一实施例，信任引擎 110 在内部执行证书发布。在该实施例中，信任引擎 110 可以访问证书系统以生成证书和 / 或可以在证书被请求时，例如在密钥生成时，或者以请求时所请求的证书标准，在内部生成证书。下面将更加详细地披露信任引擎 110。

[0093] 图 1 还示出了卖方系统 120。根据一个实施例，卖方系统 120 有利地包括 Web 服务器。典型的 Web 服务器通常使用多种互联网标记语言或文档格式标准（例如超文本标记语言 (HTML) 或可扩展标记语言 (XML)）中的一种在因特网上提供内容。Web 服务器接受来自诸如 Netscape 和 Internet Explorer 之类的浏览器的请求，然后返回适当的电子文档。多种服务器或客户端技术可以用来在传递标准电子文档的能力之外增加 Web 服务器的能力。例如，这些技术包括公共网关接口 (CGI) 脚本、安全套接字层 (SSL) 安全、以及动态服务器网页 (ASP)。卖方系统 120 可以有利地提供关于商业、个人、教育或其他交易的电子内容。

[0094] 尽管参考上面实施例披露了卖方系统 120，但是本发明不因此被限制。相反，本领域技术人员应该从此处的公开意识到，卖方系统 120 可以有利地包括参考用户系统 105 描述的任何装置或其组合。

[0095] 图 1 还示出了连接用户系统 105、信任引擎 110、证书颁发机构 115 和卖方系统 120 的通信链路 125。根据一个实施例，通信链路 125 优选地包括因特网。如在整个公开中所使用的，因特网是计算机的全球网络。本领域普通技术人员公知的因特网的结构包括网络骨干 (network backbone)，以及从骨干分支的网络。这些分支又具有从它们分支的网络，如此等等。路由器在网络级之间移动信息包，然后从网络至网络，直到包到达其目的地附近。从该目的地，目的地网络的主机将该信息包引向适当的终端或节点。在一个有利的实施例中，因特网路由集线器包括本领域公知的使用传输控制协议 / 网际协议 (TCP/IP) 的域名系统 (DNS) 服务器。路由集线器通过高速通信链路连接到一个或多个其他路由集线器。

[0096] 因特网的一个普及部分是万维网。万维网包括不同的计算机，其存储能够显示图形或文本信息的文件。在万维网上提供信息的计算机通常被称作“网站”。网站由具有相关电子页面的因特网地址定义。电子页面可以由统一资源定位符 (URL) 标识。通常，电子页面是组织文本、图形图像、音频、视频等的呈现的文档。

[0097] 尽管通信链路 125 以其优选实施例被公开，但是本领域的技术人员应该从在此的公开意识到，通信链路 125 可以包括广泛的交互通信链路。例如，通信链路 125 可以包括交互式电视网络、电话网络、无线数据传输系统、双向电缆系统、定制的私有或公共计算机网络、交互式信息站网络、自动柜员机网络、直接链路、卫星或蜂窝网络、等等。

[0098] 图 2 示出了根据本发明的一个实施例的多个方面的图 1 的信任引擎 110 的框图。如图 2 所示，信任引擎 110 包括交易引擎 205、仓库 210、认证引擎 215、以及加密引擎 220。根据本发明的一个实施例，信任引擎 110 还包括大容量存储器 225。如图 2 进一步所示，交易引擎 205 与仓库 210、认证引擎 215、以及加密引擎 220、连同大容量存储器 225 通信。此外，仓库 210 与认证引擎 215、加密引擎 220 以及大容量存储器 225 通信。此外，认证引擎 215 与加密引擎 220 通信。根据本发明的一个实施例，上述通信中的一些或所有都可以有利地包括传输 XML 文档至对应于接收装置的 IP 地址。如上所述，XML 文档有利地允许设计者创建其自身的定制文档标签，使得能够在应用程序之间和组织之间定义、传输、验证和解释数据。此外，上述通信中的一些或所有可以包括传统 SSL 技术。

[0099] 根据一个实施例，交易引擎 205 包括数据路由装置，诸如由 Netscape、Microsoft、Apache 等提供的传统 Web 服务器。例如，Web 服务器可以有利地接收来自通信链路 125 的输入数据。根据本发明的一个实施例，输入数据被定址到用于信任引擎 110 的前端安全系统。例如，前端安全系统可以有利地包括防火墙、搜索已知攻击特征的入侵检测系统、和 / 或病毒扫描器。在通过前端安全系统之后，数据由交易引擎 205 接收并路由至仓库 210、认证引擎 215、加密引擎 220 以及大容量存储器 225 之一。此外，交易引擎 205 监视来自认证引擎 215 和加密引擎 220 的输入数据，并通过通信链路 125 将数据路由至特定系统。例如，交易引擎 205 可以有利地将数据路由至用户系统 105、证书颁发机构 115 或卖方系统 120。

[0100] 根据一个实施例，数据使用传统 HTTP 路由技术被路由，例如采用 URL 或统一资源指示符 (URI)。URI 类似于 URL，然而，URI 通常指示文件或动作的源，例如，可执行文件、脚本等。因此，根据一个实施例，用户系统 105、证书颁发机构 115、卖方系统 120、以及信任引擎 210 的部件有利地包括通信 URL 或 URI 中的足够数据以便交易引擎 205 在加密系统中正确地路由数据。

[0101] 尽管参考其优选实施例公开了数据路由，但是本领域的技术人员应该意识到大量的可行的数据路由解决方案或策略。例如，XML 或其他数据包可以有利地通过其格式、内容等被拆包和识别，从而交易引擎 205 可以在信任引擎 110 中正确地路由数据。此外，本领域技术人员应该意识到，数据路由可以有利地适应符合特定网络系统——例如当通信链路 125 包括局域网时——的数据传输协议。

[0102] 根据本发明的另一实施例，交易引擎 205 包括传统 SSL 加密技术，从而上述系统可以在特定通信期间使用交易引擎 205 认证自身，反之亦然。如在整个公开中所使用的，术语“1/2SSL”表示服务器被 SSL 认证而客户端不一定被 SSL 认证的通信，术语“全 SSL”表示客户端和服务器都被 SSL 认证的通信。在当前公开使用术语“SSL”时，通信可以包括 1/2 或全 SSL。

[0103] 当交易引擎 205 将数据路由至加密系统 100 的各部件时，交易引擎 205 可以有利地创建审计跟踪 (audit trail)。根据一个实施例，审计跟踪包括至少关于由交易引擎 205 在加密系统 100 中路由的数据的类型和格式的记录。这样的审计数据可以有利地存储在大

容量存储器 225 中。

[0104] 图 2 还示出了仓库 210。根据一个实施例，仓库 210 包括一个或多个数据存储设备，诸如目录服务器、数据库服务器等。如图 2 中所示，仓库 210 存储加密密钥和注册认证数据。加密密钥可以有利地对应于信任引擎 110 或者对应于加密系统 100 的用户，诸如用户或卖方。注册认证数据可以有利地包括被设计为唯一识别用户的数据，诸如用户 ID、口令、问题答案、生物识别数据等。该注册认证数据可以有利地在用户注册时或其他可替换的稍后时间被获取。例如，信任引擎 110 可以包括注册认证数据的周期性的或别的更新或重新发布。

[0105] 根据一个实施例，往返于交易引擎 205 与认证引擎 215 和加密引擎 220 之间的通信包括安全通信，例如，传统 SSL 技术。此外，如上所述，往返于仓库 210 的通信的数据可以使用 URL、URI、HTTP 或 XML 文档传递，它们有利地将数据请求和格式嵌入其中。

[0106] 如上所述，仓库 210 可以有利地包括多个安全数据存储设备。在这样的实施例中，安全数据存储设备可以被配置为使得单个数据存储设备中的安全性损害不会泄露存储在其中的加密密钥或认证数据。例如，根据该实施例，对加密密钥和认证数据进行数学运算以便统计地并基本上随机化存储在每个数据存储设备中的数据。根据一个实施例，单个数据存储设备的数据的随机化使得数据不可破译。因此，单个数据存储设备的泄密仅仅产生随机的不可破译的数字并且不损害作为整体的加密密钥或认证数据的安全性。

[0107] 图 2 还示出了信任引擎 110 包括认证引擎 215。根据一个实施例，认证引擎 215 包括数据比较器，其被配置为将来自交易引擎 205 的数据与来自仓库 210 的数据进行比较。例如，在认证过程中，用户将当前认证数据提供至信任引擎 110，从而交易引擎 205 接收到该当前认证数据。如上所述，交易引擎 205 识别优选为 URL 或 URI 的数据请求，并将认证数据路由至认证引擎 215。此外，基于请求，仓库 210 将对应于该用户的注册认证数据转发至认证引擎 215。从而，认证引擎 215 具有当前认证数据和注册认证数据这两者以供比较。

[0108] 根据一个实施例，至认证引擎的通信包括安全通信，诸如 SSL 技术。此外，可以在信任引擎 110 部件中提供安全性，例如使用公钥技术的超级加密。例如，根据一个实施例，用户使用认证引擎 215 的公钥来加密当前认证数据。此外，仓库 210 还使用认证引擎 215 的公钥来加密注册认证数据。以该方式，只有认证引擎的私钥可以被用来解密该传输。

[0109] 如图 2 中所示，信任引擎 110 还包括加密引擎 220。根据一个实施例，加密引擎包括加密处理模块，其被配置为有利地提供传统加密功能，诸如公钥基础结构 (PKI) 功能。例如，加密引擎 220 可以有利地发布用于加密系统 100 的用户的公钥和私钥。以该方式，加密密钥在加密引擎 220 处生成，并被转发至仓库 210，从而至少私有加密密钥在信任引擎 110 外部不可用。根据另一实施例，加密引擎 220 至少随机化并拆分私有加密密钥数据，从而仅存储随机化的拆分的数据。类似于注册认证数据的拆分，拆分处理保证所存储的密钥在加密引擎 220 的外部不可用。根据另一实施例，加密引擎的功能可以与认证引擎 215 结合并由其执行。

[0110] 根据一个实施例，往返于加密引擎的通信包括安全通信，诸如 SSL 技术。此外，XML 文档可以有利地被用于传递数据和 / 或进行加密功能请求。

[0111] 图 2 还示出了信任引擎 110 具有大容量存储器 225。如上所述，交易引擎 205 保持对应于审计跟踪的数据，并将这样的数据存储在大容量存储器 225 中。类似地，根据本发明

的一个实施例，仓库 210 保持对应于审计跟踪的数据并将这样的数据存储在大容量存储装置 225 中。仓库审计跟踪数据类似于交易引擎 205 的审计跟踪数据，因为审计跟踪数据包括由仓库 210 接收的请求的记录及其响应。此外，大容器存储器 225 可以被用来存储其中包含有用户公钥的数字证书。

[0112] 尽管参考其优选和可替换实施例公开了信任引擎 110，但本发明不因此被限制。相反，本领域技术人员在此处的公开中应该意识到信任引擎 110 的大量替换例。例如，信任引擎 110 可以有利地仅执行认证，或可替换地，仅执行部分或全部加密功能，诸如数据加密和解密。根据这样的实施例，认证引擎 215 和加密引擎 220 之一可以有利地被去除，从而创建更加简单的信任引擎 110 设计。此外，加密引擎 220 还可以与证书颁发机构通信，从而证书颁发机构被包括在信任引擎 110 中。根据另一实施例，信任引擎 110 可以有利地执行认证和一个或多个加密功能，例如数字签名。

[0113] 图 3 示出了根据本发明的实施例的多个方面，图 2 的交易引擎 205 的框图。根据该实施例，交易引擎 205 包括具有处理线程和监听线程的操作系统 305。操作系统 305 可以有利地类似于在传统大容量服务器中的那些操作系统，例如由 Apache 提供的 Web 服务器。监听线程监视来自通信链路 125、认证引擎 215、以及加密引擎 220 之一的输入通信中的输入数据流。处理线程识别输入数据流的特定数据结构，例如前述的数据结构，从而将输入数据路由至通信链路 125、仓库 210、认证引擎 215、加密引擎 220、或大容量存储器 225 之一。如图 3 所示，输入和输出数据可以有利地通过例如 SSL 技术而被保护。

[0114] 图 4 示出了根据本发明的实施例的多个方面的图 2 的仓库 210 的框图。根据该实施例，仓库 210 包括一个或多个轻量级目录访问协议 (LDAP) 服务器。LDAP 目录服务器可以由多个制造商提供，诸如 Netscape、ISO 和其他制造商。图 4 还示出，目录服务器优选存储对应于加密密钥的数据 405 和对应于注册认证数据的数据 410。根据一个实施例，仓库 210 包括单个逻辑存储结构，其将认证数据和加密密钥数据索引至唯一用户 ID。该单个逻辑存储结构优选地包括保证存储在其中的数据具有高信任度或安全性的机制。例如，仓库 210 的物理位置可以有利地包括多种传统安全措施，诸如有限的雇员访问、现代监视系统等。在物理安全措施之外或取而代之，计算机系统或服务器可以有利地包括软件解决方案来保护存储的数据。例如，仓库 210 可以有利地创建并存储与所执行的动作的审计跟踪相对应的数据 415。此外，输入和输出通信可以有利地使用与传统 SSL 技术相结合的公钥加密来加密。

[0115] 根据另一实施例，仓库 210 可以包括不同的且物理上分开的数据存储设备，如进一步参考图 7 所述。

[0116] 图 5 示出了根据本发明的实施例的多个方面，图 2 的认证引擎 215 的框图。类似于图 3 的交易引擎 205，认证引擎 215 包括操作系统 505，其至少具有传统 Web 服务器（例如 Apache 提供的 Web 服务器）的修改版本的监听和处理线程。如图 5 所示，认证引擎 215 包括对至少一个私钥 510 的访问。私钥 510 可以有利地被用来例如解密来自交易引擎 205 或仓库 210 的使用认证引擎 215 的相应公钥加密过的数据。

[0117] 图 5 还示出了认证引擎 215 包括比较器 515、数据拆分模块 529、以及数据组装模块 525。根据本发明的优选实施例，比较器 515 包括能够比较与上述生物识别认证数据有关的可能复杂的图案的技术。该技术可以包括用于图案比较的硬件、软件或其组合方案，所

述图案诸如是表示指纹图案或语音图案的那些图案。此外，根据一个实施例，认证引擎 215 的比较器 515 可以有利地比较文档的传统散列值以给出比较结果。根据本发明的一个实施例，比较器 515 包括将试探法 (heuristics)530 应用于比较。试探法 530 可以有利地处理围绕认证尝试的详情，例如一天中的时间、IP 地址或子网掩码、购买简档、电子邮件地址、处理器序列号或 ID、等等。

[0118] 此外，生物识别数据比较的特性可能使得由当前生物识别认证数据与注册数据的匹配产生变化的可信度。例如，与可能仅返回肯定或否定匹配的传统口令不同，指纹可能被确定为是部分匹配，例如，90％匹配、75％匹配或 10％匹配，而不是简单地正确或不正确。诸如声纹分析或面貌识别之类的其他生物识别标识也可以具有该概率认证性质，而不是绝对认证。

[0119] 当使用这样的概率认证或者在认证被认为并非绝对可靠的情况下，希望应用试探法 530 来确定所提供的认证中的可信度是否足够高到能够认证正在进行的交易。

[0120] 有时候的情况是，待解决的交易是相对低价值的交易，其中被认证为较低可信度是可以接受的。这可能包括与其相关的美金值较低的交易（例如，$10 购买）或具有较低风险的交易（例如，仅会员可进的网站）。

[0121] 相反，为了认证其他交易，希望在允许交易进行之前要求认证的高可信度。这样的交易可能包括大额美金值的交易（例如，签订几百万美金供应合同）或如果发生不正确的认证会具有高风险的交易（例如远程登录至政府计算机）。

[0122] 下面将描述使用试探法 530 与可信度和交易值结合，以允许比较器提供动态的上下文相关的认证系统。

[0123] 根据本发明的另一实施例，比较器 515 可以有利地跟踪对于特定交易的认证尝试。例如，当交易失败时，信任引擎 110 可以请求用户重新输入他或她的当前认证数据。认证引擎 215 的比较器 515 可以有利地使用尝试限制器 535 来限制认证尝试的次数，从而禁止试图模仿用户的认证数据的暴力尝试。根据一个实施例，尝试限制器 535 包括监视交易中的重复认证尝试并例如将对于给定交易的认证尝试限制为三次的软件模块。因此，尝试限制器 535 将限制模仿个体的认证数据的自动化尝试，例如简单的三次“guesses”。一旦三次失败，尝试限制器 535 可以有利地拒绝追加的认证尝试。这样的拒绝可以有利地通过例如不管被传输的当前认证数据是什么，比较器 515 都返回否定结果来实现。另一方面，交易引擎 205 可以有利地阻止与之前已有三次失败尝试的交易有关的任何其它的认证尝试。

[0124] 认证引擎 215 还包括数据拆分模块 520 和数据组装模块 525。数据拆分模块 520 有利地包括软件、硬件或组合模块，具有对各种数据进行数学运算从而基本上随机化数据并将其拆分成多个部分的能力。根据一个实施例，原始数据不能从单个部分中重建。数据组装模块 525 有利地包括软件、硬件或组合模块，其被配置为对上述基本上随机化的部分进行数学运算，从而其组合提供原始的被破译数据。根据一个实施例，认证引擎 215 使用数据拆分模块 520 来随机化注册认证数据并将其拆分成多个部分，以及使用数据组装模块 525 来将这些部分重新组装成可用的注册认证数据。

[0125] 图 6 示出了根据本发明的一个实施例的多个方面，图 2 的信任引擎 200 的加密引擎 220 的框图。类似于图 3 的交易引擎 205，加密引擎 220 包括操作系统 605，其至少具有传统 Web 服务器（例如，Apache 提供的 Web 服务器）的修改版本的监听线程和处理线程。

如图 6 中所示，加密引擎 220 包括数据拆分模块 610 和数据组装模块 620，其功能类似于图 5 中的那些模块。然而，根据一个实施例，数据拆分模块 610 和数据组装模块 620 处理加密密钥数据，不同于上述注册认证数据。尽管如此，本领域技术人员从此处的公开应意识到数据拆分模块 610 和数据组装模块 620 可以与认证引擎 215 的数据拆分模块和数据组装模块结合。

[0126] 加密引擎 220 还包括加密处理模块 625，其被配置为执行大量加密功能中的一个、部分或所有。根据一个实施例，加密处理模块 625 可以包括软件模块或程序、硬件、或两者。根据另一实施例，加密处理模块 625 可以执行数据比较、数据解析、数据拆分、数据分离、数据散列、数据加密或解密、数字签名校验或创建、数字证书的生成、存储或请求、加密密钥生成、等等。此外，本领域技术人员应该从在此处的公开意识到，加密处理模块 625 可以有利地包括公钥基础结构，诸如优秀保密 (Pretty Good Privacy，PGP)、基于 RSA 的公钥系统、或大量可替换的密钥管理系统。此外，加密处理模块 625 可以执行公钥加密、对称密钥加密、或两者。除了上面所述，加密处理模块 625 可以包括一个或多个计算机程序或模块、硬件、或两者，用于实现无缝、透明、互用性功能。

[0127] 本领域的技术人员从在此的公开还应该意识到，加密功能可以包括通常与加密密钥管理系统有关的大量或多种功能。

[0128] 图 7 示出了根据本发明的实施例的多个方面的仓库系统 700 的简化框图。如图 7 所示，仓库系统 700 有利地包括多个数据存储设备，例如，数据存储设备 D1、D2、D3 和 D4。然而，本领域的普通技术人员容易理解仓库系统可能仅具有一个数据存储设备。根据本发明的一个实施例，每个数据存储设备 D1 至 D4 可以有利地包括参考图 4 的仓库 210 公开的部分或所有元件。类似于仓库 210，数据存储设备 D1 至 D4 与交易引擎 205、认证引擎 215、以及加密引擎 220 通信，优选通过传统 SSL 通信。通信链路传输例如 XML 文档。来自交易引擎 205 的通信可以有利地包括对数据的请求，其中该请求被有利地广播至每个数据存储设备 D1 至 D4 的 IP 地址。另一方面，交易引擎 205 可以基于诸如响应时间、服务器负荷、维护计划等大量标准而将请求广播至特定数据存储设备。

[0129] 响应于来自交易引擎 205 的对于数据的请求，仓库系统 700 有利地转发所存储的数据至认证引擎 215 和加密引擎 220。各个数据组装模块接收转发的数据并将数据组装成可用格式。另一方面，从认证引擎 215 和加密引擎 220 到数据存储设备 D1 至 D4 的通信可以包括传输要存储的敏感数据。例如，根据一个实施例，认证引擎 215 和加密引擎 220 可以有利地使用其各自的数据拆分模块以将敏感数据分解成不可破译的部分，然后将敏感数据的一个或多个不可破译的部分传输至特定数据存储设备。

[0130] 根据一个实施例，每个数据存储设备 D1 至 D4 包括分开且独立的存储系统，例如目录服务器。根据本发明的另一实施例，仓库系统 700 包括多个地理上分开的独立的数据存储系统。通过将敏感数据分发至不同且独立的存储设备 D1 至 D4——其中部分或所有存储设备可以有利地在地理上分开，仓库系统 700 与其他安全措施一起提供冗余。例如，根据一个实施例，仅来自多个数据存储设备 D1 至 D4 中的两个数据存储设备的数据需要解密并重新组装该敏感数据。因此，四个数据存储设备 D1 至 D4 中的两个数据存储设备可以由于维护、系统故障、电源故障等而不工作，但不影响信任引擎 110 的功能。此外，因为根据一个实施例，存储在每个数据存储设备中的数据被随机化并不可破译，所以任何单个数据存储设

备的泄密不必然地泄露敏感数据。此外，在数据存储设备地理上分开的实施例中，多个地理上远离的设备的泄密变得越发困难。事实上，不良员工要暗中破坏所需的多个独立的地理上远离的数据存储设备将面临更大的挑战。

[0131] 尽管参考其优选和可替换实施例描述了仓库系统 700，但是本发明不因此被限制。相反，本领域技术人员将由在此的公开意识到仓库系统 700 的多个替换例。例如，仓库系统 700 可以包括一个、两个或更多数据存储设备。此外，敏感数据可以被数学运算，使得需要来自两个或更多数据存储设备的部分以便重新组装并解密该敏感数据。

[0132] 如上所述，认证引擎 215 和加密引擎 220 每个都分别包括数据拆分模块 520 和 610，用于拆分任何类型或形式的敏感数据，诸如文本、音频、视频、认证数据和加密密钥数据。图 8 示出了根据本发明的实施例的多个方面，由数据拆分模块执行的数据拆分处理 800 的流程图。如图 8 所示，数据拆分处理 800 在步骤 805 处开始，此时敏感数据“S”由认证引擎 215 或加密引擎 220 的数据拆分模块接收。优选地，在步骤 810，数据拆分模块然后生成基本上随机的数、值、或串、或位组“A”。例如，随机数 A 可以用本领域技术人员已知的用于产生适于在加密应用中使用的高质量随机数的多种不同传统技术生成。此外，根据一个实施例，随机数 A 包括可以是任何适当长度的位长度，诸如短于、长于、或等于敏感数据 S 的位长度。

[0133] 此外，在步骤 820 中，数据拆分处理 800 生成另一统计上随机的数“C”。根据优选实施例，统计上随机的数 A 和 C 可以有利地并行生成。数据拆分模块然后将数 A 和 C 与敏感数据 S 结合，从而生成新的数“B”和“D”。例如，数 B 可以包括 A XOR S 的二进制结合，数 D 可以包括 C XOR S 的二进制结合。XOR 函数或“异或”函数对于本领域的技术人员是已知的。上述结合优选地分别发生在步骤 825 和 830，以及根据一个实施例，上述结合还并行发生。数据拆分处理 800 然后进行到步骤 835，其中随机数 A 和 C 以及数 B 和 D 被配对，使得没有哪个配对本身包含重新组织和解密原始敏感数据 S 的足够数据。例如，这些数可以被如下配对 :AC，AD，BC 和 BD。根据一个实施例，每一个上述配对被分配至图 7 中的仓库 D1 至 D4 之一。根据另一实施例，每个上述配对被随机分配给仓库 D1 至 D4 之一。例如，在第一数据拆分处理 800 的过程中，配对 AC 可以通过例如随机选择的 D2 的 IP 地址被发送至仓库 D2。然后，在第二数据拆分处理 800 的过程中，配对 AC 可以通过例如随机选择的 D4 的 IP 地址被发送至仓库 D4。此外，这些配对所有都可以被存储在一个仓库，以及可以被存储在所述仓库中分开的位置。

[0134] 基于上面所述，数据拆分处理 800 有利地将敏感数据的多个部分放置在四个数据存储设备 D1 至 D4 中的每一个中，从而没有单个数据存储设备 D1 至 D4 包括重建原始敏感数据 S 的足够的被加密的数据。如上所述，这样的将数据随机化为单独不可用的加密部分提升了安全性并提供了对数据的信任的维持，即使数据存储设备 D1 至 D4 之一被泄密。

[0135] 尽管参考其优选实施例公开了数据拆分处理 800，本发明不因此被限制。相反，本领域的技术人员应从此处的公开意识到数据拆分处理 800 的多种替换。例如，数据拆分处理可以有利地将数据拆分成两个数，例如，随机数 A 和数 B，并且通过两个数据存储设备随机分配 A 和 B。此外，数据拆分处理 800 可以有利地通过生成另外的随机数而在大量数据存储设备中拆分数据。数据可以被拆分成任何期望的、选择的、预定的或随机分配的尺寸单元，包括但不限于一位、多位、字节、千字节、兆字节或更大、或尺寸的任何组合或序列。

此外，由拆分处理导致的改变数据单元的尺寸可以使得数据更难以恢复成可用形式，从而增加敏感数据的安全性。本领域的普通技术人员容易理解，拆分数据单元的尺寸可以是多种不同的数据单元尺寸或尺寸的图样 (pattern) 或尺寸的组合。例如，数据单元尺寸可以被选择或预定为都具有相同的尺寸、具有不同尺寸的固定的组、尺寸的组合、或随机生成尺寸。类似地，数据单元可以根据固定或预定的数据单元尺寸、数据单元尺寸的图样或组合、或随机生成的数据单元尺寸或每份的尺寸而被分配成一份或多份。

[0136] 如上所述，为了重建敏感数据 S，数据部分需要被解随机和重新组织。该处理可以有利地分别发生在认证引擎 215 和加密引擎 220 的数据组装模块 525 和 620 中。数据组装模块，例如数据组装模块 525，接收来自数据存储设备 D1 至 D4 的数据部分，以及将数据组装成可用形式。例如，根据一个实施例，数据拆分模块 520 使用图 8 的数据拆分处理 800，而数据组装模块 525 使用来自数据存储设备 D1 至 D4 中至少两个的数据部分以重建敏感数据 S。例如，配对 AC、AD、BC 和 BD 被分配以使得任何两个提供 A 和 B 或 C 和 D 之一。注意 S = A XOR B 或 S = C XOR D 表示当数据组装模块接收 A 和 B 或 C 和 D 之一时，数据组装模块 525 可以有利地重新组装敏感数据 S。因此，数据组装模块 525 可以在例如接收到来自数据存储设备 D1 至 D4 中至少前两个的数据部分时，组装敏感数据 S，以响应于信任引擎 110 的组装请求。

[0137] 基于上述的数据拆分和组装处理，敏感数据 S 仅在信任引擎 110 的有限区域中以可用格式存在。例如，当敏感数据 S 包括注册认证数据时，可用的未随机化的注册认证数据仅在认证引擎 215 中可用。类似地，当敏感数据 S 包括私有加密密钥数据时，可用的未随机化的私有加密密钥数据仅在加密引擎 220 中可用。

[0138] 尽管参考其优选实施例公开了数据拆分和组装处理，本发明不因此被限制。相反，本领域的技术人员从此处的公开应该意识到用于拆分和重新组装敏感数据 S 的多种替换。例如，公钥加密可以被用于在数据存储设备 D1 至 D4 处进一步保护数据。此外，本领域的技术人员容易理解在此描述的数据拆分模块也是本发明的单独且独立的实施例，可以合并到包括任何已有计算机系统、软件套件、数据库、或其组合、或本发明的其他实施例（诸如在此公开和描述的信任引擎、认证引擎以及交易引擎）中，或与之结合，或作为其中一部分。

[0139] 图 9A 示出了根据本发明的实施例的多个方面的注册处理 900 的数据流。如图 9A 所示，注册处理 900 在步骤 905 处开始，用户期望使用加密系统 100 的信任引擎 110 注册。根据该实施例，用户系统 105 有利地包括例如基于 Java 的客户端小程序 (applet)，其询问用户以输入注册数据，例如人口统计数据和注册认证数据。根据一个实施例，注册认证数据包括用户 ID、口令、生物识别信息等。根据一个实施例，在询问处理过程中，客户端小程序优选地与信任引擎 110 通信以保证所选择的用户 ID 是唯一的。当用户 ID 非唯一时，信任引擎 110 可以有利地建议唯一用户 ID。客户端小程序收集注册数据并将注册数据例如通过 XML 文档传输至信任引擎 110，更具体地，至交易引擎 205。根据一个实施例，用认证引擎 215 的公钥编码该传输。

[0140] 根据一个实施例，用户在注册处理 900 的步骤 905 期间执行单个注册。例如，用户将他或她本身注册为特定人，例如 Joe User。当 Joe User 期望注册为 Joe User，Mega 公司的 CEO 时，根据该实施例，Joe User 第二次注册，接收到第二个唯一用户 ID，并且信任引擎 110 不将两个身份相关联。根据本发明的另一实施例，注册处理 900 为单个用户 ID 提供多

个用户身份。从而，在上述示例中，信任引擎 110 有利地将 Joe User 的两个身份相关联。如本领域技术人员从此处的公开可以理解的，用户可以具有许多身份，例如，一家之主的 Joe User，慈善基金会成员 Joe User，等等。即使用户可以具有多个身份，根据该实施例，信任引擎 110 优选地仅存储一组注册数据。此外，用户可以根据他们的需要有利地增加、编辑 / 更新、或删除身份。

[0141] 尽管参考其优选实施例公开了注册处理 900，但是本发明不因此被限制。相反，本领域技术人员从此处的公开应理解用于收集注册数据尤其是注册认证数据的多种替换例。例如，小程序可以是基于通用对象模型 (COM) 的小程序，等等。

[0142] 另一方面，注册处理可以包括分级注册。例如，在最低级的注册中，用户可以通过通信链路 125 注册而不产生关于他或她身份的文件。根据注册等级的提高，用户使用诸如数字公证之类的可信第三方进行注册。例如，用户可以亲自出现在可信第三方面前，制作诸如出生证明、驾照、军人 ID 等，以及可信第三方可以有利地在注册提交时包括例如他们的数字签名。可信第三方可以包括实际公证处、政府机构（例如邮局或机动车部）、大公司中注册员工的人力资源人员等。本领域的技术人员从此处的公开应该理解在注册处理 900 中可能发生多种不同等级的注册。

[0143] 在步骤 915 处接收注册认证数据之后，交易引擎 205 使用传统全 SSL 技术将注册认证数据转发至认证引擎 215。在步骤 920，认证引擎 215 使用认证引擎 215 的私钥解密注册认证数据。此外，认证引擎 215 使用数据拆分模块对注册认证数据进行数学运算，从而将数据拆分成至少两个独立的不可破译的随机化的数。如上所述，至少两个数可以包括统计随机的数和二进制异或的数。在步骤 925，认证引擎 215 将随机化的数的每个部分转发给数据存储设备 D1 至 D4 之一。如上所述，认证引擎 215 还可以有利地随机化哪些部分被传递到哪些仓库。

[0144] 通常在注册处理 900 的过程中，用户还期望发布数字证书，从而他或她可以接收来自加密系统 100 外部的其他系统的加密文档。如上所述，证书颁发机构 115 通常根据多种传统标准中的一个或多个标准发布数字证书。通常，数字证书包括每个人都知道的用户或系统的公钥。

[0145] 无论用户是在注册时还是在其他时间请求数字证书，该请求通过信任引擎 110 传递至认证引擎 215。根据一个实施例，该请求包括具有例如用户正确姓名的 XML 文档。根据步骤 935，认证引擎 215 将该请求传递至加密引擎 220，指示加密引擎 220 生成加密密钥或密钥对。

[0146] 一经请求，在步骤 935，加密引擎 220 生成至少一个加密密钥。根据一个实施例，加密处理模块 625 生成密钥对，其中一个密钥被用作私钥，以及一个被用作公钥。加密引擎 220 存储私钥，以及根据一个实施例存储公钥的拷贝。在步骤 945 中，加密引擎 220 向交易引擎 205 传输数字证书请求。根据一个实施例，该请求有利地包括标准化请求，例如，嵌入在例如 XML 文档中的 PKCS10。数字证书请求可以有利地对应于一个或多个证书颁发机构，以及证书颁发机构要求的一个或多个标准格式。

[0147] 在步骤 950，交易引擎 205 转发该请求至证书颁发机构 115，后者在步骤 955 返回数字证书。返回的数字证书可以有利地为标准化格式，例如 PKCS7，或者为一个或多个证书颁发机构 115 的专有格式。在步骤 960 中，数字证书被交易引擎 205 接收，副本被转发至用

户，以及用信任引擎 110 存储副本。信任引擎 110 存储证书的副本，从而信任引擎 110 不需要依赖于证书颁发机构 115 的可用性。例如，当用户期望发送数字证书或第三方请求用户的数字证书时，数字证书请求通常被发送到证书颁发机构 115。然而，如果证书颁发机构 115 正在进行维护或已经成为故障或安全损害的牺牲品，则数字证书可能不可用。

[0148] 在发布加密密钥之后的任何时间，加密引擎 220 可以有利地采用上述的数据拆分处理 800 以使得加密密钥被拆分成独立不可破译的随机化的数。类似于认证数据，在步骤 965，加密引擎 220 将随机化的数传送到数据存储设备 D1 至 D4。

[0149] 本领域技术人员从此处公开应理解，用户可以在注册之后的任何时间请求数字证书。此外，系统之间的通信可以有利地包括全 SSL 或公钥加密技术。此外，注册处理可以发布来自多个证书颁发机构的多个数字证书，所述证书颁发机构包括信任引擎 110 内部或外部的一个或多个专有证书颁发机构。

[0150] 如在步骤 935 至 960 中所公开的，本发明的一个实施例包括最终存储在信任引擎 110 上的证书请求。根据一个实施例，因为加密处理模块 625 发布由信任引擎 110 使用的密钥，因此每个证书对应于一个私钥。因此，信任引擎 110 可以有利地通过监视由用户拥有或与用户相关联的证书来提供互用性。例如，当加密引擎 220 接收加密功能请求时，加密处理模块 625 可以调查由进行请求的用户所拥有的证书以确定该用户是否拥有与该请求的属性相匹配的私钥。当存在这样的证书时，加密处理模块 625 可以使用证书或与其相关的公钥或私钥，来执行所请求的功能。当这样的证书不存在时，加密处理模块 625 可以有利且透明地执行若干动作以试图对缺少适当的密钥进行补救。例如，图 9B 示出了根据本发明的实施例的多个方面的互用性处理 970 的流程图，公开上述步骤以保证加密处理模块 625 使用适当密钥来执行加密功能。

[0151] 如图 9B 中所示，互用性处理 970 从步骤 972 开始，在该处，加密处理模块 925 确定期望的证书类型。根据本发明的一个实施例，证书的类型可以有利地在加密功能请求或请求者提供的其他数据中指定。根据另一实施例，证书类型可以由请求的数据格式确定。例如，加密处理模块 925 可以有利地识别该请求对应于特定的类型。

[0152] 根据一个实施例，证书类型可包括一个或多个算法标准，例如，RSA、ELGAMAL 等。此外，证书类型可以包括一个或多个密钥类型，例如对称密钥、公钥、诸如 256 位密钥的强加密密钥、低安全密钥等。此外，证书类型可以包括一个或多个上述算法标准或密钥、一个或多个消息或数据格式、一个或多个数据封装或编码机制（例如 Base 32 或 Base 64）的升级或替换。证书类型还可以包括与一个或多个第三方加密应用或接口、一个或多个通信协议、或一个或多个证书标准或协议的兼容性。本领域的技术人员从在此的公开应该理解其他差异可能存在于证书类型中，以及如在此所公开的，可以执行这些差异之间的来回转换。

[0153] 一旦加密处理模块 625 确定证书类型，互用性处理 970 进行到步骤 974，并确定用户是否拥有与在步骤 974 中确定的类型相匹配的证书。当用户拥有匹配证书时，例如，信任引擎 110 可以通过例如其先前的存储获得匹配证书，加密处理模块 825 知道匹配私钥也存储在信任引擎 110 中。例如，匹配私钥可以被存储在仓库 210 或仓库系统 700 中。加密处理模块 625 可以有利地请求从例如仓库 210 组装匹配私钥，然后在步骤 976，使用匹配私钥执行加密动作或功能。例如，如上所述，加密处理模块 625 可以有利地执行散列、散列比较、数据加密或解密、数字签名校验或创建、等等。

[0154] 当用户不拥有匹配证书时，互用性处理 970 进行到步骤 978，在这里，加密处理模块 625 确定用户是否拥有交叉证明证书。根据一个实施例，证书颁发机构之间的交叉证明在第一证书颁发机构确定信任来自第二证书颁发机构的证书时发生。换句话说，第一证书颁发机构确定来自第二证书颁发机构的证书符合一定质量标准，并因此可以被“证明”为等同于第一证书颁发机构本身的证书。交叉证明在证书颁发机构发布例如具有信任等级的证书时变得更复杂。例如，第一证书颁发机构通常基于注册处理中的可靠程度，可以为特定证书提供三个信任等级，而第二证书颁发机构可以提供七个信任等级。交叉证明可以有利地跟踪来自第二证书颁发机构的哪些等级以及哪些证书可以代替来自第一证书颁发机构的哪些等级和哪些证书。当在两个证书颁发机构之间官方和公开地完成上述交叉证明时，彼此的证书和等级的映射通常被称作“链接 (chaining)”。

[0155] 根据本发明的另一实施例，加密处理模块 625 可以有利地开发由证书颁发机构认为一致的那些之外的交叉证明。例如，加密处理模块 625 可以访问第一证书颁发机构的证书实践声明 (certificatepractice statement，CPS) 或其他公开的政策声明，并且使用例如特定信任等级所需的认证令牌使第一证书颁发机构的证书匹配至另一证书颁发机构的那些证书。

[0156] 当在步骤 978 中加密处理模块 625 确定用户拥有交叉证明的证书时，互用性处理 970 进行到步骤 976，并使用交叉证明的公钥、私钥或两者来执行加密动作或功能。可替换地，当加密处理模块 625 确定用户不拥有交叉证明证书时，互用性处理 970 进行到步骤 980，在这里，加密处理模块 625 选择发布所请求的证书类型或被交叉证明的证书的证书颁发机构。在步骤 982，加密处理模块 625 确定前面所讨论的用户注册认证数据是否符合所选证书颁发机构的认证要求。例如，如果用户是通过例如回答人口统计学和其他问题在网络上注册的，所提供的认证数据与提供生物识别数据和呈现在例如公证人的第三方面前的用户相比可以建立较低的信任等级。根据一个实施例，上述认证要求可以有利地设置在所选证书颁发机构的 CPS 中。

[0157] 当用户已经向信任引擎 110 提供符合所选证书颁发机构的要求的注册认证数据时，互用性处理 970 进行到步骤 984，在这里，加密处理模块 825 从所选证书颁发机构获取证书。根据一个实施例，加密处理模块 625 通过遵照注册处理 900 的步骤 945 至 960 来获取证书。例如，加密处理模块 625 可以有利地使用来自加密引擎 220 已经可用的一个或多个密钥对的一个或多个公钥来从证书颁发机构请求证书。根据另一实施例，加密处理模块 625 可以有利地生成一个或多个新的密钥对，以及使用与其对应的公钥来从证书颁发机构请求证书。

[0158] 根据另一实施例，信任引擎 110 可以有利地包括能够发布一个或多个证书类型的一个或多个证书发布模块。根据该实施例，证书发布模块可以提供上述证书。当加密处理模块 625 获取到证书，互用性处理 970 进行到步骤 976，使用对应于所获取的证书的公钥、私钥或两者来执行加密动作或功能。

[0159] 当在步骤 982 中用户没有向信任引擎 110 提供满足所选证书颁发机构的要求的注册认证数据时，加密处理模块 625 在步骤 986 中确定是否存在具有不同认证要求的其他证书颁发机构。例如，加密处理模块 625 可以寻找具有较低认证要求的证书颁发机构，但仍然发布所选证书或其交叉证明。

[0160] 当存在上述具有较低要求的证书颁发机构时，互用性处理 970 进行到步骤 980 并选择该证书颁发机构。可替换地，当没有这样的证书颁发机构存在时，在步骤 988，信任引擎 110 可以要求来自用户的其他认证令牌。例如，信任引擎 110 可以要求包括例如生物识别数据的新注册认证数据。同样，信任引擎 110 可以要求用户在可信第三方前出现，并提供适当的认证凭证，例如带着驾照、社会保障卡、银行卡、出生证明、军人 ID 等出现在公证人面前。当信任引擎 110 接收到更新的认证数据时，互用性处理 970 进行到步骤 984 并获取上述所选的证书。

[0161] 通过上述互用性处理 970，加密处理模块 625 有利地提供不同加密系统之间的无缝、透明的翻译和转换。本领域技术人员从在此的公开应该理解上述互用系统的优点和实施方式。例如，互用性处理 970 的上述步骤 986 可以有利地包括下面将更详细描述的信任仲裁的各个方面，其中证书颁发机构在可以特定情况下接受较低等级的交叉证明。此外，互用性处理 970 可以包括保证标准证书撤销之间的互用性，以及采用标准证书撤销，例如采用证书撤销列表 (CRL)、在线证书状态协议 (OCSP) 等。

[0162] 图 10 示出了根据本发明的实施例的多个方面的认证处理 1000 的数据流。根据一个实施例，认证处理 1000 包括从用户收集当前认证数据并将其与用户的注册认证数据进行比较。例如，认证处理 1000 在步骤 1005 处开始，在这里，用户期望与例如卖方执行交易。这样的交易可以包括例如选择购买选项、请求对卖方系统 120 的受限制区域或装置的访问。在步骤 1010，卖方向用户提供交易 ID 和认证请求。交易 ID 可以有利地包括 192 位的量，其具有与 128 位随机量级联的 32 位时间戳，或与 32 位特定于卖方的常量级联的“临时随机数 (nonce)”。这样的交易 ID 唯一地标识该交易，从而模仿交易能够被信任引擎 110 拒绝。

[0163] 认证请求可以有利地包括特定交易所需的认证等级。例如卖方可以在发布时指定该交易所需的特定可信度。如果不能如下所述地对该可信度进行认证，则除非有用户提升可信度的进一步认证或卖方和服务器之间的认证改变，否则不能发生该交易。这些发布将在下面详细讨论。

[0164] 根据一个实施例，交易 ID 和认证请求可以有利地由卖方端小程序或其他软件程序生成。此外，交易 ID 和认证数据的传输可以包括使用传统 SSL 技术（例如 1/2SSL、或换句话说，卖方端认证的 SSL）加密的一个或多个 XML 文档。

[0165] 在用户系统 105 接收到交易 ID 和认证请求之后，用户系统 105 收集来自用户的当前认证数据，可能包括当前生物识别信息。用户系统 105 在步骤 1015 使用认证引擎 215 的公钥来至少加密当前认证数据“B”和交易 ID，以及将该数据传输至信任引擎 110。该传输优选地包括至少使用传统 1/2SSL 技术加密的 XML 文档。在步骤 1020，交易引擎 205 接收该传输，优选地识别 URL 或 URI 中的数据格式或请求，并将该传输转发至认证引擎 215。

[0166] 在步骤 1015 和 1020 中，卖方系统 120 在步骤 1025 使用优选的全 SSL 技术转发交易 ID 和认证请求至信任引擎 110。该通信还可以包括卖方 ID，尽管卖方标识还可以通过交易 ID 的非随机部分被传送。在步骤 1030 和 1035，交易引擎 205 接收该通信，在审计跟踪中创建记录，以及生成对将从数据存储设备 D1 至 D4 组装的用户的注册认证数据的请求。在步骤 1040，仓库系统 700 将对应于用户的注册认证数据的各部分传输至认证引擎 215。在步骤 1045，认证引擎 215 使用其私钥解密该传输，并将注册认证数据与用户提供的当前认

证数据进行比较。

[0167] 步骤 1045 的比较可以优选地应用试探式上下文相关认证，如上面提及并将在下面详细讨论的。例如，如果所接收的生物识别信息没有完美地匹配，则导致较低的信任匹配。在特定实施例中，认证的可信度与交易的性质和用户与卖方的期望被权衡考虑。这也将在下面更详细讨论。

[0168] 在步骤 1050，认证引擎 215 使用步骤 1045 的比较结果填充认证请求。根据本发明的一个实施例，认证请求被填充有认证处理 1000 的是 / 否或真服结果。在步骤 1055，被填充的认证请求返回到卖方供卖方采取行动，例如允许用户完成发起过认证请求的交易。根据一个实施例，确认消息被传递至用户。

[0169] 基于上面的描述，认证处理 1000 优选地保持敏感数据的安全性，并产生用于维护敏感数据完整性的结果。例如，敏感数据仅在认证引擎 215 中被组装。例如，注册认证数据不可破译，直到其在认证引擎 215 中被数据组装模块组装，并且当前认证数据不可破译，直到其被传统 SSL 技术和认证引擎 215 的私钥解包装。此外，传输至卖方的认证结果不包括敏感数据，用户可能甚至不知道他或她是否产生了有效的认证数据。

[0170] 尽管参考优选和可替换实施例公开了认证处理 1000，本发明不因此被限制。相反，本领域技术人员应该从在此的公开理解认证处理 1000 的多种替换例。例如，卖方可以有利地由几乎任何请求应用来代替，甚至是那些与用户系统 105 一起驻留的应用。例如，客户端应用，诸如 Microsoft Word，可以使用应用程序接口 (API) 或加密 API(CAPI) 来在解锁文件之前请求认证。可替换地，邮件服务器、网络、蜂窝电话、个人或移动计算装置、网络工作站等都可以发出能够被认证处理 1000 填充的认证请求。事实上，在提供上述可信任的认证处理 1000 之后，请求应用或装置可以提供对多个电子或计算机装置或系统的访问和使用。

[0171] 此外，认证处理 1000 可以在认证失败时使用多种替换过程。例如，认证失败可以保持相同的交易 ID 并请求用户重新输入他或她的当前认证数据。如上所述，使用相同的交易 ID 使得认证引擎 215 的比较器能够监视和限制对特定交易的认证尝试的次数，从而创建更安全的加密系统 100。

[0172] 此外，认证处理 1000 可以有利地被使用来开发简洁的单一登录解决方案 (single sign-on solution)，例如解锁敏感数据资料库。例如，成功或肯定认证可以为被认证的用户提供自动访问几乎无限多个系统和应用的任何数量的口令的能力。例如，用户的认证可以为用户提供对与多个在线卖方、局域网、各种个人计算装置、因特网业务提供商、拍卖商、投资经纪等相关联的口令、登录、金融凭证等的访问。通过使用敏感数据资料库，用户可以选择非常大且随机的口令，因为不再需要通过联想来记住它们。相反，认证处理 1000 提供对其的访问。例如，用户可以选择长度为 20 多位的随机混合符号串，而不是与可记住的数据、姓名等有关的东西。

[0173] 根据一个实施例，与给定用户相关的敏感数据资料库可以有利地存储在仓库 210 的数据存储设备中，或在仓库系统 700 中被拆分和存储。根据该实施例，在肯定的用户认证之后，信任引擎 110 向进行请求的应用提供所请求的敏感数据，例如适当的口令。根据另一实施例，信任引擎 110 可以包括用于存储敏感数据资料库的单独系统。例如，信任引擎 110 可以包括执行数据资料库功能并表现为驻留在上述信任引擎 110 的前端安全系统“之后”的独立软件引擎。根据该实施例，软件引擎在该软件引擎接收到来自信任引擎 110 的表示

肯定用户认证的信号之后提供所请求的敏感数据。

[0174] 在另一实施例中，数据资料库可以由第三方系统实现。类似于软件引擎实施例，第三方系统可以有利地在第三方系统从信任引擎 110 接收表示肯定用户认证的信号之后提供所请求的敏感数据。根据另一实施例，数据资料库可以在用户系统 105 上实现。用户端软件引擎可以有利地在接收到来自信任引擎 110 的表示肯定用户认证的信号之后提供前述数据。

[0175] 尽管参考可替换实施例公开了前述数据资料库，但是本领域技术人员应该从在此的公开理解多种其他实施方式。例如，特定数据资料库可以包括前述实施例中的部分或所有实施例的多个方面。此外，任何前述数据资料库可在不同时间使用一个或多个认证请求。例如，任何数据资料库可以要求每一个或多个交易、周期性地、每一个或多个会话、每访问一个或多个网页或网站、以一个或多个其他指定间隔、等等，进行认证。

[0176] 图 11 示出了根据本发明的实施例的多个方面的签名处理 1100 的数据流。如图 11 所示，签名处理 1100 包括类似于前面参考图 10 所述的认证处理 1000 的步骤。根据本发明的一个实施例，如下面将进一步具体讨论的，签名处理 1100 首先认证用户，然后执行若干数字签名功能中的一个或多个。根据另一实施例，签名处理 1100 可以有利地存储与其相关的数据，诸如消息或文件等的散列。该数据可以有利地被用在审计或任何其他事件中，例如在参与方企图抵赖交易时。

[0177] 如图 11 中所示，在认证步骤期间，用户和卖方可以有利地对诸如合同之类的消息达成一致。在签名过程中，签名处理 1100 有利地保证由用户签署的合同与卖方提供的合同相同。因此，根据一个实施例，在认证期间，卖方和用户在传输至认证引擎 215 的数据中包括他们各自的消息或合同副本的散列。通过仅使用消息或合同的散列，信任引擎 110 可以有利地存储显著减少的数据量，从而提供更高效以及节省成本的加密系统。此外，所存储的散列可以有利地与尚存疑的文件的散列进行比较，以确定该尚存疑的文件是否与由任何一方签名的文件匹配。这种确定文件是否与和交易相关的一个文件相同的能力提供了附加的证据，其能够用于反对一方抵赖交易的主张。

[0178] 在步骤 1103，认证引擎 215 组装注册认证数据并将其与由用户提供的当前认证数据进行比较。当认证引擎 215 的比较器指示注册认证数据匹配当前认证数据时，认证引擎 215 的比较器还将由卖方提供的消息的散列与由用户提供的消息的散列进行比较。因此，认证引擎 215 有利地保证用户同意的消息与卖方同意的消息相同。

[0179] 在步骤 1105，认证引擎 215 传输数字签名至加密引擎 220。根据本发明的一个实施例，该请求包括消息或合同的散列。然而，本领域技术人员从此处的公开应了解加密引擎 220 实际上可以加密任何类型的数据，包括但不限于视频、音频、生物识别信息、图像、或文本，以形成期望的数字签名。返回到步骤 1105，数字签名请求优选地包括通过传统 SSL 技术传送的 XML 文档。

[0180] 在步骤 1110 中，认证引擎 215 传输请求至每个数据存储设备 D1 至 D4，从而每个数据存储设备 D1 至 D4 传输与签名方相对应的加密密钥（一个或多个）中它们各自的部分。根据另一实施例，加密引擎 220 使用上面所述的互用性处理 970 的部分或所有步骤，从而加密引擎 220 首先确定要从仓库 210 或仓库系统 700 请求的用于签名方的一个或多个适当密钥，以及采取行动来提供适当的匹配密钥。根据另一实施例，认证引擎 215 或加密引擎 220

可以有利地请求与签名方相关联并存储在仓库 210 或仓库系统 700 中的一个或多个密钥。

[0181] 根据一个实施例，签名方包括用户和卖方中的一个或两者。在这样的情况下，认证引擎 215 有利地请求对应于用户和 / 或卖方的加密密钥。根据另一实施例，签名方包括信任引擎 110。在该实施例中，信任引擎 110 证明认证处理 1000 正确地认证了用户、卖方、或两者。因此，认证引擎 215 请求信任引擎 110 的加密密钥，例如属于加密引擎 220 的密钥，以执行数字签名。根据另一实施例，信任引擎 110 执行类似数字公证的功能。在该实施例中，签名方包括用户、卖方、或两者，连同信任引擎 110。因此，信任引擎 110 提供用户和 / 或卖方的数字签名，然后使用其本身的数字签名表示用户和 / 或卖方已经被正确认证。在该实施例中，认证引擎 215 可以有利地请求组装对应于用户、卖方或两者的加密密钥。根据另一实施例，认证引擎 215 可以有利地请求组装对应于信任引擎 110 的加密密钥。

[0182] 根据另一实施例，信任引擎 110 执行类似委托书的功能。例如，信任引擎 110 可以以第三方的名义数字签名该消息。在这种情况下，认证引擎 215 请求与第三方相关联的加密密钥。根据该实施例，在允许类似委托书的功能之前，签名处理 1100 可以有利地包括第三方的认证。此外，认证处理 1000 可以包括检查第三方约束，例如指示何时以及在什么情况下可以使用特定第三方签名的商业逻辑等。

[0183] 基于上面所述，在步骤 1110，认证引擎从数据存储设备 D1 至 D4 请求对应于签名方的加密密钥。在步骤 1115 中，数据存储设备 D1 至 D4 传输与签名方相对应的加密密钥中它们各自的部分至加密引擎 220。根据一个实施例，上述传输包括 SSL 技术。根据另一实施例，上述传输可以有利地使用加密引擎 220 的公钥被超级加密 (super-encrypt)。

[0184] 在步骤 1120，加密引擎 220 组装签名方的上述加密密钥并以其加密该消息，从而形成数字签名（一个或多个）。在签名处理 1100 的步骤 1125，加密引擎 220 传输数字签名至认证引擎 215。在步骤 1130，认证引擎 215 传输填充的认证请求连同散列的消息的副本以及数字签名至交易引擎 205。在步骤 1135，交易引擎 205 传输包括交易 ID、关于认证是否成功的指示和数字签名的收据 (receipt) 至卖方。根据一个实施例，上述传输可以有利地包括信任引擎 110 的数字签名。例如，信任引擎 110 可以使用其私钥加密收据的散列，从而形成将被附加到至卖方的传输的数字签名。

[0185] 根据一个实施例，交易引擎 205 还传输确认消息至用户。尽管参考其优选和可替换实施例公开了签名处理 1100，但是本发明不因此被限制。相反，本领域技术人员应该从此处的公开了解签名处理 1100 的多种替换例。例如，卖方可以由用户应用（例如电子邮件应用）来代替。例如，用户可能希望用他或她的数字签名来数字签名特定电子邮件。在这样的实施例中，通过签名处理 1100 的传输可以有利地仅包括消息的散列的一份副本。此外，本领域技术人员应该从此处的公开理解多种客户端应用可以请求数字签名。例如，客户端应用可以包括文字处理器、电子表格、电子邮件、语音邮件、对受限系统区域的访问，等等。

[0186] 此外，本领域技术人员应该从此处的公开理解签名处理 1100 的步骤 1105 至 1120 可以有利地使用图 9B 的互用性处理 970 的部分或全部步骤，从而提供不同加密系统之间的互用性，其中不同加密系统可能例如需要处理不同签名类型的数字签名。

[0187] 图 12 示出了根据本发明的一个实施例的多个方面的加密 / 解密处理 1200 的数据流。如图 12 中所示，解密处理 1200 从使用认证处理 1000 认证用户开始。根据一个实施例，认证处理 1000 包括认证请求中的同步会话密钥。例如，在传统 PKI 技术中，本领域技术人

员应该理解使用公钥和私钥加密或解密数据是数学密集的，并且可能需要相当多的系统资源。然而，在对称密钥加密系统中或在消息的发送者或接收者共享用于加密和解密消息的单个共同密钥的系统中，数学运算要简单和快速得多。因此，在传统 PKI 技术中，消息的发送者将生成同步会话密钥，并使用较为简单且较快速的同步密钥系统加密该消息。然后，发送者将使用接收者的公钥加密会话密钥。加密的会话密钥将附加到同步加密的消息，并且两个数据都被发送至接收者。接收者使用他或她的私钥来解密该会话密钥，然后使用该会话密钥来解密该消息。基于上面所述，较简单且较快速的对称密钥系统被用于大部分的加密 / 解密处理。因此，在解密处理 1200 中，解密有利地假设同步密钥已经用用户的公钥被加密。因此，如上所述，加密的会话密钥被包括在认证请求中。

[0188] 返回到解密处理 1200，在用户已经在步骤 1205 中被认证之后，认证引擎 215 将加密的会话密钥转发至加密引擎 220。在步骤 1210，认证引擎 215 将请求转发至每个数据存储设备 D1 至 D4，请求用户的加密密钥数据。在步骤 1215，每个数据存储设备 D1 至 D4 传输其各自的加密密钥部分至加密引擎 220。根据一个实施例，上述传输使用加密引擎 220 的公钥被加密。

[0189] 在解密处理 1200 的步骤 1220，加密引擎 220 组装加密密钥并以其解密会话密钥。在步骤 1225，加密引擎转发会话密钥至认证引擎 215。在步骤 1227，认证引擎 215 填充认证请求以包括解密的会话密钥，并将填充的认证请求传输至交易引擎 205。在步骤 1230，交易引擎 205 将认证请求连同会话密钥转发至进行请求的应用或卖方。然后，根据一个实施例，进行请求的应用或卖方使用该会话密钥来解密被加密的消息。

[0190] 尽管参考其优选和可替换实施例公开了解密处理 1200，本领域技术人员从此处的公开应该了解解密处理 1200 的多种替换例。例如，解密处理 1200 可以位于同步密钥加密之前并且依赖于全公钥技术。在这样的实施例中，进行请求的应用可以传输整个消息至加密引擎 220，或可以使用一些类型的压缩或可逆散列以传输消息至加密引擎 220。本领域技术人员从此处的公开也应该理解上述通信可以有利地包括以 SSL 技术包装的 XML 文档。

[0191] 加密 / 解密处理 1200 还提供文档或其他数据的加密。因此，在步骤 1235，进行请求的应用或卖方可以有利地传输对用户公钥的请求至信任引擎 110 的交易引擎 205。该进行请求的应用或卖方进行该请求，是因为进行请求的应用或卖方使用用户的公钥来例如加密将被用来加密文档或消息的会话密钥。如在注册处理 900 中所述，交易引擎 205 将用户的数字证书的副本存储在例如大容量存储器 225 中。因此，在加密处理 1200 的步骤 1240，交易引擎 205 从大容量存储器 225 请求用户的数字证书。在步骤 1245，大容量存储器 225 将对应于该用户的数字证书传输至交易引擎 205。在步骤 1250，交易引擎 205 将数字证书传输至进行请求的应用或卖方。根据一个实施例，加密处理 1200 的加密部分不包括用户的认证。这是因为进行请求的卖方仅需要用户的公钥，并且不请求任何敏感数据。

[0192] 本领域技术人员从此处的公开应理解，如果特定用户不具有数字证书，则信任引擎 110 可以使用注册处理 900 的部分或全部来生成用于该特定用户的数字证书。然后，信任引擎 110 可以启动加密 / 解密处理 1200，从而提供适当的数字证书。此外，本领域的技术人员从此处的公开应该理解，加密 / 解密处理 1200 的步骤 1220 和 1235 至 1250 可以有利地使用图 9B 的互用性处理的部分或全部步骤，从而提供在可能例如需要处理加密的不同加密系统之间的互用性。

[0193] 图 13 示出了根据本发明的另一实施例的多个方面的信任引擎系统 1300 的简化框图。如图 13 中所示，信任引擎系统 1300 包括多个不同的信任引擎 1305、1310、1315 和 1320。为了有利于更全面理解本发明，图 13 示出了每个信任引擎 1305、1310、1315 和 1320 具有交易引擎、仓库和认证引擎。然而，本领域技术人员应该理解，每个交易引擎可以有利地包括参考图 1-8 公开的元件和通信信道的部分、组合或所有。例如，一个实施例可以有利地包括具有一个或多个交易引擎、多个仓库和多个加密服务器、或其任意组合的信任引擎。

[0194] 根据本发明的一个实施例，每个信任引擎 1305、1310、1315 和 1320 在地理上分开，从而例如信任引擎 1305 可以位于第一位置，信任引擎 1310 可以位于第二位置，信任引擎 1315 可以位于第三位置，而信任引擎 1320 可以位于第四位置。上述地理分开有利地减小了系统响应时间而增加了整个信任引擎系统 1300 的安全性。

[0195] 例如，当用户登录至加密系统 100 时，用户可能最靠近第一位置，并且可能期望被认证。如参考图 10 所述，为了被认证，用户提供当前认证数据，诸如生物识别信息等等，并且当前认证数据与该用户的注册认证数据进行比较。因此，根据一个示例，用户有利地提供当前认证数据至地理上最靠近的信任引擎 1305。信任引擎 1305 的交易引擎 1321 然后转发当前认证数据至同样位于第一位置的认证引擎 1322。根据另一实施例，交易引擎 1321 转发当前认证数据至信任引擎 1310、1315 或 1320 的一个或多个认证引擎。

[0196] 交易引擎 1321 还请求组装来自例如每个信任引擎 1305 至 1320 的仓库的注册认证数据。根据该实施例，每个仓库提供注册认证数据中它的部分至信任引擎 1305 的认证引擎 1322。认证引擎 1322 然后使用来自例如前两个响应的仓库的加密数据部分，并将注册认证数据组装成被破译的形式。认证引擎 1322 将注册认证数据与当前认证数据进行比较并返回认证结果至信任引擎 1305 的交易引擎 1321。

[0197] 基于上面所述，信任引擎系统 1300 使用多个地理上分开的信任引擎 1305 至 1320 中最近的一个来执行认证处理。根据本发明的一个实施例，将信息路由至最近的交易引擎可以有利地在运行在用户系统 105、卖方系统 120 或证书颁发机构 115 中的一个或多个上的客户端小程序处执行。根据一个可替换实施例，可以使用更复杂的判定处理来从信任引擎 1305 至 1320 中进行选择。例如，判定可以基于给定信任引擎的可用性、可操作性、连接的速度、负荷、性能、地理接近度、或其组合。

[0198] 以该方式，信任引擎系统 1300 降低其响应时间同时维持与地理上远离的数据存储设备相关联的安全性优点，例如参考图 7 所讨论的那些优点，其中在图 7 中每个数据存储设备存储敏感数据的随机化部分。例如，在例如信任引擎 1315 的仓库 1325 处的安全性损害不必然泄露信任引擎系统 1300 的敏感数据。这是因为仓库 1325 仅包括不可破译的随机化的数据，该数据在没有更多数据的情况下是完全无用的。

[0199] 根据另一实施例，信任引擎系统 1300 可以有利地包括多个类似于认证引擎而布置的加密引擎。加密引擎可以有利地执行诸如参考图 1-8 所公开的加密功能。根据另一实施例，信任引擎系统 1300 可以有利地用多个加密引擎代替多个认证引擎，从而执行诸如参考图 1-8 所公开的加密功能。根据本发明的又一实施例，如上所述，信任引擎系统 1300 可以使用具有认证引擎、加密引擎或两者的部分或所有功能的引擎来代替每个多认证引擎。

[0200] 尽管参考其优选和可替换实施例公开了信任引擎系统 1300，本领域的技术人员应该理解，信任引擎系统 1300 可以包括部分的信任引擎 1305 至 1320。例如，信任引擎系统

1300 可以包括一个或多个交易引擎，一个或多个仓库、一个或多个认证引擎、或一个或多个加密引擎、或其组合。

[0201] 图 14 示出了根据本发明的另一实施例的多个方面的信任引擎系统 1400 的简化框图。如图 14 所示，信任引擎系统 1400 包括多个信任引擎 1405、1410、1415 和 1420。根据一个实施例，每个信任引擎 1405、1410、1415、和 1420 包括参考图 1-8 公开的信任引擎 110 的部分或全部元件。根据该实施例，当用户系统 105、卖方系统 120 或证书颁发机构 115 的客户端小程序与信任引擎系统 1400 通信时，这些通信被发送至每个信任引擎 1405 至 1420 的 IP 地址。此外，每个信任引擎 1405、1410、1415 和 1420 的每个交易引擎类似于参考图 13 公开的信任引擎 1305 的交易引擎 1321 而工作。例如，在认证处理过程中，每个信任引擎 1405、1410、1415 和 1420 的每个交易引擎传输当前认证数据至其各自的认证引擎，并传输请求以组装存储在每个信任引擎 1405 至 1420 的每个仓库中的随机化的数据。图 14 没有示出所有这些通信，因为这样示出的话就变得过于复杂了。继续认证处理，每个仓库然后将其随机化数据部分传送至每个信任引擎 1405 至 1420 的每个认证引擎。每个信任引擎的每个认证引擎使用其比较器来确定当前认证数据是否与每个信任引擎 1405 至 1420 的仓库提供的注册认证数据匹配。根据该实施例，每个认证引擎的比较结果然后被传输至其他三个信任引擎的冗余模块。例如，来自信任引擎 1405 的认证引擎的结果被传输至信任引擎 1410、1415 和 1420 的冗余模块。从而类似地，信任引擎 1405 的冗余模块接收来自信任引擎 1410、1415 和 1420 的认证引擎的结果。

[0202] 图 15 示出了图 14 的冗余模块的框图。该冗余模块包括比较器，用于接收来自三个认证引擎的认证结果以及将该结果传输至第四个信任引擎的交易引擎。比较器将来自这三个认证引擎的认证结果进行比较，如果有两个结果是一致的，则比较器断定认证结果与两个达成一致的认证引擎的认证结果匹配。该结果然后被传输回对应于与这三个认证引擎不相关联的信任引擎的交易引擎。

[0203] 基于上面所述，冗余模块基于从优选地地理上与该冗余模块的信任引擎相远离的认证引擎接收到的数据确定认证结果。通过提供这样的冗余功能，信任引擎系统 1400 保证信任引擎 1405 至 1420 之一的认证引擎的损害不足以损害该特定信任引擎的冗余模块的认证结果。本领域技术人员应该理解，信任引擎系统 1400 的冗余模块功能也可以被应用于每个信任引擎 1405 至 1420 的加密引擎。然而，这样的加密引擎通信没有在图 14 中示出以避免复杂。此外，本领域技术人员应该理解，用于图 15 的比较器的多种可替换的认证结果冲突解决算法适于用在本发明中。

[0204] 根据本发明的另一实施例，信任引擎系统 1400 可以有利地在加密比较步骤过程中采用冗余模块。例如，上述参考图 14 和 15 公开的冗余模块的部分或全部可以有利地在对一方或多方在特定交易期间提供的文档进行散列比较期间实施。

[0205] 尽管已经根据某些优选和可替换实施例描述了上述发明，但是本领域技术人员可以由此处公开显而易见其他实施例。例如，信任引擎 110 可以发布短期证书，而私有加密密钥被释放给用户某个预定时间段。例如，当前证书标准包括能够被设置为在预定时间量后过期的有效性字段。因此，信任引擎 110 可以向用户释放私钥，其中该私钥在例如 24 小时内有效。根据这样的实施例，信任引擎 110 可以有利地发布与特定用户相关联的新的加密密钥对，然后释放该新的加密密钥对的私钥。然后，一旦私有加密密钥被释放，信任引擎 110

就立即终止这样的私钥的任何内部有效使用，因为其信任引擎 110 不再保证其安全。

[0206] 此外，本领域技术人员应该意识到，加密系统 100 或信任引擎 110 可以包括识别任何类型的装置的能力，所述装置诸如但不限于膝上型电脑、蜂窝电话、网络、生物识别装置、等等。根据一个实施例，这样的识别可以来自于在对特定服务的请求中提供的数据，所述请求诸如对引起访问或使用的认证的请求、对加密功能的请求等。根据一个实施例，上述请求可以包括唯一装置标识符，例如处理器 ID。可替换地，该请求可以包括具有特定可识别数据格式的数据。例如，移动和卫星电话通常不包括对于完全 X509.V3 重加密证书 (full X509.V3 heavy encryption certificates) 的处理能力，从而不请求它们。根据该实施例，信任引擎 110 可以识别所呈现的数据格式的类型，并仅以同样方式响应。

[0207] 在上述系统的其他方面，上下文相关认证可以使用下面将描述的各种技术来提供。上下文相关认证，例如图 16 中所示的，提供了不仅评估在用户试图认证其本身时发送的实际数据，而且还评估在生成和传递该数据的周围的环境的能力。如下面将描述的，这样的技术也可以支持用户与信任引擎 110 之间或卖方与信任引擎 110 之间的特定于交易的信任仲裁。

[0208] 如上所讨论的，认证是证明用户是其所声称的那个人的过程。通常，认证要求向认证权力机构证实一些事实。本发明的信任引擎 110 代表用户必须向其认证自身的权力机构。用户必须通过知道一些仅该用户应知道的东西（基于知识的认证）、具有一些仅该用户应具有的东西（基于令牌的认证）、或是仅该用户应该是的东西（基于生物识别的认证）来向信任引擎 110 证实他是他所声称的那个人。

[0209] 基于知识的认证的示例包括但不限于口令、PIN 号、或锁组合。基于令牌的认证的示例包括但不限于房间钥匙、物理信用卡、驾照、或特定电话号码。基于生物识别的认证的示例包括但不限于指纹、笔迹分析、面部扫描、手扫描、耳朵扫描、虹膜扫描、血管模式、DNA、语音分析、或视网膜扫描。

[0210] 每种类型的认证都具有特定优点和缺点，并且每种类型提供不同的安全等级。例如，与偶然听到某个人的口令并重复该口令相比，通常较难创建与其他人的指纹匹配的假指纹。每种类型的认证还要求不同类型的数据是认证权力机构所知道的，以便校验使用这种认证形式的人。

[0211] 如在此所使用的，“认证”广义地表示证实某人的身份是他声称他是的那个人的整个过程。“认证技术”表示基于特定的知识、物理令牌、或生物识别读取的特定类型的认证。“认证数据”表示发送至认证权力机构或向其证实以建立身份的信息。“注册数据”表示初始提交给认证权力机构以建立用于与认证数据进行比较的基准的数据。“认证实例 (authentication instance)”表示与用认证技术进行认证的尝试相关联的数据。

[0212] 参考上面图 10 来描述在认证用户的处理中所涉及的内部协议和通信。该处理中发生上下文相关认证的部分出现在图 10 的步骤 1045 所示的比较步骤中。该步骤在认证引擎 215 中发生，并涉及组装从仓库 210 取回的注册数据 410 以及将其与用户提供的认证数据进行比较。该处理的一个特定实施例在图 16 中示出并在下面描述。

[0213] 由用户提供的当前认证数据和从仓库取回的注册数据在图 16 的步骤 1600 中被认证引擎 215 接收。这两个数据集合都可能包含与认证的分离技术有关的数据。在步骤 1605，认证引擎 215 分离与每个单独认证实例相关联的认证数据。这是必要的，从而认证数

据与用户的注册数据的适当子集进行比较（例如，指纹认证数据应该与指纹注册数据进行比较，而不是与口令注册数据进行比较）。

[0214] 通常，认证用户涉及一个或多个单独认证实例，取决于用户可用的认证技术。这些方法被注册过程中由用户提供的注册数据所限制（如果用户在注册时没有提供视网膜扫描，他就不能使用视网膜扫描来认证自己），以及被用户当前可用的手段所限制（例如，如果用户在他当前位置不具有指纹读取器，指纹认证将不可行）。在一些情况下，单个认证实例可能足以认证用户；然而在某些情况下，多个认证实例的组合可以被使用以更加确信地认证特定交易的用户。

[0215] 每个认证实例包括与一种特定认证技术（例如指纹、口令、智能卡等）以及在获取和传递用于该特定技术的数据周围的环境有关的数据。例如，尝试通过口令进行认证的特定实例不仅生成与口令本身有关的数据，还生成与该口令尝试有关的环境数据，被称为“元数据”。该环境数据包括诸如以下的信息：特定认证实例发生的时间、认证信息从其递送的网络地址、以及本领域技术人员所知的可以确定的关于认证数据的来源的任何其他信息（连接的类型、处理器序列号等）。

[0216] 在许多情况下，仅有少量的环境元数据可用。例如，如果用户位于使用代理或网络地址转换或其它掩盖发源计算机地址的技术的网络上，则仅仅可以确定代理服务器或路由器的地址。类似地，在许多情况下，诸如处理器序列号之类的信息将不可用，这是由于所使用的硬件或操作系统的限制、系统的操作者禁用这样的特征、或用户的系统与信任引擎 110 之间的连接的其他限制。

[0217] 如图 16 中所示，一旦在认证数据中表示的单独认证实例在步骤 1605 中被提取和分离，认证引擎 215 就评估每个实例在表示该用户是其所声称的那个人这方面的可靠性。单个认证实例的可靠性通常基于若干因素来确定。这些可以被成组为跟与认证技术相关联的可靠性有关的因素（其在步骤 1610 中被评估）和跟所提供的特定认证数据的可靠性有关的因素（其在步骤 1815 中被评估）。第一组包括但不限于所使用的认证技术的固有可靠性，以及与该方法一起使用的注册数据的可靠性。第二组包括但不限于注册数据与认证实例所提供的数据之间的匹配度以及与该认证实例相关联的元数据。这些因素中的每一个可以独立于其他因素而改变。

[0218] 认证技术的固有可靠性基于冒名顶替者提供其他人的正确数据有多困难，以及认证技术的整体错误率。对于基于口令和知识的认证方法，该可靠性通常相当低，因为不存在任何东西来阻止某人将其口令泄露给另一个人以及阻止该另一个人使用该口令。更复杂的基于知识的系统可能仅具有中等可靠性，这是因为知识可以相当容易地从一个人传到另一个人。基于令牌的认证，诸如具有正确的智能卡或使用特定终端来执行认证，类似地具有由其本身使用的低可靠性，这是因为不能保证正确的人持有该正确的令牌。

[0219] 然而，生物识别技术更固有地可靠，因为其通常难以方便地甚至故意地为其他人提供使用你的指纹的能力。因为破坏生物识别认证技术更加困难，所以生物识别方法的固有可靠性通常高于纯粹基于知识或令牌的认证技术的可靠性。然而，即使生物识别技术也可能具有一些产生错误接受或错误拒绝的情况。这些情况的发生可以由相同的生物识别技术的不同实施方式的不同可靠性反映出来。例如，由一个公司提供的指纹匹配系统可以提供比由另一公司提供的指纹匹配系统更高的可靠性，因为该公司使用更高质量的光学器件

或更好的扫描分辨率或一些其他的减少错误接受或错误拒绝的发生的改进。

[0220] 注意该可靠性可以以不同方式表示。可靠性被期望以某种能够被试探法 530 和认证引擎 215 的算法使用以计算每种认证的可信度的衡量标准来表示。表示这些可靠性的一种优选模式是百分比或分数。例如，指纹可以被赋予 97％的固有可靠性，而口令可能仅被赋予 50％的固有可靠性。本领域的技术人员应该理解这些特定值仅是示例性的，并且可以在具体实施方式之间改变。

[0221] 评估可靠性必须针对的第二个因素是注册的可靠性。这是上面提及的“分级注册”处理的一部分。该可靠性因素反映在初始注册处理期间提供的标识的可靠性。例如，如果个人以物理地将他们身份的证据出示给公证人或其他政府官员的方式初始注册，并且注册数据在此时被记录和公证，则该数据将比在注册过程中通过网络提供并且仅由数字签名或其他没有真实绑定至个人的信息担保的数据更可靠。

[0222] 具有不同可靠性等级的其他注册技术包括但不限于 ：在信任引擎 110 操作者的物理办公室注册 ；在用户的就业地点注册 ；在邮局或护照办公室注册 ；通过附属方或可信方向信任引擎 110 操作者注册 ；匿名或笔名注册，其中注册的身份尚不等同于特定的真实个人 ；以及本领域已知的这样的其他手段。

[0223] 这些因素反映信任引擎 110 和注册处理期间提供的标识的源之间的信任。例如，如果在提供身份证据的初始处理期间与雇主相关联地执行注册，则该信息可以被认为在公司内部极为可靠，但是可能被政府机构或竞争者认为是较低等级可信。因此，这些其他组织中每一个所操作的信任引擎可以给该注册分配不同的可靠性等级。

[0224] 类似地，通过网络提交但是用相同的信任引擎 110 由在先前注册过程中提供的其他信任数据认证的另外的数据可以被认为与原始注册数据一样可靠，即使后者数据是通过开放网络提交的。在这样的情况下，后续公证将有效地增加与原始注册数据相关联的可靠性等级。以该方式，例如，通过向一些注册官员证明个人身份与注册数据相匹配，匿名或笔名注册就可以上升为完全注册。

[0225] 上述可靠性因素通常是可以在任何特定认证实例之前确定的值。这是因为他们基于注册和技术而不是实际的认证。在一个实施例中，基于这些因素生成可靠性的步骤包括查找以前确定的用于该特定认证技术和用户注册数据的值。在本发明的一个有利实施例的另一方面，这样的可靠性可以包括在注册数据本身中。以该方式，这些因素连同从仓库 210 发送的注册数据一起被自动传递至认证引擎 215。

[0226] 尽管这些因素通常可以在任何单个认证实例之前确定，但是他们仍然对为用户使用特定认证技术的每个认证实例有影响。此外，尽管这些值可能随着时间改变（例如，如果用户以更可靠的方式重新注册），但是他们不取决于认证数据本身。相反，与单个特定实例的数据相关联的可靠性因素可能随每个情况而改变。如下所讨论，对于每个新认证，都必须评估这些因素，从而在步骤 1815 生成可靠性分数。

[0227] 认证数据的可靠性反映了由用户在特定认证实例中提供的数据与在认证注册期间提供的数据之间的匹配。这是认证数据是否匹配于用户声称是其的个人的注册数据的基本问题。通常，当数据不匹配时，用户被认为是没有被成功认证，并且认证失败。被评估的方式可以根据所使用的认证技术而改变。这种数据的比较是由图 5 中所示的认证引擎 215 的比较器 515 功能来实现的。

[0228] 例如，口令的匹配通常以二元 (binary) 形式评估。换句话说，口令或者是完美匹配，或者是失败匹配。通常不期望接受即使是部分匹配，即口令接近正确口令但不是完全正确。因此，当评估口令认证时，由比较器 515 返回的认证的可靠性通常是 100％（正确）或 0％（错误），而没有中间值的可能。

[0229] 与用于口令的规则相类似的规则通常被应用于基于令牌的认证方法，例如智能卡。这是因为拥有具有类似标识符的智能卡或类似于正确智能卡的智能卡，仍然如同拥有任何其他不正确令牌一样是错误的。因此，令牌也趋向于是二元认证：用户或者具有正确令牌，或者不具有。

[0230] 然而，某些类型的认证数据，诸如问卷和生物识别，通常不是二元认证。例如，指纹可以与参考指纹具有不同程度的匹配。某种程度上，这可能是由于在初始注册或在后续认证过程中获取的数据质量的变化。（指纹可能被弄脏或人可能在特定手指上具有静态愈合的伤疤或烧伤）。在其他情况下，数据可能匹配得不够完美，这是因为信息本身在某种程度上是可变的并且是基于图案匹配。（由于背景噪声、或者录制语音的环境的音响效果、或者由于该人感冒了，而导致语音分析可能像是接近但不完全正确）。最后，在大量数据被比较的情形中，情况可能是大部分数据匹配很好而一些匹配不好。（十个问题的问卷可能导致个人问题中八个正确答案，但是两个不正确答案）。由于这些原因中的任一个，注册数据和用于特定认证实例的数据之间的匹配可能期望由比较器 515 赋予一个部分匹配值。以该方式，例如，可以假定指纹 85％匹配，声纹 65％匹配，以及问卷 80％匹配。

[0231] 由比较器 515 产生的测量结果（匹配度）是表示认证是否正确这一基本问题的因素。然而，如上所述，这仅是可用于确定给定认证实例的可靠性的多个因素之一。还注意，即使能够确定某种部分程度的匹配，最终还是期望基于部分匹配来提供二元结果。在一种可替换的操作模式中，也可以基于匹配度是否通过特定的阈值匹配水平将部分匹配当做二元的，即，完美匹配 (100％）或失败匹配 (0％）。这样的处理可以被用来为否则将产生部分匹配的系统提供简单的通过 / 失败匹配等级。

[0232] 在评估给定认证实例的可靠性时考虑的另一因素是关于提供用于该特定实例的认证数据的环境。如上所述，环境指的是与特定认证实例相关联的元数据。这可包括但不限于这样的信息：认证者的网络地址，到其能够被确定的程度；认证的时间；认证数据的传输模式（电话线、蜂窝网络等）；以及认证者的系统的序列号。

[0233] 这些因素可以被用来产生通常由用户请求的认证的类型的简档 (profile)。然后，该信息可以被用来以至少两种方式评估可靠性。一种方式是考虑用户是否正以与该用户的认证的正常简档相一致的方式请求认证。如果用户正常情况下在工作日（当其工作时）从一个网络地址进行认证请求而在晚间或周末（当其在家时）从一不同的网络地址进行认证请求，那么在工作日期间从家庭地址发生的认证就不那么可靠，因为这是在正常认证简档之外的。类似地，如果用户正常情况下使用指纹生物识别并且在晚间进行认证，则在白天仅使用口令发起的认证就不那么可靠。

[0234] 环境元数据可以被用来评估认证实例的可靠性的另一方法是确定环境提供多少证据来证明认证者就是其所声称的个人。例如，如果认证来自于具有已知是与用户相关联的序列号的系统，则其是良好的环境指示器，指示用户是其声明的用户。相反，如果认证来源于已知在洛杉矶的网络地址而已知用户居住在伦敦时，这就指示，基于其环境，该认证是

不可靠的。

[0235] Cookie 或其他电子数据也可能被放置在当用户与卖方系统或信任引擎 110 交互时由用户使用的系统上。该数据被写入用户的系统的存储器，并且可以包括可以由用户系统上的网络浏览器或其他软件读取的标识。如果允许该数据在会话之间驻留在用户系统上（“持续的 cookie”），其可以在认证特定用户期间作为过去使用过该系统的进一步证据与认证数据一起被发送。事实上，给定实例的元数据，特别是持续的 cookie，其本身可以形成一种基于令牌的认证者。

[0236] 一旦基于认证实例的技术和数据的适当可靠性因素如在上面步骤 1610 和 1615 中分别描述的那样被生成，它们就被用来产生在步骤 1620 中提供的认证实例的总可靠性。实现这个的一种方式是简单地将每个可靠性表示为百分数，然后将它们相乘。

[0237] 例如，假设认证数据是从已知是完全符合用户过去的认证简档的用户家庭计算机的网络地址发送的 (100％)，并且所使用的技术是指纹识别 (97％)，并且初始指纹数据是通过用户的雇主使用信任引擎 110 登记的 (90％)，并且认证数据和注册数据中的原始指纹模板之间的匹配非常好 (99％)。则该认证实例的总可靠性可以被计算为这些可靠性的乘积：100％＊ 97％＊ 90％＊ 99％——86.4％可靠性。

[0238] 这样计算的可靠性表示一个单个认证实例的可靠性。单个认证实例的总可靠性还可以使用不同地对待不同可靠性因素的技术来计算，例如，通过使用将不同权重分配给每个可靠性因素的公式。此外，本领域的技术人员将意识到所使用的实际值可以表示百分比之外的值，并且可以使用非算术系统。一个实施例可以包括由认证请求者使用以便为每个因素设置权重的模块，以及用于建立认证实例的总可靠性的算法。

[0239] 认证引擎 215 可以使用上述技术及其变型来确定单个认证实例的可靠性，如步骤 1620 所示。然而，在许多认证情况下，同时提供多个认证实例可能是有用的。例如，当尝试使用本发明的系统认证自身时，用户可以提供用户标识、指纹认证数据、智能卡以及口令。在这样情况下，三个独立的认证实例被提供至信任引擎 110 供评估。进行到步骤 1625，如果认证引擎 215 确定由用户提供的数据包括多于一个认证实例，则每个实例依次如在步骤 1630 中所示的被选择以及如上面步骤 1610、1615 和 1620 中所述的被评估。

[0240] 注意，所讨论的许多可靠性因素因实例而异。例如，这些技术的固有可靠性很可能不同，在认证数据和注册数据之间提供的匹配度也可能不同。此外，用户可能在不同时间和在不同环境下为这些技术中的每一种技术提供了注册数据，这为这些实例中的每个实例提供不同的注册可靠性。最后，即使这些实例中的每个实例被提交的环境是相同的，这些技术的使用可能每个都不同地适合用户的简档，因此可以被赋予不同的环境可靠性。（例如，用户可能在正常情况下使用其口令和指纹，而不是其智能卡）。

[0241] 因此，这些认证实例中的每个实例的最终可靠性可能彼此不同。然而，通过一起使用多个实例，认证的总可信度趋向于增加。

[0242] 一旦认证引擎已经对认证数据中提供的所有认证实例执行了步骤 1610 至 1620，每个实例的可靠性被用在步骤 1635 中以评估总的认证可信度。将单个认证实例的可靠性合并成认证可信度的处理可以用各种将所产生的单独可靠性联系起来的方法来建模，也可以处理这些认证技术中的一些认证技术之间的特定相互作用。（例如，诸如多个口令之类的多个基于知识的系统产生的可信度可以小于单个口令和甚至相当弱的生物识别（诸如基

本语音分析）的可信度。）

[0243] 认证引擎 215 可以合并多个同时存在的认证实例的可靠性以生成最终可信度的一种方式是，将每个实例的不可靠性相乘以得到总的不可靠性。不可靠性通常是可靠性的互补百分比。例如，可靠性为 84％的技术的不可靠性是 16％。产生 86％、75％和 72％的可靠性的上面所述三个认证实例（指纹、智能卡、口令）分别具有对应的不可靠性 (100-86)％、(100-75)％和 (100-72)％，或 14％、25％和 28％。通过将这些不可靠性相乘，我们得到累积不可靠性 14％＊ 25％＊ 28％——0.98％的不可靠性，对应于 99.02％的可靠性。

[0244] 在另一操作模式中，另外的因素和试探法 530 可以被应用在认证引擎 215 中以解释各种认证技术的相互依赖。例如，如果有人已经未经授权地访问了特定家庭计算机，则他们可能也能够访问该地址的电话线。因此，既基于发起电话号码也基于认证系统序列号的认证不会对认证的总可信度增加很多。然而，基于知识的认证大大地独立于基于令牌的认证（即，如果有人窃取了你的蜂窝电话号码或密钥，如果他们不曾知道你的 PIN 或口令，他们就不再可能知道）。

[0245] 此外，不同的卖方或其他认证请求者可能希望为认证的不同方面不同地加权。这可以包括使用不同的加权因子或用于计算单个实例的可靠性的算法，以及使用不同方式来评估具有多个实例的认证事件。

[0246] 例如，某些类型的交易（例如公司电子邮件系统）的卖方可能期望主要基于试探法和其他默认的环境数据来进行认证。因此，他们可以对跟元数据和其他与认证事件周围的环境相关联的关于简档的信息有关的因素应用高权重。由于除了在工作时间期间登录到正确机器之外不向用户要求别的，因此这种安排可以用来减轻用户在正常操作时间期间的负担。然而，另一卖方可能由于策略决定而给予来自特定技术的认证（例如指纹匹配）以最重的权重，其中该策略决定是，这种技术对于该特定卖方的目的而言最适于认证。

[0247] 在一种操作模式中，这些不同的权重可以由认证请求者在生成认证请求时定义，并与认证请求一起发送至信任引擎 110。在另一操作模式中，这样的选项也可以在认证请求者的初始注册过程中被设置为优选项并存储在认证引擎中。

[0248] 一旦认证引擎 215 产生所提供的认证数据的认证可信度，该可信度就被用来在步骤 1640 中完成认证请求，并且该信息从认证引擎 215 转发至交易引擎 205，以便包括在至认证请求者的消息中。

[0249] 上述处理仅是示例性的，本领域的技术人员应该理解所述步骤不需要以所示的顺序执行，或者仅希望执行某些步骤，或者可以期望步骤的多种组合。此外，如果环境允许，某些步骤，例如对所提供的每个认证实例的可靠性的评估，可以彼此并行执行。

[0250] 在本发明的另一方面，提供了一种方法以适应当由上述处理产生的认证可信度不能满足卖方或要求认证的其他方所需的信任等级时的情况。在诸如在所提供的可信度和所期望的信任等级之间存在差距的情况下，信任引擎 110 的操作员能够为一方或双方提供用于提供替换数据或要求的机会以弥补该信任差距。该处理在这里将被称作“信任仲裁”。

[0251] 信任仲裁可以在如上面参考图 10 和 11 所述的加密认证的框架中发生。如在此所示，卖方或其他方将请求与特定交易相关联的特定用户的认证。在一种情况下，卖方简单地请求认证，或者肯定或者否定，并且在接收到来自用户的适当数据之后，信任引擎 110 将提供这样的二元认证。在诸如这些的情况下，为了保证肯定认证所需要的可信度是基于信任

引擎 110 中设置的优选项而确定的。

[0252] 然而，卖方可能要求特定的信任等级以完成特定交易也是有可能的。所要求的等级可以包括在认证请求中（例如，认证该用户为 98％的可信度），或可以由信任引擎 110 基于与交易相关联的其他因素确定（即认证该用户为适于该交易）。一个这样的因素可能是交易的经济价值。对于具有较大经济价值的交易，可能需要较高的信任度。类似地，对于具有高风险度的交易，可能需要高信任度。相反，对于或者低风险或者低价值的交易，卖方或其他认证请求者可能要求低的信任等级。

[0253] 信任仲裁的处理发生在图 10 的步骤 1050 中信任引擎 110 接收认证数据的步骤与图 10 的步骤 1055 中将认证结果返回到卖方的步骤之间。在这些步骤之间，如图 17 所示，发生导致信任等级评估和可能的信任仲裁的处理。在执行简单的二元认证的情况下，如图 17 所示的处理减少为使交易引擎 205 直接比较所提供的认证数据和被识别的用户的注册数据，如上参考图 10 所述，将任何不同标记为否定认证。

[0254] 如图 17 所示，在步骤 1050 中接收数据之后的第一步骤是在步骤 1710 中交易引擎 205 确定该特定交易的肯定认证所需的信任等级。该步骤可以由若干不同方法中的一种来执行。所需的信任等级可以由认证请求者在进行认证请求时指定给信任引擎 110。认证请求者还可以事先设置优选项，其被存储在仓库 210 或可由交易引擎 205 访问的其他存储器中。该优选项然后可以在每次由该认证请求者进行认证请求时读取和使用。该优选项还可以与特定用户相关联作为安全性度量，使得总是需要特定的信任等级来认证该用户，用户优选项存储在仓库 210 或其他可由交易引擎 205 访问的存储介质中。所需的等级也可以由交易引擎 205 或认证引擎 215 基于在认证请求中提供的信息而获得，所述信息诸如要认证的交易的价值和风险等级。

[0255] 在一种操作模式中，在生成认证请求时所使用的策略管理模块或其他软件被用来规定认证该交易所需的信任度。这可以被用来提供在基于在策略管理模块中规定的策略而赋予所需信任等级时要遵循的一系列规则。对于这样的模块，一种有利的操作模式是与卖方的 web 服务器合并以适当地确定使用卖方的 web 服务器启动的交易所需的信任等级。以该方式，来自用户的交易请求可以被赋予与卖方的策略一致的所需信任等级，并且这样的信息可以连同认证请求一起被转发至信任引擎 110。

[0256] 所需的信任等级与卖方想要的该认证个体事实上就是他将自己标识为的那个人的确定度有关。例如，如果交易是卖方想要普通确定度的交易——因为货物是易手，则卖方可能要求 85％的信任等级。对于卖方仅是认证用户以允许他观看会员专用内容或在聊天室享有特权，则负面风险很小以至卖方仅要求 60％的信任等级。然而，为了订立价值几万美金的生产合同，卖方可能要求 99％或更高的信任等级。

[0257] 该要求的信任等级代表用户必须认证自己以便完成该交易的衡量标准。如果所要求的信任等级例如是 85％，则用户必须向信任引擎 110 提供足以使信任引擎 110 具有 85％信心认为该用户是他们所声称的那个用户的认证。这是在该要求的信任等级和认证可信度之间的平衡，其或者产生肯定认证（令卖方满意），或者产生信任仲裁的可能。

[0258] 如图 17 所示，在交易引擎 205 接收所要求的信任等级后，在步骤 1720 中将所要求的信任等级与认证引擎 215 为当前认证所计算的认证可信度（如图 16 所讨论的）进行比较。如果在步骤 1730 中认证可信度高于交易所要求的信任等级，则处理进行到步骤 1740，

由交易引擎 205 产生对于该交易的肯定认证。带有这个意思的消息然后被插入认证结果中并通过交易引擎 205 返回至卖方，如步骤 1055 中所示（见图 10）。

[0259] 然而，如果在步骤 1730 中认证可信度不能满足所要求的信任等级，则当前认证存在信任差距，并且在步骤 1750 中进行信任仲裁。下面将参考图 18 更全面地描述信任仲裁。如下所述的该处理在信任引擎 110 的交易引擎 205 中发生。因为执行信任仲裁不需要认证或其他加密操作（除了交易引擎 205 和其他部件之间的 SSL 通信所需的），该处理可以在认证引擎 215 的外部执行。然而，如下所讨论的，认证数据或其他加密或认证事件的任何重新评估都将要求交易引擎 205 重新提交适当数据至认证引擎 215。本领域的技术人员应该意识到，信任仲裁处理可以可替换地被构造成部分或全部在认证引擎 215 本身中发生。

[0260] 如上所述，信任仲裁是信任引擎 110 调解卖方和用户之间的协商以尝试酌情保证肯定认证的处理。如在步骤 1805 中所示，交易引擎 205 首先确定当前状态是否适于信任仲裁。这可以基于认证的环境来确定，例如，如下面将进一步讨论的，基于该认证是否已经经过多次仲裁循环，以及基于卖方或用户的优选项。

[0261] 在仲裁不可能的环境下，处理进行到步骤 1810，在此交易引擎 205 生成否定认证，然后将其插入在步骤 1055（见图 10）中发送至卖方的认证结果中。一个可以被有利地使用以防止认证无限期待定的限制是设置从初始认证请求开始的超时周期。以该方式，任何在该期限内未被肯定认证的交易被拒绝进一步仲裁并被否定认证。本领域的技术人员应该意识到，这样的期限可以根据交易的环境以及用户和卖方的期望而改变。也可以基于在提供成功认证时可进行的尝试次数来设置多种限制。这样的限制可以由如图 5 所示的尝试限制器 535 来处理。

[0262] 如果在步骤 1805 中没有禁止仲裁，则交易引擎 205 将参与与交易一方或双方的协商。如在步骤 1820 中所示的，交易引擎 205 可以发送消息至用户以请求某种形式的附加认证，以便提升所产生的认证可信度。最简单的形式，这可以简单地表明认证不足。也可以发送产生一个或多个另外的认证实例以改进认证的总可信度的请求。

[0263] 如果用户在步骤 1825 提供一些另外的认证实例，则交易引擎 205 将这些认证实例增加到用于交易的认证数据并将其转发至认证引擎 215，如步骤 1015 中所示（见图 10），并基于之前已有的该交易的认证实例和新提供的认证实例重新评估该认证。

[0264] 一种附加的认证类型可以是来自信任引擎 110 的请求在信任引擎 110 操作者（或可信的同事）与用户之间进行某种形式的人与人的联系（例如通过电话）的请求。该电话或其他非计算机认证可以被用来提供与该个人的个人联系以及进行基于认证的某种形式的问卷。这还给出校验发起的电话号码和潜在的在用户打进电话时对其进行语音分析的机会。即使不能提供另外的认证数据，与用户的电话号码相关联的附加情境可以提高认证情境的可靠性。基于该电话的任何修订的数据或环境都被提供至信任引擎 110，供考虑认证请求时使用。

[0265] 此外，在步骤 1820，信任引擎 110 可以为用户提供购买保险的机会，以有效地购买更确信的认证。信任引擎 110 的操作者有时仅希望在认证的可信度高于开始的一定阈值的情况下使这样的选项可用。事实上，这种用户侧保险是信任引擎 110 在认证满足信任引擎 110 对于认证的正常所需信任等级但是不满足卖方对于该交易所要求的信任等级时担保用户的方式。以该方式，用户仍可以成功地认证至卖方所要求的非常高的等级，即使他仅仅具

有产生足以用于信任引擎 110 的可信度的认证实例。

[0266] 信任引擎 110 的该功能允许信任引擎 110 为被认证为满足信任引擎 110 而不满足卖方的人担保。这类似于由公证员执行的功能，即，将其签名添加至文档以向后面读取该文档的人表明其签名出现在文档上的人就是事实上签名的人。公证员的签名证明用户签名的动作。以相同的方式，信任引擎提供交易人就是他们所声称的人的指示。

[0267] 然而，因为信任引擎 110 人为地提升由用户提供的可信度，因此，对于信任引擎 110 操作者来说存在较大风险，这是因为用户实际上没有达到卖方所要求的信任等级。保险的费用被设计为抵消信任引擎 110（其可以有效地公证用户的认证）的错误肯定认证的风险。用户付款给信任引擎 110 操作者以承担认证至高于实际已提供的可信度的风险。

[0268] 因为这样的保险系统允许某个人从信任引擎 110 有效购买较高信任评级，所以卖方和用户可能都希望防止在某些交易中使用用户侧保险。卖方可能希望将肯定认证限制到他们知道实际认证数据支持他们所需的可信度的情况，因而可能指示信任引擎 110 不允许用户侧保险。类似地，为了保护他的在线身份，用户可能希望防止在其帐户上使用用户侧保险，或可能希望对于没有保险时的认证可信度高于一定限度的情况限制该保险的使用。这可以被用作安全措施以防止有人偷听口令或盗取智能卡并使用它们来错误地认证为低的可信度，然后购买保险来产生非常高的（错误的）可信度。在确定是否允许用户侧保险时可以评估这些因素。

[0269] 如果在步骤 1840 中用户购买保险，则在步骤 1845 中，认证可信度基于所购买的保险被调整，以及在步骤 1730 中，认证可信度和所要求的信任等级再次被比较（见图 17）。处理从这里继续，并且可能进行到步骤 1740 中的肯定认证（见图 17）或回到步骤 1750 中的信任仲裁处理，以便进一步仲裁（如果允许）或在进一步仲裁被禁止的情况下进行步骤 1810 中的否定认证。

[0270] 除了在步骤 1820 中发送信息至用户之外，交易引擎 205 还在步骤 1830 中发送消息至卖方，表示待定认证目前低于所要求的信任等级。该消息还向卖方提供关于如何继续进行的各种选项。这些选项之一是简单地通知卖方当前认证可信度是什么以及询问卖方是否希望维持其当前未满足的所要求信任等级。这样是有好处的，因为在一些情况下，卖方可能具有用于认证该交易的独立方式，或者可能已经使用默认的一组要求，其通常导致初始规定的所需等级高于其手边的特定交易实际需要的等级。

[0271] 例如，标准作法可能期望该卖方的所有进入的购买订单交易都满足 98％的信任等级。然而，如果订单是近来通过电话在卖方和长期顾客之间讨论的，并且然后马上认证该交易，但是仅达到 93％可信度，则卖方可能希望简单地降低对于该交易的接受阈值，因为电话有效地向卖方提供了附加认证。在某些情况下，卖方可能愿意降低其所要求的信任等级，但是不是一直降低到当前认证的可信度。例如，上述示例中的卖方可能认为在该订单之前的电话可以值得所需信任度减少 4％，然而，这还是大于由用户产生的 93％的可信度。

[0272] 如果在步骤 1835 中卖方不调节其所要求的信任等级，则通过认证产生的认证可信度和所要求的信任等级在步骤 1730 中被比较（见图 17）。如果可信度现在超过了所要求的信任等级，则在步骤 1740 中在交易引擎 205 中可以生成肯定认证（见图 17）。如果没有超过，则如果允许的话，可以如上所述地尝试进一步仲裁。

[0273] 除了请求调节所要求的信任等级，交易引擎 205 还可以向请求认证的卖方提供卖

方侧保险。该保险起到类似于上述用户侧保险的目的。然而，在这里，当接受认证中的较低信任等级时，保险的费用对应于由卖方承担的风险，而不同于在认证高于所产生的实际认证可信度时，费用对应于由信任引擎 110 所承担的风险的情况。

[0274] 代替仅减低他们实际所要求的信任等级，卖方可以选择购买保险以使其避免与在认证用户时的较低信任等级相关联的其它风险。如上所述，对于卖方来说，仅仅在现有认证已经高于某个阈值的情况下考虑购买这样的保险来弥补信任差距是有利的。

[0275] 提供这样卖方侧保险使得卖方能够选择：直接降低他的信任要求而无需他的附加花费，自己承担错误认证的风险（基于所要求的较低信任等级）；或者为认证可信度和其要求之间的信任差距购买保险，使信任引擎 110 操作者承担已提供的较低可信度的风险。通过购买保险，卖方有效地保持其高的信任等级要求；因为错误认证的风险被转移到信任引擎 110 操作者。

[0276] 如果在步骤 1840 中卖方购买保险，则在步骤 1730 中比较认证可信度和所要求的信任等级（见图 17），并且处理如上所述地继续。

[0277] 注意，用户和卖方也可以都响应来自信任引擎 110 的消息。本领域技术人员应该意识到存在多种能够处理这样的情况的方法。处理多个响应的可能性的一种有利模式是简单地以先到先服务的方式对待响应。例如，如果卖方以降低所要求的信任等级作为响应，并且紧跟其后用户也购买了保险来提高他的认证等级，则认证首先基于来自卖方的降低的信任要求而被重新评估。如果认证现在是肯定的，则用户的保险购买被忽略。在另一种有利的操作模式中，用户可能仅被收取满足新的降低的卖方信任要求所需的保险等级的费用（如果即使使用降低的卖方信任要求，仍然存在信任差距）。

[0278] 在为认证设置的时限之内，如果没有来自任一方的响应在步骤 1850 的信任仲裁处理期间被接收到，则在步骤 1805 中重新评估仲裁。这就有效地再次开始仲裁处理。如果时限结束或其他情况阻止在步骤 1805 中的进一步仲裁，则由交易引擎 205 在步骤 1810 中生成否定认证，并在步骤 1055（见图 10）中返回至卖方。如果不是，则新消息可以发送至用户和卖方，并且处理可以根据需要被重复。

[0279] 注意，对于某些类型的交易，例如，对文档进行数字签名，其不是交易的一部分，可能不一定存在卖方或其他第三方；因此，该交易主要是在用户和信任引擎 110 之间。在这样的情况下，信任引擎 110 将具有其本身要求的信任等级，其必须被满足以生成肯定认证。然而，在这样的情况下，信任引擎 110 通常不希望向用户提供保险以使他增加他自己的签名的可信度。

[0280] 上面所述的以及在图 16-18 中示出的处理可以使用如上参考信任引擎 110 所述的各种通信模式来执行。例如，消息可以是基于网络的，并使用信任引擎 110 与实时下载到在用户或卖方系统上运行的浏览器的小程序之间的 SSL 连接来发送。在替换的操作模式中，用户和卖方可以使用某些专用应用程序，其有助于这样的仲裁和保险交易。在另一替换的操作模式中，可以使用安全电子邮件操作来调解上述仲裁，从而允许延迟的评估和认证批处理。本领域的技术人员应该理解，可以使用适合于环境和卖方的认证要求的不同的通信模式。

[0281] 下面参考图 19 的说明描述结合了上述本发明各个方面的范例交易。该示例示出了由信任引擎 110 调解的在用户和卖方之间的整个处理。尽管上面具体描述的各个步骤和

部件可以被用来执行下面的交易，但是所描述的处理集中在信任引擎 110、用户和卖方之间的相互作用。

[0282] 在步骤 1900，当用户在线观看网页时填写卖方的网站上的订单时，交易开始。用户希望将该使用他的数字签名来签名的订单提交至卖方。为了实现该目的，在步骤 1905，用户将订单和其签名请求一起提交至信任引擎 110。用户还提供将如上所述用于认证其身份的认证数据。

[0283] 在步骤 1910，信任引擎 110 如上所述地将认证数据与注册数据进行比较，如果产生肯定认证，则将用该用户的私钥签名的订单的散列连同订单本身一起转发至卖方。

[0284] 在步骤 1915，卖方接收该签名的订单，然后在步骤 1920，卖方将生成发货单 (invoice) 或其他与将进行的购买有关的合同。在步骤 1925，该合同连同签名请求被发送回用户。在步骤 1930，卖方还向信任引擎 110 发送对该合同交易的认证请求，包括将由双方签名的合同的散列。为了使合同能够被双方数字签名，卖方还包括其本身的认证数据，从而如有必要，卖方在该合同上的签名还可以在以后被校验。

[0285] 如上所讨论的，信任引擎 110 然后校验卖方提供的认证数据以确认卖方的身份，以及如果在步骤 1935 中该数据产生肯定认证，则在从用户接收到数据时继续步骤 1955。如果卖方的认证数据不与卖方的注册数据匹配至期望程度，则返回消息至卖方以请求进一步认证。如上所述的，如有必要，在此可以执行信任仲裁，以便卖方向信任引擎 110 成功认证其本身。

[0286] 在步骤 1940，当用户接收到该合同时，他检查该合同，在步骤 1945 生成认证数据以便在合同可接受的情况下签署该合同，然后在步骤 1950 发送合同的散列和他的认证数据至信任引擎 110。在步骤 1955，信任引擎 110 校验该认证数据，并且如果认证良好，则如下所述地继续处理该合同。如上参考图 17 和 18 所述，信任仲裁可以酌情执行以弥补在认证可信度和交易所要求的认证等级之间的任何信任差距。

[0287] 在步骤 1960，信任引擎 110 使用用户的私钥签署合同的散列，并将该已签名的散列发送至卖方，其中以其自己的名义签署完整消息，即，包括用信任引擎 110 的私钥 510 加密的完整消息（包括用户的签名）的散列。在步骤 1965 中，该消息被卖方接收。该消息代表已签名的合同（用用户的私钥加密的合同的散列）以来自信任引擎 110 的收据（用信任引擎 110 的私钥加密的、包括已签名的合同的消息的散列）。

[0288] 在步骤 1970 中，信任引擎 110 类似地准备具有卖方的私钥的合同的散列，并将由信任引擎 110 签名的该合同的散列转发至用户。这样，在步骤 1975，用户也接收到一份由卖方签名的合同的副本，以及由信任引擎 110 签名的关于该已签名的合同的交付的收据。

[0289] 除了上面所述，本发明的另一方面提供了加密服务提供者模块 (SPM)，其可以用于客户端应用以作为用于访问由上述信任引擎 110 提供的功能的手段。提供这样的服务的一种有利方式是加密 SPM 作为中介实现第三方应用编程接口 (API) 与可通过网络或其他远程连接来访问的信任引擎 110 之间的通信。下面参考图 20 描述范例性的加密 SPM。

[0290] 例如，在典型的系统上，多个 API 对于程序员可用。每个 API 提供一组函数调用，其可以被运行在系统上的应用程序 2000 调用。提供适于加密功能、认证功能和其他安全功能的编程接口的 API 的示例包括由微软提供的使用其 Windows 操作系统的加密 API(CAPI)2010、以及由 IBM、Intel 和 The Open Group 的其他成员提议的通用数据安全结

构 (CDSA)。在下面的讨论中将使用 CAPI 作为示例性安全 API。但是，所述的加密 SPM 可以使用 CDSA 或本领域已知的其他安全 API。

[0291] 在调用加密功能时，该 API 由用户系统 105 或卖方系统 120 使用。包括在这些功能中的可能是与执行各种加密操作相关联的请求，诸如用特定私钥加密文档、签名文档、请求数字证书、校验已签名的文档上的签名、以及如在此所述或本领域技术人员已知的这类其他的加密功能。

[0292] 这样的加密功能通常在 CAPI 2010 所位于的系统处本地地执行。这是因为通常所调用的功能需要使用本地用户系统 105 的资源（例如，指纹读取器）或用在本地机器上执行的库来编程的软件功能。对这些本地资源的访问通常由如上所述的一个或多个服务提供者模块 (SPM)2015、2020 提供，这些模块提供执行加密功能所使用的资源。这样的 SPM 可以包括软件库 2015 以执行加密或解密操作，或包括能够访问专用硬件的驱动器和应用程序 2020，例如生物识别扫描装置。以与 CAPI 2010 提供可由系统 105 的应用程序 2000 使用的功能差不多的方式，SPM 2015、2020 向 CAPI 提供对与系统上可用的服务相关联的较低等级的功能和资源的访问。

[0293] 根据本发明，可以提供一种加密 SPM 2030，其能够访问由信任引擎 110 提供的加密功能，并且能够通过 CAPI 2010 使这些功能对于应用程序 2000 可用。不同于 CAPI 2010 仅能够通过 SPM 2015、2020 访问本地可用的资源的实施例，在此所述的加密 SPM 2030 能够向位于远程的、网络可访问的信任引擎 110 提交加密操作请求以执行期望的操作。

[0294] 例如，如果应用程序 2000 需要加密操作，例如签署文档，则应用程序 2000 对适当的 CAPI 2010 函数进行函数调用。CAPI 2010 随后执行该函数，以使用由 SPM 2015、2020 和加密 SPM 2030 为其提供的资源。在数字签名功能的情况下，加密 SPM 2030 将生成将通过通信链路 125 发送至信任引擎 110 的适当请求。

[0295] 发生在加密 SPM 2030 和信任引擎 110 之间的操作是与任何其他系统和信任引擎 110 之间可能的操作相同的操作。然而，这些功能通过 CAPI 2010 被有效地提供至用户系统 105，从而在用户系统 105 本身看来它们是本地可用的。然而，不同于一般的 SPM 2015、2020，这些功能是在远程信任引擎 110 上执行的，并且结果响应于适当的请求通过通信链路 125 中继到加密 SPM 2030。

[0296] 加密 SPM 2030 使得原本可能不可用于用户系统 105 或卖方系统 12 的大量操作对于用户系统 105 或卖方系统 120 可用。这些功能包括但不限于：加密和解密文档；发布数字证书；文档的数字签名；校验数字签名；以及对于本领域技术人员显而易见的其他操作。

[0297] 在另一实施例中，本发明包括用于在任何数据集上执行本发明的数据安全方法的完整系统。本发明的该计算机系统包括数据拆分模块，其包括图 8 中所示并在此描述的功能。在本发明的一个实施例中，数据拆分模块（有时在此被称为安全数据解析器）包括解析程序或软件套件，其包括数据拆分、加密和解密、重建或重新组装功能。该实施例还可以进一步包括一个数据存储设备或多个数据存储设备。数据拆分模块或安全数据解析器包括跨平台软件模块套件，其集成在电子基础结构中，或作为需要其数据元素极为安全的任何应用程序的插件。该解析处理对任何类型的数据集、以及对任何和所有文件类型进行操作，或在数据库中对该数据库中的任何数据行、列或单元进行操作。

[0298] 在一个实施例中，本发明的解析处理可以以模块化的分层形式来设计，并且任何

加密处理都适于用在本发明的处理中。本发明的解析和拆分处理的模块化层级可以包括但不限于 :1) 加密拆分，分散并安全存储在多个位置 ;2) 加密，加密拆分，分散并安全存储在多个位置 ;3) 加密，加密拆分，加密每份，然后分散并安全存储在多个位置 ;以及 4) 加密，加密拆分，用不同于在第一步骤中使用的加密类型加密每份，然后分散并安全存储在多个位置。

[0299] 在一个实施例中，所述处理包括根据生成的随机数的内容或密钥来拆分数据，以及对在要保护的数据的拆分的加密中使用的密钥执行相同的加密拆分，以形成两个或更多部分或份的解析和拆分数据，并且在一个实施例中优选地形成四个或更多部分的解析和拆分数据，加密所有部分，然后将这些部分分散并存回至数据库，或将它们重新放置到任何指定的装置，这些装置是固定或可移动的，取决于请求者对保密性和安全性的要求。可替换地，在另一实施例中，加密可以在由拆分模块或安全数据解析器拆分数据集之前发生。如在此实施例中描述的那样被处理的原始数据被加密和混乱 (obfuscate) 并且被保护。如果希望，加密元素的分散事实上可以在任何地方，包括但不限于单个服务器或数据存储装置，或在分开的数据存储设备或装置之间。在一个实施例中，加密密钥管理可以被包括在软件套件中，或者在另一实施例中，可以集成在已有基础结构或任何其他期望位置中。

[0300] 加密的拆分 ( 加密拆分，cryptosplit) 将数据划分成 N 份。该划分可以基于任何大小的数据单元，包括单个位、多个位、字节、千字节、兆字节或更大的单元，以及预定或随机生成的数据单元大小的任何模式或组合。基于随机或预定一组值，这些单元也可以具有不同的大小。这意味着数据可以被看做是这些单元的序列。以该方式，数据单元的大小本身可以使得数据更安全，例如，通过使用数据单元大小的一个或多个预定或随机生成的模式、序列或组合。单元然后 ( 随机地或以预定的一组值 ) 被分配成 N 份。该分配也可以包括打乱备份中的单元的顺序。本领域的普通技术人员应该容易理解将数据单元分成多个份可以根据多种可能的选择来执行，包括但不限于固定大小、预定大小、或预定或随机生成的数据单元大小的一种或多种组合、模式或序列。

[0301] 这种加密的拆分处理或加密拆分的一个示例将考虑数据的大小为 23 字节，数据单元大小被选择为 1 字节，以及被选择的份数是 4。每个字节将被分配至这 4 份之一。假设随机分配，密钥可能被得到以创建具有 23 个随机数 (r1，r2，r3 至 r23) 的序列，每个随机数具有对应于 4 份的在 1 和 4 之间的值。每个数据单元 ( 在该示例中有 23 个单独字节的数据 ) 与对应于 4 份之一的 23 个随机数之一相关联。通过将数据的第一字节置于份号 r1、字节 2 置于份 r2、字节 3 置于份 r3、直到数据的第 23 个字节置于份 r23 来将数据的各字节分配成 4 份。本领域的普通技术人员容易理解多种其他可能步骤和步骤的组合和序列，包括数据单元的大小，可以被应用在本发明的加密拆分处理中，并且上述示例是加密拆分数据的一种处理的非限制性描述。为了重新创建原始数据，将执行相反操作。

[0302] 在本发明的加密拆分处理的另一实施例中，加密拆分处理的一个选项是提供份的足够冗余，从而仅需要份的子集来重新组装或恢复数据至其原始或可用形式。作为非限制示例，加密拆分可以作为“4 中 3 个”加密拆分来执行，从而 4 份中仅 3 份是必要的以重新组装或恢复数据至其原始或可用形式。这也被称为“N 中 M 个加密拆分”，其中 N 是份的总数，以及 M 至少比 N 小 1。本领域的技术人员应该理解在本发明的加密拆分处理中存在创建该冗余的多种可能性。

[0303] 在本发明的加密拆分处理的一个实施例中，每个数据单元被存储在两份中，主份和备用份。使用上述的“4 中 3 个”加密拆分处理，任何一份可以缺少，由于仅需要总计 4 份中的 3 份，所以这足以重新组装或恢复没有缺少数据单元的原始数据。如在此所述的，生成对应于多个份之一的随机数。随机数与数据单元相关联，并基于密钥存储在对应的份中。在该实施例中，使用一个密钥来生成主份和备用份随机数。如在此所述的，对于本发明的加密拆分处理，生成等于数据单元的数量的从 0 至 3 的一组随机数（也称作主份号）。然后生成等于数据单元的数量的从 1 至 3 的另一组随机数（也称作备用份号）。每个数据单元然后与一个主份号和一个备用份号相关联。可替换地，可以生成小于数据单元数量的一组随机数，但是这可能降低敏感数据的安全性。主份号被用于确定数据单元被存储在哪个份中。备用份号与主份号结合以创建 0 和 3 之间的第三份号，并且该号被用于确定数据单元被存储在哪个份中。在该示例中，确定第三份号的等式是：

[0304] （主份号 + 备用份号）MOD 4 = 第三份号。

[0305] 在上述实施例中，主份号在 0 和 3 之间以及备用份号在 1 和 3 之间确保第三份号不同于主份号。这就导致数据单元存储在两个不同份中。本领域的技术人员容易理解除了在此公开的实施例，存在多种执行冗余加密拆分和非冗余加密拆分的方法。例如，在每份中的数据单元可以使用不同算法被打乱。这种数据单元打乱例如可以在原始数据被拆分成数据单元时、或在数据单元被置于份中之后、或在份满了之后执行。

[0306] 在此描述的各种加密拆分处理和数据打乱处理，以及本发明的加密拆分和数据打乱方法的所有其他实施例可以在任何大小的数据单元上执行，包括但不限于，与单个位一样小、多位、字节、千字节、兆字节或更大。

[0307] 执行在此所述的加密拆分处理的源代码的一个实施例的一个示例是：

```
[0308] DATA[1:24]- 具有将被拆分的数据的字节数组
[0309] SHARES[0:3 ;1:24]-2 维数组，每行代表份之一
[0310] RANDOM[1:24]- 在 0..3 范围内的数组随机数
[0311] S1 = 1 ;
[0312] S2 = 1 ;
[0313] S3 = 1 ;
[0314] S4 = 1 ;
[0315] For J = 1 to 24 do
[0316]   Begin
[0317]   IF RANDOM[J[ == 0then
[0318]       Begin
[0319]       SHARES[1,S1] = DATA[J] ;
[0320]       S1 = S1+1 ;
[0321]       End
[0322]   ELSE IF RANDOM[J[ == 1then
[0323]       Begin
[0324]       SHARES[2,S2] = DATA[J] ;
[0325]       S2 = S2+1 ;
```

[0326] END

[0327] ELSE IF RANDOM[J[ == 2then

[0328] Begin

[0329] Shares[3,S3] = data[J] ;

[0330] S3 = S3+1 ;

[0331] End

[0332] Else begin

[0333] Shares[4,S4] = data[J] ;

[0334] S4 = S4+1 ;

[0335] End ;

[0336] END ;

[0337] 执行在此描述的加密拆分 RAID 处理的源代码的一个实施例的示例是 :

[0338] 生成两组数,PrimaryShare 是 0 至 3,BackupShare 是 1 至 3。然后使用与上述加密拆分中的处理相同的处理将每个数据单元置于 share[primaryshare[1]] 以及 share[(primaryshare[1]+backupshare[1])]mod 4 中。该方法可以被缩放至任何大小 N,其中仅 N-1 份是恢复数据所必要的。

[0339] 取回、重新结合、重新组装或重建加密数据元素可以使用任何数量的认证技术,包括但不限于 :生物识别,例如指纹识别、面部扫描、手扫描、虹膜扫描、视网膜扫描、耳朵扫描、血管模式识别或 DNA 分析。本发明的数据拆分和 / 或解析模块可以根据需要被集成在多种基本结构产品或应用中。

[0340] 本领域已知的传统加密技术依赖于用于加密数据的一个或多个密钥,使其在没有密钥时不可用。然而,该数据保持完整和完好并易于受攻击。在一个实施例中,本发明的安全数据解析器通过执行加密解析和拆分加密文件成两个或更多个部分或份 ( 在另一个实施例中,优选为四个或更多个份 ),添加另一层加密至每个数据份,然后将各份存储在不同物理和 / 或逻辑位置,来解决这个问题。当通过使用诸如数据存储装置之类的可移除装置或将该份置于另一方的控制下而将一个或多个数据份物理地从系统中去除时,有效地去除了任何泄露被保护数据的可能性。

[0341] 本发明的安全数据解析器的一个实施例的示例和其如何被使用的示例在图 21 中示出并在下面描述。然而,本领域的技术人员应该容易理解本发明的安全数据解析器可以以除下面非限制示例之外的多种方式被应用。作为一个部署选项,以及在一个实施例中,安全数据解析器可以使用会话密钥的外部会话密钥管理或安全内部存储来实现。在实现时,解析器主密钥将被生成,其将被用于保护应用和用于加密目的。还应该注意,在得到的被保护数据中结合解析器主密钥考虑到了由工作组、企业或扩充受众中的个体共享被保护数据的灵活性。

[0342] 如图 21 中所示,本发明的该实施例示出了由安全数据解析器对数据执行的用于与解析数据一起存储会话主密钥的处理的步骤 :

[0343] 1. 生成会话主密钥以及使用 RS1 流密码 (stream cipher) 来加密数据。

[0344] 2. 根据会话主密钥的模式将得到的加密数据分成四个份或部分的解析数据。

[0345] 3. 在该方法的实施例中,会话主密钥将与被保护数据份一起存储在数据仓库中。

根据解析器主密钥的模式将会话主密钥分开并将密钥数据附加至加密的解析数据。

[0346] 4. 得到的四份数据将包括原始数据的加密部分和会话主密钥的各部分。生成用于四份数据中的每份数据的流密钥。

[0347] 5. 加密每份，然后将加密密钥存储在与加密数据部分或份不同的位置：份 1 得到密钥 4，份 2 得到密钥 1，份 3 得到密钥 2，份 4 得到密钥 3。

[0348] 为了恢复原始数据格式，步骤被颠倒。

[0349] 本领域的技术人员应该容易理解在此描述的方法的某些步骤可以根据需要以不同的顺序被执行，或被重复多次。本领域的技术人员还容易理解数据的各部分可以彼此不同地被处理。例如，可以仅对解析数据的一个部分执行多个解析步骤。只要数据可以被重新组装、重建、重组、解密或恢复至其原始或其他可用形式，解析数据的每个部分可以以任何期望方式被唯一地保护。

[0350] 如图 22 所示和在此所述，本发明的另一实施例包括由安全数据解析器对数据执行的用于在一个或多个分开的密钥管理表中存储会话主密钥数据的处理的步骤：

[0351] 1. 生成会话主密钥以及使用 RS1 流密码来加密数据。

[0352] 2. 根据会话主密钥的模式，将得到的加密数据分成 4 个份或部分的解析数据。

[0353] 3. 在本发明的方法的实施例中，会话主密钥将被存储在数据仓库中的单独的密钥管理表中。为该交易生成唯一的交易 ID。将交易 ID 和会话主密钥存储在单独的密钥管理表中根据解析器主密钥的模式分割交易 ID 并将数据附加至加密的被解析或被分割的数据。

[0354] 4. 得到的四份数据将包括加密的原始数据各部分和交易 ID 各部分。

[0355] 5. 为四份数据中的每一份生成流密钥。

[0356] 6. 加密每份，然后将加密密钥存储在与加密数据部分或份不同的位置：份 1 得到密钥 4，份 2 得到密钥 1，份 3 得到密钥 2，份 4 得到密钥 3。

[0357] 为了恢复原始数据格式，步骤被颠倒。

[0358] 本领域的技术人员应该容易理解在此描述的方法的某些步骤可以根据需要以不同的顺序被执行，或被重复多次。本领域的技术人员还容易理解数据的各部分可以彼此不同地被处理。例如，可以仅对解析数据的一个部分执行多个分割或解析步骤。只要数据可以被重新组装、重建、重组、解密或恢复至其原始或其他可用形式，解析数据的每个部分可以以任何期望方式被唯一地保护。

[0359] 如图 23 中所示，本发明的这个实施例示出了由安全数据解析器对数据执行的用于与解析数据一起存储会话主密钥的处理的步骤：

[0360] 1. 访问与认证的用户相关联的解析器主密钥。

[0361] 2. 生成唯一的会话主密钥。

[0362] 3. 从解析器主密钥和会话主密钥的异或函数得到中间密钥。

[0363] 4. 可选地，使用以中间密钥作为密钥的已有或新的加密算法来加密数据。

[0364] 5. 根据中间密钥的模式，将得到的可选加密的数据分成四个份或部分的解析数据。

[0365] 6. 在该方法的实施例中，会话主密钥将与被保护数据份一起存储在数据仓库中。根据解析器主密钥的模式分割会话主密钥并将密钥数据附加至可选加密的解析数据份。

[0366] 7. 得到的多份数据将包括原始数据的各可选加密部分和会话主密钥的各部分。

[0367] 8. 可选地，为四个数据份中的每份生成加密密钥。

[0368] 9. 可选地，使用已有或新的加密算法加密每份，然后将加密密钥存储在不同于加密数据部分或份的位置：例如，份 1 得到密钥 4，份 2 得到密钥 1，份 3 得到密钥 2，份 4 得到密钥 3。

[0369] 为了恢复原始数据格式，步骤被颠倒。

[0370] 本领域的技术人员应该容易理解在此描述的方法的某些步骤可以根据需要以不同的顺序被执行，或被重复多次。本领域的技术人员还容易理解数据的各部分可以彼此不同地被处理。例如，可以仅对解析数据的一个部分执行多个解析步骤。只要数据可以被重新组装、重建、重组、解密或恢复至其原始或其他可用形式，解析数据的每个部分可以以任何期望方式被唯一地保护。

[0371] 如图 24 所示和在此所述，本发明的另一实施例包括由安全数据解析器对数据执行的用于在一个或多个分开的密钥管理表中存储会话主密钥数据的处理的步骤：

[0372] 1. 访问与认证的用户相关联的解析器主密钥。

[0373] 2. 生成唯一的会话主密钥。

[0374] 3. 从解析器主密钥和会话主密钥的异或函数得到中间密钥。

[0375] 4. 可选地，使用以中间密钥作为密钥的已有或新的加密算法来加密数据。

[0376] 5. 根据中间密钥的模式，将得到的可选加密的数据分成四个份或部分的解析数据。

[0377] 6. 在本发明的方法的实施例中，会话主密钥将存储在数据仓库中的单独的密钥管理表中。为该交易生成唯一的交易 ID。将交易 ID 和会话主密钥存储在单独的密钥管理表中或将会话主密钥和交易 ID 传回调用程序以便外部管理。根据解析器主密钥的模式分割交易 ID 并将数据附加至可选加密的被解析或分割的数据。

[0378] 7. 得到的四份数据将包括原始数据的各可选加密部分和交易 ID 的各部分。

[0379] 8. 可选地，为四个数据份中的每份生成加密密钥。

[0380] 9. 可选地，加密每份，然后将加密密钥存储在不同于加密数据部分或份的位置。例如：份 1 得到密钥 4，份 2 得到密钥 1，份 3 得到密钥 2，份 4 得到密钥 3。

[0381] 为了恢复原始数据格式，步骤被颠倒。

[0382] 本领域的技术人员应该容易理解在此描述的方法的某些步骤可以根据需要以不同的顺序被执行，或被重复多次。本领域的技术人员还容易理解数据的各部分可以彼此不同地被处理。例如，可以仅对解析数据的一个部分执行多个分割或解析步骤。只要数据可以被重新组装、重建、重组、解密或恢复至其原始或其他可用形式，解析数据的每个部分可以以任何期望方式被唯一地保护。

[0383] 本领域的技术人员易于理解多种加密方法适于在本发明的方法中使用。一次性密钥 (One Time Pad) 算法通常被称作最安全的加密方法之一，并且适于在本发明的发明中使用。使用一次性密钥算法要求只要要保护数据就产生一个密钥。使用该方法在某些情况下可能是不太期望的，例如因为要保护的数据集的大小而导致要生成和管理非常长的密钥的那些情况。在一次性密钥 (OTP) 算法中，使用简单的异或函数 XOR。对于相同长度的两个二进制流 x 和 y，x XOR y 意味着 x 和 y 的逐位异或。

[0384] 在位级生成：

[0385] 0 XOR 0 = 0

[0386] 0 XOR 1 = 1

[0387] 1 XOR 0 = 1

[0388] 1 XOR 1 = 0

[0389] 在此描述针对要拆分的 n 字节秘密 s（或数据集）的该处理的示例。该处理将生成 n 字节随机值 a，然后设置：

[0390] b = a XOR s。

[0391] 注意可以通过以下等式得到“s”：

[0392] s = a XOR b。

[0393] 值 a 和 b 被称作份或部分并被放置在分开的仓库中。一旦秘密 s 被拆分成两个或更多份，就以安全方式将其丢弃。

[0394] 本发明的安全数据解析器可以使用该功能，执行结合多个不同秘密密钥值 K1，K2，K3，Kn，K5 的多个 XOR 功能。在操作的开始，要保护的数据通过第一加密操作，安全数据 = 数据 XOR 秘密密钥 5：

[0395] S = D XOR K5

[0396] 为了安全地将得到的加密数据存储在例如四个份 S1，S2，S3，Sn 中，根据 K5 的值，数据被解析并被拆分成“n”个段或份。该操作产生原始加密数据的“n”个伪随机份。然后可以对具有剩余秘密密钥值的每个份执行随后的 XOR 函数，例如：安全数据段 1 = 加密数据份 1XOR 秘密密钥 1：

[0397] SD1 = S1 XOR K1

[0398] SD2 = S2 XOR K2

[0399] SD3 = S3 XOR K3

[0400] SDn = Sn XOR Kn.

[0401] 在一个实施例中，可能不期望使任何一个仓库包括足够信息来解密保存在那里的信息，因此解密该份所需的密钥被存储在不同的数据仓库中：

[0402] 仓库 1：SD1，Kn

[0403] 仓库 2：SD2，K1

[0404] 仓库 3：SD3，K2

[0405] 仓库 n：SDn，K3

[0406] 此外，附加至每份的可以是找回原始会话加密密钥 K5 所需的信息。因此，在这里所述的密钥管理示例中，被拆分成“n”份的交易 ID 根据依赖于安装的解析器主密钥 (TID1，TID2，TID3，TIDn) 的内容参考原始会话主密钥：

[0407] 仓库 1：SD1，Kn，TID1

[0408] 仓库 2：SD2，K1，TID2

[0409] 仓库 3：SD3，K2，TID3

[0410] 仓库 n：SDn，K3，TIDn

[0411] 在这里描述的结合会话密钥示例中，会话主密钥根据依赖于安装的解析器主密钥 (SK1，SK2，SK3，SKn) 被拆分成“n”份：

[0412] 仓库 1：SD1，Kn，SK1

[0413] 仓库 2 :SD2，K1，SK2

[0414] 仓库 3 :SD3，K2，SK3

[0415] 仓库 n :SDn，K3，SKn

[0416] 根据该示例，除非所有四份都被找回，否则数据不能被重新组装。即使所有四份都被获取，在无法访问会话主密钥和解析器主密钥的情况下，也不存在重新组装或恢复原始信息的可能性。

[0417] 该示例已经描述了本发明的方法的实施例，在另一实施例中还描述了将份置于仓库中所用的算法，从而来自所有仓库的份能够被结合以形成秘密认证材料。所需的计算非常简单和迅速。然而，对于一次性密钥 (OTP) 算法，可能存在导致其不太被希望使用的情况，例如，因为密钥大小与将被存储的数据大小相同而导致将被保护的数据集很大。因此，需要存储和传输大约两倍于原始数据的量，这在某些情况下是不太期望的。

[0418] 流密码 RS1

[0419] 流密码 RS1 拆分技术非常类似于在此描述的 OTP 拆分技术。代替 n 字节随机值，生成 n’＝ min(n,16) 字节随机值并用作 RS1 流密码算法的密钥。RS1 流密码算法的优点是从小得多的种子数来生成伪随机密钥。RS1 流密码加密执行的速度也大约是本领域已知的三重 DES 加密的速度的 10 倍而不损害安全性。RS1 流密码算法在本领域是公知的，并且可以用于生成在 XOR 函数中使用的密钥。RS1 流密码算法与其他商用的流密码算法（例如，RSA Security，Inc 的 RC4™ 流密码算法）共同使用，并且适于在本发明的方法中使用。

[0420] 使用上述密钥符号，K1 至 K5 现在是 n’字节随机值，并且我们设置 ：

[0421] SD1 ＝ S1 XOR E(K1)

[0422] SD2 ＝ S2 XOR E(K2)

[0423] SD3 ＝ S3 XOR E(K3)

[0424] SDn ＝ Sn XOR E(Kn)

[0425] 其中 E(K1) 至 E(Kn) 是从以 K1 至 Kn 作为密钥的 RS1 流密码算法的输出的前 n’字节。如在此所述，份现在被放置到数据仓库中。

[0426] 在该流密码 RS1 算法中，所需的要求的计算几乎与 OTP 算法一样简单和迅速。该示例中使用 RS1 流密码的好处在于系统平均需要存储和发送仅仅比每份要保护的原始数据的大小多大约 16 字节。当原始数据的大小大于 16 字节时，该 RS1 算法比 OTP 算法更有效，因为其较短。本领域的技术人员容易理解多种加密方法或算法适合在本发明中使用，包括但不限于 RS1、OTP、RC4™、三重 DES 和 AES。

[0427] 与传统加密方法相比，本发明的数据安全方法和计算机系统提供重要优点。一个优点是从将数据的份移动至在一个或多个数据仓库或存储装置上的不同位置（可能在不同逻辑、物理或地理位置）所得到的安全性。例如，当数据的份被物理拆分并在不同人员的控制下时，泄露数据的可能性被大大降低。

[0428] 本发明的方法和系统的另一优点是用于保护数据的本发明的方法的步骤组合提供了一种维护敏感数据的安全性的综合处理。数据被用安全密钥加密并根据该安全密钥被拆分成一个或多个份（在一个实施例中是四份）。安全密钥被安全地存储，带有根据安全密钥被保护成四个份的引用指针。各数据份然后被分别加密，并且各密钥随着不同加密份被安全存储。当组合时，根据在此公开的方法保护数据的整个处理成为数据安全的综合包。

[0429] 根据本发明的方法保护的数据容易被取回以及被恢复、重建、重新组装、解密或返回成其原始的或其他适于使用的形式。为恢复原始数据，可以使用下面的项目：

[0430] 1. 数据集的所有份或部分。

[0431] 2. 再现用于保护数据的方法的处理流程的知识和能力。

[0432] 3. 对会话主密钥的访问。

[0433] 4. 对解析器主密钥的访问。

[0434] 因此，可以期望计划一种安全的安装，其中上述元素中的至少一个可能与系统的剩余部分物理分开（例如处于不同系统管理员的控制之下）。

[0435] 通过使用解析器主密钥可以加强对调用数据保护方法应用的不良应用程序的防范。在采取任何动作之前，在本发明的该实施例中可以要求安全数据解析器和应用程序之间的相互认证握手。

[0436] 系统的安全性表示不存在重新创建原始数据的“后门”方法。对于可能发生数据恢复问题的安装，安全数据解析器可以被加强以提供四个份和会话主密钥仓库的镜像。诸如 RAID（廉价磁盘冗余阵列，用于在多个磁盘上分布信息）之类的硬件选择和诸如复制之类的软件选择也可以有助于数据恢复计划。

[0437] 密钥管理

[0438] 在本发明的一个实施例中，数据保护方法使用三组密钥用于加密操作。基于安装，每组密钥可以具有单独的密钥存储、取回、安全和恢复选项。可以被使用的密钥包括但不限于：

[0439] 解析器主密钥

[0440] 该密钥是与安全数据解析器的安装相关联的单独密钥。其被安装在已经被配置了安全数据解析器的服务器上。存在多种适于保护该密钥的选项，包括但不限于，例如，智能卡、分开的硬件密钥存储器、标准密钥存储器、定制密钥存储器、或在安全的数据库表中。

[0441] 会话主密钥

[0442] 会话主密钥可以在每次保护数据时生成。会话主密钥被用于在解析和拆分操作之前加密该数据。其也可以被结合为解析被加密的数据的方式（如果会话主密钥没有集成在解析数据中）。会话主密钥可以以多种方式被保护，包括但不限于，例如，标准密钥存储器、定制密钥存储器、分开的数据库表、或在加密份中被保护。

[0443] 份加密密钥

[0444] 对于创建的数据集的每个份或部分，可以生成单独的份加密密钥以进一步加密这些份。份加密密钥可以被存储在与被加密的份不同的份中。

[0445] 本领域的技术人员容易理解，本发明的数据保护方法和计算机系统被广泛应用于任何设置或环境中的任何类型的数据。除了通过因特网或在顾客与卖方之间执行的商业应用，本发明的数据保护方法和计算机系统非常适合应用于非商业或私有设置或环境中。可以使用在此描述的方法和系统来保护期望对任何未授权用户保密的任何数据集。例如，通过采用本发明的用于保护数据的方法和系统，对公司或组织内部的特定数据库的访问可以有利地被限制到仅仅所选的用户。另一示例是生成、修改或访问文档，其中期望限制访问或者防止在所选择的个人、计算机或工作站群组之外的未授权或意外访问或公开。本发明的数据保护的方法和系统的这些和其他示例方式能够应用于需要任何设置的任何非商业或

商业环境或设置，包括但不限于任何组织、政府机构或公司。

[0446] 在本发明的另一实施例中，数据保护方法使用三组密钥用于加密操作。基于安装，每组密钥可以具有单独的密钥存储、取回、安全和恢复选项。可以被使用的密钥包括但不限于：

[0447] 1. 解析器主密钥

[0448] 该密钥是与安全数据解析器的安装相关联的单独的密钥。其安装在已配置了安全数据解析器的服务器上。有多种适合于保护该密钥的选项，包括但不限于，例如，智能卡、分开的硬件密钥存储器、标准密钥存储器、定制密钥存储器、或在安全的数据库表中。

[0449] 2. 会话主密钥

[0450] 会话主密钥可以在每次数据被保护时生成。会话主密钥用于与解析器主密钥联合以得到中间密钥。会话主密钥可以以多种方式被保护，包括但不限于，例如，标准密钥存储器、定制密钥存储器、分开的数据库表、或在加密的份中被保护。

[0451] 3. 中间密钥

[0452] 中间密钥可以在每次数据被保护时生成。中间密钥用于在解析和拆分操作之前加密数据。其也可以被结合为解析加密数据的一种方式。

[0453] 4. 份加密密钥

[0454] 对于创建的数据集的每个份或部分，可以生成单独的份加密密钥以进一步加密这些份。份加密密钥可以存储在与被加密的份不同的份中。

[0455] 本领域的普通技术人员容易理解，本发明的数据保护方法和计算机系统可以广泛用于任何设置或环境中的任何类型的数据。除了通过因特网或在顾客和卖方之间进行的商业应用，本发明的数据保护方法和计算机系统还非常适合于非商业或私有设置或环境。可以使用此处描述的方法和系统来保护期望对任何未授权用户保密的任何数据集。例如，通过使用本发明的用于保护数据的方法和系统，对公司或组织内的特定数据库的访问可以有利地被限制到为仅仅所选择的用户。另一个示例是生成、修改或访问文档，其中，期望限制访问或者防止在所选择的个人、计算机或工作站群组之外的未授权或意外访问或公开。本发明的数据保护的方法和系统的这些和其他示例方式适合于需要任何设置的任何非商业或商业环境或设置，包括但不限于，任何组织、政府机构或公司。

[0456] 工作组、项目、个人 PC/ 膝上型电脑、或跨平台数据安全

[0457] 本发明的数据保护方法或计算机系统还可用于保护用在例如创建、处理或存储敏感数据的商业、办公室、政府机构或任何设置中的工作组、项目、个人 PC/ 膝上型电脑和任何其他平台的数据。本发明提供保护数据的方法和计算机系统，已知它们是诸如美国政府之类的组织所寻求的，以便跨整个政府组织或在州一级或联邦一级的政府之间实施。

[0458] 本发明的数据保护方法和计算机系统不仅提供解析和拆分平面文件 (flat file) 的能力，还提供解析和拆分任何类型的数据字段、集和或表的能力。此外，所有形式的数据都能够在该处理下被保护，包括但不限于，文本、视频、图像、生物识别、或语音数据。本发明的保护数据的方法的可伸缩性、速度和数据处理量仅受用户具有的可自由支配的硬件限制。

[0459] 在本发明的一个实施例中，数据保护方法如下所述在工作组环境中使用。在一个实施例中，如图 23 所示和以下描述的，本发明的工作组规模数据保护方法使用信任引擎的

私钥管理功能来存储用户/组关系和用户组共享安全数据所必需的相关私钥(解析器组主密钥)。本发明的方法具有根据解析器主密钥是如何配置的来保护用于企业、工作组、或个人用户的数据的能力。

[0460] 在一个实施例中,可以提供附加的密钥管理和用户/组管理程序,实现具有单点管理和密钥管理的大规模工作组。密钥生成、管理和废除由单个维护程序来处理,其随着用户数量的增加都变得特别重要。在另一个实施例中,密钥管理还可以跨一个或多个不同系统管理者来建立,其可以根据需要不允许任何一个人或组控制数据。这允许通过如由组织定义的角色、责任、从属关系、权限等来获得对被保护数据的管理,并且对于那些被准许或要求仅仅能够访问他们工作的部分的人,可以限制他们对被保护数据的访问,而其他人,例如经理或主管,可以访问全部的被保护数据。该实施例允许公司或组织内的不同组之间共享被保护的数据,而同时仅允许某些被选择的个人,例如具有被授权和预定的角色和责任的人,来观察作为整体的数据。此外,本发明的方法和系统的该实施例还允许在例如分开的公司之间、或公司的分开的部或部门之间、或任何政府或组织或任何类型的任何分开的组织部、组、机构、或办公室等等之间共享数据,其中需要某种共享,但是没有任何一方可以被允许有权访问所有数据。需要和使用本发明的此类方法和系统的特别明显的示例允许共享,但在例如政府区域、机构和办公室之间和大公司或任何其他组织的不同部门、部或办公室之间保持安全性。

[0461] 本发明方法的较小规模应用的示例如下。解析器主密钥被用作对组织的安全数据解析器的编序或标记。因为解析器主密钥的使用的规模从整个企业减小到较小的工作组,所以此处描述的数据保护方法用于在用户组内共享文件。

[0462] 在图 25 中示出并在以下描述的示例中,定义了六个用户以及他们在组织内的职务或角色。侧栏代表五个可能的组,用户根据他们的角色可以属于其中。箭头代表一个或多个组中用户的从属关系。

[0463] 当配置用于在该示例中使用的安全数据解析器时,系统管理者通过维护程序来从操作系统访问用户和组信息。该维护程序基于组中的从属关系生成解析器组主密钥并将其赋予用户。

[0464] 在该示例中,高级员工组中有三个成员。对于该组,动作为:

[0465] 1. 访问用于高级员工组的解析器组主密钥(如果不可用则生成一个密钥);

[0466] 2. 生成关联 CEO 和高级员工组的数字证书;

[0467] 3. 生成关联 CFO 和高级员工组的数字证书;

[0468] 4. 生成关联市场副总裁和高级员工组的数字证书。

[0469] 对每个组以及每个组内的每个成员,都会执行相同的动作集。当维护程序完成时,解析器组主密钥成为组中每个成员的共享凭证。在用户从组中被移除时,所赋予的数字证书的废除可以通过维护程序自动完成,而不影响组中的剩余成员。

[0470] 一旦已经定义了共享的凭证,解析和拆分处理保持相同。当文件、文档或数据元素要被保护时,提示用户在保护数据时要使用的目标组。得到的被保护数据仅对该目标组的其他成员是可访问的。本发明的方法和系统的该功能可以与任何其他的计算机系统或软件平台一起使用,并且可以例如集成在已有的应用程序中或单独用于文件安全。

[0471] 本领域的普通技术人员容易理解,加密算法的任何一种或组合都适于在本发明的

方法和系统中使用。例如，在一个实施例中，可以重复加密步骤以产生多层加密方案。此外，不同的加密算法或加密算法的组合可以在重复加密步骤中使用，从而不同的加密算法可以应用于多层加密方案的不同层。同样地，加密方案本身可以成为用于保护敏感数据免于未授权使用或访问的本发明的方法的组成部分。

[0472] 安全数据解析器可以包括作为内部部件、作为外部部件、或作为两者的误差检查部件。例如，在一种适当的方法中，因为使用根据本发明的安全数据解析器来创建数据的各部分，为了保证部分内数据的完整性，在该部分内以预设间隔采用散列值并将其附加在间隔的末端。散列值是数据的可预测并可再生的数字表示。如果数据内的任何位发生改变，散列值都会不同。然后扫描模块（作为安全数据解析器外部的独立部件或作为内部部件）可以扫描由安全数据解析器生成的数据的各部分。将数据的每个部分（或可替换地，根据某些间隔或通过随机或伪随机采样，少于数据的所有部分）与附加的一个或多个散列值进行比较并且可以采取动作。该动作可以包括匹配或不匹配的值的报告，不匹配的值的警告，或引发数据恢复的一些外部或内部程序的调用。例如，根据本发明，基于可以需要少于所有的部分来生成原始数据这一概念，数据的恢复可以通过调用恢复模块来执行。

[0473] 任何其他适当的完整性检查可以使用附加在所有数据部分或其子集的任何位置的任何适当的完整性信息来实现。完整性信息可以包括可用于确定数据部分的完整性的任何适当的信息。完整性信息的示例可以包括基于任何适当参数计算的散列值（例如，基于各自的数据部分）、数字签名信息、消息认证码 (MAC) 信息、任何其他适当的信息、或其任意组合。

[0474] 本发明的安全数据解析器可以用于任何适当的应用。即，此处描述的安全数据解析器在计算和技术的不同领域内具有各种应用。几个此类领域将在以下讨论。应当理解，这些实际上仅仅是示例性的，并且任何其他适当的应用都可以使用该安全数据解析器。应当进一步理解，所描述的示例仅仅是示例性实施例，其可以以各种适当的方式被修改，以满足任何适当的愿望。例如，解析和拆分可以基于任何适当的单位，例如以位、以字节、以千字节、以兆字节、以其任意组合，或以任何其他适当的单位。

[0475] 本发明的安全数据解析器可以用于实现安全的物理令牌，存储在物理令牌中的数据可以被要求以访问存储在另一个存储区域中的其他数据。根据本发明，在一适当方法中，物理令牌，诸如紧凑型 USB 闪存、软盘、光盘、智能卡、或任何其他适当的物理令牌，都可以用于存储解析数据的至少两个部分之一。为了访问原始数据，USB 闪存需被访问。因此，持有解析数据的一个部分的个人计算机在能够访问原始数据之前需要将具有解析数据的另一部分的 USB 闪存连接上。图 26 示出了该应用。存储区域 2500 包括解析数据的一个部分 2502。具有解析数据的一个部分 2506 的物理令牌 2504 需要使用任何适当的通信接口 2508（例如，USB、串行、并行、蓝牙、IR、IEEE 1394、以太网、或任何其他适当的通信接口）连接至存储区域 2500 以访问原始数据。在例如计算机上的敏感数据不被管理并遭受未授权访问尝试的情况下，这是有用的。通过移除物理令牌（例如，USB 闪存），敏感数据就不可被访问。应当理解，使用物理令牌的任何其他适当方法都可以使用。

[0476] 本发明的安全数据解析器可以用于实现安全认证系统，通过该系统，使用安全数据解析器来解析和拆分用户注册数据（例如，口令、私有加密密钥、指纹模板、生物识别数据、或任何其他适当的用户注册数据）。用户注册数据可以被解析和拆分，藉此一个或多个

部分被存储在智能卡、政府公共访问卡、任何适当的物理存储装置（例如，磁盘或光盘、USB密钥驱动器等）、或任何其他适当的装置上。解析的用户注册数据的一个或多个其他部分可以被存储在执行认证的系统中。这为认证处理提供了增加的安全等级（例如，除了从生物识别源获取的生物识别认证信息，还必须通过适当的解析和拆分数据部分获取用户注册数据）。

[0477] 本发明的安全数据解析器可以集成到任何适当的已有系统中，以在每个系统分别的环境中提供其功能的使用。图 27 示出了示例性系统 2600 的框图，其可以包括软件、硬件、或两者以实现任何适当的应用。系统 2600 可以是现有的系统，在其中安全数据解析器 2602 可以作为集成的部件被装备。可替换地，安全数据解析器 2602 可以例如从其最早设计阶段就集成到任何适当系统 2600 在。安全数据解析器 2600 可以被集成在系统 2600 的任何适当层。例如，安全数据解析器 2602 可以集成到系统 2600 中足够后端 (back-end) 的层，从而安全数据解析器 2602 的存在可以对系统 2600 的终端用户基本上是透明的。根据本发明，安全数据解析器 2602 可以用于在一个或多个存储装置 2604 之中解析和拆分数据。具有集成在其中的安全数据解析器的系统的一些示例性示例在以下讨论。

[0478] 本发明的安全数据解析器可以集成到操作系统内核（例如，Linux，Unix，或其他任何适当的商用或私有操作系统）。该集成可以用于在装置级别上保护数据，藉此，例如，通常存储在一个或多个装置中的数据由集成到操作系统中的安全数据解析器分成为一定数量的部分并存储在一个或多个装置中。当试图访问原始数据时，也集成到操作系统中的适当的软件可以以对终端用户透明的方式将解析的数据部分重组为原始数据。

[0479] 本发明的安全数据解析器可以被集成到存储器系统的卷管理器或任何其他适当的部件中，以便跨任何或所有支持的平台来保护本地和网络数据存储器。例如，使用集成的安全数据解析器，存储器系统可以利用由安全数据解析器提供的冗余（即，用于实现需要少于所有分开的数据部分来重建原始数据这一特征的冗余）以保护数据免于丢失。安全数据解析器还允许所有数据以根据本发明的解析所生成的多个部分的形式写入存储装置，不论是否使用冗余。当试图访问原始数据时，也集成到存储系统的卷管理器或其他适当部件中的适当软件可以以对终端用户透明的方法将解析数据部分重组为原始数据。

[0480] 以一种适当的方法，本发明的安全数据解析器可以被集成到 RAID 控制器中（作为硬件或软件）。这允许安全存储数据至多个驱动器，而在驱动器故障的情况下保持容错。

[0481] 本发明的安全数据解析器可以集成到数据库中以便例如保护敏感的表信息。例如，在一种适当的方法中，与数据库表的特定单元（例如，单独的单元、一个或多个特定列、一个或多个特定行、或其任意组合、或整个数据库表）相关联的数据可以根据本发明被解析或分离（例如，不同部分存储在一个或多个位置处的一个或多个存储装置上或存储在单个存储装置上）。可以通过传统认证方法（例如，用户名和密码询问）准许为了查看原始数据而访问以重组多个部分。

[0482] 本发明的安全解析器可以集成到涉及移动数据（即，数据从一个位置转移到另一个位置）的任何适当的系统。这样的系统包括例如电子邮件、流数据广播、和无线（例如，WiFi）通信。关于电子邮件，在一种适当的方法中，安全解析器可以用于解析发出的消息（即，包含文本、二进制数据、或两者（例如，附加到电子邮件消息的文件））以及沿不同通道发送解析数据的不同部分，从而创建多个数据流。如果这些数据流中的任何一个被泄露，原

始信息保持安全，因为根据本发明，系统可能需要一个以上的部分以生成原始数据。在另一适当的方法中，数据的不同部分可以沿一个路径顺序发送，从而如果一个部分被获取，可能不足以生成原始数据。根据本发明，这些不同部分到达预定接收者的位置，且可以被组合以生成原始数据。

[0483] 图 28 和 29 是此类电子邮件系统的示例性框图。图 28 示出了发送者系统 2700，其可以包括任何适当的硬件，诸如计算机终端、个人计算机、手持装置（例如，PDA、Blackberry）、蜂窝电话、计算机网络、任何其他适当的硬件、或其任意组合。发送者系统 2700 用于生成和 / 或存储消息 2704，其可以是例如电子邮件消息、二进制数据文件（例如，图形、声音、视频等）、或两者。消息 2704 由根据本发明的安全数据解析器 2702 被解析和拆分。得到的各数据部分可以经过网络 2708（例如，因特网、内部网、LAN、WiFi、蓝牙、任何其他适当的有线或无线通信方式、或其任意组合）上的一个或多个分开的通信通道 2706 传送到接收者系统 2710。这些数据部分可以时间上并行地或可替换地根据不同数据部分的通信之间的任何适当时间延迟而被传送。接收者系统 2710 可以是上面参照发送者系统 2700 描述的任何适当的硬件。根据本发明，沿通信通道 2706 运送的分开的数据部分在接收者系统 2710 处被重组以生成原始消息或数据。

[0484] 图 29 示出了发送者系统 2800，其可以包括任何适当的硬件，例如计算机终端、个人计算机、手持装置（例如，PDA）、蜂窝电话、计算机网络、任何其他适当的硬件、或其任意组合。发送者系统 2800 用于生成和 / 或存储消息 2804，其可以是例如，电子邮件消息、二进制数据文件（例如，图形，声音，视频等）、或两者。消息 2804 由根据本发明的安全数据解析器 2802 被解析和拆分。得到的各数据部分可以通过网络 2808（例如，因特网、内部网、LAN、WiFi、蓝牙、任何其他适当的通信方式、或其任意组合）上的单个通信通道 2806 传送到接收者系统 2810。这些数据部分可以通过通信通道 2806 相对于彼此串行地通信。接收者系统 2810 可以是上面参照发送者系统 2800 描述的任何适当的硬件。根据本发明，沿通信通道 2806 运送的分开的数据部分在接收者系统 2810 处重组，以生成原始消息或数据。

[0485] 应当理解，图 28 和 29 的布置仅是示例性的。可以使用任何其他适当的布置。例如，在另一种适当的方法中，图 28 和 29 中的系统的特征可以组合，从而使用图 28 中的多通道方法并且一个或多个通信通道 2706 被用于运送一个以上数据部分，如图 29 的上下文中通信通道 2806 所做的那样。

[0486] 安全数据解析器可以集成到数据移动系统的任何适当层级。例如，在电子邮件系统的情形中，安全解析器可以集成到用户界面层级中（例如，集成到**Microsoft®** Outlook），在该情况下，在使用电子邮件时，用户可以控制安全数据解析器特征的使用。可替换地，安全解析器可以在例如交换服务器的后端部件中实现，在该种情况下消息可以根据本发明被自动地解析、拆分，并沿不同通道传送而无需任何用户干涉。

[0487] 类似地，在流广播数据（例如，音频、视频）的情况下，发出的数据可以被解析并分成多个流，每个流包括解析数据的一部分。根据本发明，多个流可以沿一个或多个通道被传输并在接收者位置处被重组。该方法的益处之一为其避免了与传统数据加密然后在单个通信信道上传输加密数据相关联的相对大的开销。本发明的安全解析器允许移动的数据以多个并行流发送，增加了速度和效率。

[0488] 应当理解，安全数据解析器可以被集成以保护和容错通过任何传输介质（包括

例如有线、无线、或物理的）的任何类型的移动数据。例如，因特网协议语音（VoIP）应用可以使用本发明的安全数据解析器。往返于任何适当的个人数字助手（PDA）装置（例如Blackberry 和 SmartPhone）的无线或有线数据传输可以使用本发明的安全数据解析器来保护。使用用于对等和基于集线器的无线网络的无线 802.11 协议的通信、卫星通信、点对点无线通信、因特网客户 / 服务器通信、或任何其他适当的通信，可以包括根据本发明的安全数据解析器的移动数据（data in motion）的能力。计算机外围装置（例如，打印机，扫描器，监视器，键盘，网络路由器，生物识别认证装置（例如，指纹扫描器），或任何其他适当的外围装置）之间、计算机和计算机外围装置之间、计算机外围装置和任何其他适当的装置之间、或其任意组合之间的数据通信，可以使用本发明的移动数据特征。

[0489] 本发明的移动数据特征还可以应用于使用例如分开的路径、传输媒介、方法、任何其他适当的物理传输、或其任意组合的安全份的物理传输。例如，数据的物理传输可以发生在数字 / 磁带、软盘、光盘、物理令牌、USB 驱动器、可移除硬盘、具有闪存的消费电子装置（例如，苹果 IPOD 或其他 MP3 播放器）、闪存、用于传输数据的任何其他适当的介质，或其任意组合上。

[0490] 根据本发明的安全数据解析器可以提供具有灾难恢复能力的安全性。根据本发明，为了恢复原始数据，可能需要少于安全数据解析器生成的分离的数据的所有部分。就是说，在存储的 m 个部分中，n 个可以是该 m 个部分中恢复原始数据所需的最小数目，其中 n ＜＝ m。例如，如果四个部分的每一个都存储在相对于其他三个部分不同的物理位置，则如果在该示例中 n ＝ 2，则两个位置可以被损坏从而数据被毁坏或不可访问，但原始数据还可以从其他两个位置中的部分来恢复。任何适当的 n 或 m 的值都可以使用。

[0491] 此外，本发明的 m 中 n 个这一特征可以用于创建“二人法则”，藉此避免了委托单个人或任何其他实体具有对可能是敏感数据的全访问权限，为了恢复原始数据，两个或更多个不同的实体（其中每一个具有通过本发明的安全解析器解析的分开的数据的一部分）可能需要同意将他们的部分放到一起。

[0492] 本发明的安全数据解析器可以用于向一组实体提供组范围密钥（group-wide key），该组范围密钥允许组成员访问被授权由该特定组访问的特定信息。组密钥可以是由根据本发明的安全解析器生成的数据部分之一，其需要与中心存储的另一个部分组合，例如以恢复寻求的信息。该特征允许例如在组当中的安全合作。其可以应用于例如专用网络、虚拟专用网络、内部网、或任何其他适当的网络。

[0493] 这样使用安全解析器的具体应用包括，例如，联合信息共享，其中，例如，多国友好政府部队被给予通过单个网络或双网络（dualnetwork）（即，与现今使用的包括相对基本人工的处理的很多网络相比）在向每个各自国家授权的安全等级上传送军事行动数据或其他敏感数据的能力。该能力还可应用于公司或其他组织，其中需要被一个或多个特定个人（组织内或外）知道的信息可以通过单个网络传送，而不需要担心未授权的个人查看该信息。

[0494] 另一个具体应用包括用于政府系统的多等级安全体系。就是说，本发明的安全解析器可以提供使用单个网络以不同保密信息等级（非保密、保密、机密、绝对机密）操作政府系统的能力。如果期望，更多网络可以被使用（例如，用于绝对机密的单独的网络），但本发明允许大大少于当前每个保密等级使用一个单独的网络的布置。

[0495] 应当理解，本发明的安全解析器的上述应用的任何组合都可以被使用。例如，组密钥应用可以与移动数据安全应用一起使用（即，根据本发明，在网络上通信的数据仅能被各自组的成员访问，并且当数据在移动时，数据被拆分到多个通道之中（或以顺序的部分被发送））。

[0496] 本发明的安全数据解析器可以集成到任何中间件应用程序以使得应用程序能够安全地将数据存储到不同的数据库产品中或存储到不同的装置中，而无需修改应用程序或数据库。中间件是允许两个分开的并且已经存在的程序进行通信的产品的通称。例如，在一种适当的方法中，集成有安全数据解析器的中间件可以用于允许为特定数据库所写的程序与其他数据库通信，而不用自定义编码。

[0497] 本发明的安全数据解析器可以被实现为具有任何适当能力（例如此处所讨论的那些能力）的任意组合。在本发明的一些实施例中，例如，安全数据解析器可以被实现为仅具有某些能力，而其他能力可以通过使用直接或间接地与安全数据解析器接口的外部软件、硬件、或两者来获得。

[0498] 例如，图 30 示出了安全数据解析器 3000 作为安全数据解析器的示例性实施方式。安全数据解析器 3000 可以被实现为具有非常少的内置能力。如图所示，根据本发明，安全数据解析器 3000 可以包括使用模块 3002 将数据解析和拆分为多个部分（此处也称作份）的内置能力。安全数据解析器 3000 还可以包括使用模块 3004 来执行冗余以便能够实现例如上述 n 中 m 个这一特征（即，使用少于所有份的被解析和拆分的数据来重建原始数据）的内置能力。根据本发明，安全数据解析器 3000 还可以包括份分配能力，其使用模块 3006 来将数据份放置到缓冲器中，数据从缓冲器被发送以便通信至远程位置、以便存储等。应当理解，任何其他适当的能力都可以内置到安全数据解析器 3000 中。

[0499] 组装数据缓冲器 3008 可以是用于存储原始数据（但不必以其原始形式）的任何适当的存储器，该原始数据将被安全数据解析器 3000 解析和拆分。在拆分操作中，组装数据缓冲器 3008 提供到安全数据解析器 3008 的输入。在恢复操作中，组装数据缓冲器 3008 可以用于存储安全数据解析器 3000 的输出。

[0500] 拆分份缓冲器 3010 可以是一个或多个存储器模块，其可以用于存储从解析和拆分原始数据得到的多个数据份。在拆分操作中，拆分份缓冲器 3010 持有安全数据解析器的输出。在恢复操作中，拆分份缓冲器持有至安全数据解析器 3000 的输入。

[0501] 应当理解，任何其他适当的能力配置都可以对数据解析器 3000 内置。任何附加的特征都可以内置，并且示出的任何特征都可以移除、使得更鲁棒、使得更不鲁棒，或者可以以任何适当的方式进行修改。缓冲器 3008 和 3010 同样也仅是示例性的，并且可以以任何适当的方式被修改、移除、或增加。

[0502] 以软件、硬件或两者实现的任何适当的模块都可以被安全数据解析器 3000 调用或者对其调用。如果期望，甚至是内置到安全数据解析器 3000 中的能力也可以由一个或多个外部模块代替。如图所示，一些外部模块包括随机数生成器 3012，密码反馈密钥生成器 3014，散列算法 3016，任何一种或多种类型的加密 3018，和密钥管理 3020。应当理解，这些仅是示例性的外部模块。任何其他适当的模块可以用于增加或替换所示出的这些。

[0503] 在安全数据解析器 3000 外部的密码反馈密钥生成器 3014 可以为每个安全数据解析操作生成一个唯一密钥，或随机数（例如，使用随机数生成器 3012），以用作操作的种子

值，其将原始会话密钥的大小（例如，128、256、512 或 1024 位的值）扩展为等于将被解析和拆分的数据的长度的值。任何适当的算法都可以用于密码反馈密钥生成，包括例如 AES 密码反馈密钥生成算法。

[0504] 为有助于将安全数据解析器 3000 和其外部模块（即，安全数据解析器层 3026）集成到应用层 3024（例如，电子邮件应用，数据库应用等），可以使用可利用例如 API 功能调用的包装层。有助于将安全数据解析器层 3026 集成到应用层 3024 的任何其他适当的配置都可以被使用。

[0505] 图 31 示例性地示出了在写入（例如，到存储装置）、插入（例如，在数据库字段中）、或传输（例如，经过网络）命令在应用层 3024 中被发布时，图 30 的配置可以如何被使用。在步骤 3100，要保护的数据被识别，并对安全数据解析器进行调用。该调用通过包装层 3022，其中在步骤 3102，包装层 3022 将在步骤 3100 被识别的输入数据流入至组装数据缓冲器 3008。还是在步骤 3102，任何适当的份信息、文件名、任何其他适当的信息、或其任意组合都可以被存储（例如，作为包装层 3022 处的信息 3106）。根据本发明，安全数据处理器 3000 然后解析和拆分作为来自组装数据缓冲器 3008 的输入而得到的数据。其将数据份输出到拆分份缓冲器 3010。在步骤 3104，包装层 3022 从存储的信息 3106 中得到任何适当的份信息（即，在步骤 3102 由包装 3022 存储的）和份位置（一个或多个）（例如，来自一个或多个配置文件）。包装层 3022 然后适当地将输出的份（从拆分份缓冲器 3010 获取的）写入（例如，写入一个或多个存储装置，在网络上传送等）。

[0506] 图 32 示例性地示出了当读取（例如，从存储装置）、选择（例如，从数据库字段）、或接收（例如，从网络）发生时，图 30 的配置可以被如何使用。在步骤 3200，要被恢复的数据被识别并且从应用层 3024 对安全数据解析器 3000 进行调用。在步骤 3202，从包装层 3022 获取任何适当的份信息以及确定份位置。包装层 3022 将在步骤 3200 识别的数据部分载入拆分份缓冲器 3010。根据本发明，安全数据解析器 3000 然后处理这些份（例如，如果四个份中仅三个可用，则安全数据解析器 3000 的冗余能力可以用于仅使用该三个份来恢复原始数据）。恢复的数据然后被存储在组装数据缓冲器 3008 中。在步骤 3204，应用层 3022 将存储在组装数据缓冲器 3008 中的数据转换为其原始数据格式（如果必要）并以其原始格式提供原始数据到应用层 3024。

[0507] 应当理解，图 31 中示出的原始数据的解析和拆分以及图 32 中示出的将多个数据部分恢复为原始数据都仅仅是示例性的。任何其他适当的处理、部件、或两者都可以用于增加或替换示出的这些。

[0508] 图 33 是根据本发明的一个实施例的用于将原始数据解析和拆分为两个或更多个数据部分的示例性处理流程的框图。如图所示，期望被解析和拆分的原始数据是明文 3306（即，单词“SUMMIT”被用作示例）。应当理解，根据本发明，任何其他类型的数据都可以被解析和拆分。生成会话密钥 3300。如果会话密钥 3300 的长度和原始数据 3306 的长度不一致，则可以生成密码反馈会话密钥 3304。

[0509] 在一种适当的方法中，原始数据 3306 可以在解析、拆分、或两者之前被加密。例如，如图 33 所示，原始数据 3306 可以与任何适当的值（例如，与密码反馈会话密钥 3304，或与任何其他适当的值）进行异或。应当理解，任何其他适当的加密技术都可以用于替换或增加示出的 XOR 技术。还应当理解，尽管以逐字节的操作示出了图 33，但该操作可以以位级

或任何其他适合的级来进行。应当进一步理解，如果期望，原始数据 3306 的无论什么都不需要加密。

[0510] 得到的加密数据（或原始数据，如果没有加密发生）然后被散列以确定如何在输出存储桶 (output bucket)（例如，在示出的示例中有四个）之间拆分加密（或原始）数据。在示例性示例中，散列按字节进行且为密码反馈会话密钥 3304 的函数。应当理解，这仅仅是示例性的。如果期望，散列可以以位级来执行。散列还可以是密码反馈会话密钥 3304 之外任何其他适当的值的函数。在另一种适当的方法中，不需要使用散列。相反，用于拆分数据的任何其他适当的技术可以被采用。

[0511] 图 34 是根据本发明的一个实施例，用于从原始数据 3306 的两个或更多个解析或拆分部分恢复原始数据 3306 的示例性处理流程的框图。该处理包括作为密码反馈会话密钥 3304 的函数，反向（即，与图 33 中的处理相反）散列各部分以恢复加密的原始数据（或原始数据，如果在解析和拆分之前没有加密的话）。然后加密密钥可以用于恢复原始数据（即，在示出的示例中，密码反馈会话密钥 3304 用于通过将其与加密数据进行异或以解密该异或加密）。这恢复了原始数据 3306。

[0512] 图 35 示出了在图 33 和 34 的示例中，可以如何实现位拆分。可以使用散列（例如，作为密码反馈会话密钥的函数，作为任何其他适合值的函数）以确定一个位值，以该位值拆分数据的每个字节。应当理解，这仅是实现以位级拆分的一个示例性的方式。可以使用任何其他适当的技术。

[0513] 应当理解，在此引用的散列函数可以参考任何适当的散列算法进行。这些包括例如 MD5 和 SHA-1。不同的散列算法可以在不同时候被本发明的不同部件使用。

[0514] 在已经根据上述示例性过程或通过任何其他过程或算法确定拆分点之后，可以确定哪些数据部分要附加每个左段和右段。可以使用任何适当的算法来进行确定。例如，在一种适当的方法中，可以创建所有可能的分配的表（例如，以左段和右段的目的地的配对的形式），藉此，可以通过对会话密钥、密码反馈会话密钥、或任何其他适当的随机或伪随机值（其可以被生成和扩展为原始数据的大小）中对应数据使用任何适当的散列函数来确定对于每个左段和右段的目的地份值。例如，可以进行随机或伪随机值中对应字节的散列函数。散列函数的输出用于确定从所有目的地组合的表中选择哪些目的地配对（即，一个用于左段，一个用于右段）。基于该结果，拆分数据单元的每个段都附加到由作为散列函数结果所选择的表值所指示的相应的两个份。

[0515] 根据本发明，冗余信息可以附加到数据部分以允许使用少于所有的数据部分来恢复原始数据。例如，如果期望四个部分中的两个足以用于恢复数据，则份中的其他数据可以相应地以例如循环 (round-robin) 方式附加到每个份（例如，原始数据的大小为 4MB，则份 1 取得其自己的份以及份 2 和 3 的份；份 2 取得其自己的份以及份 3 和 4 的份；份 3 取得其自己的份以及份 4 和 1 的份；份 4 取得其自己的份以及份 1 和 2 的份）。根据本发明可以使用任何这样适当的冗余。

[0516] 应当理解，根据本发明，任何其他适当的解析和拆分方法可以用于从原始数据集生成各个数据部分。例如，可以逐位地、随机或伪随机地处理解析和拆分。可以使用随机或伪随机值（例如，会话密钥，密码反馈会话密钥等），藉此对于原始数据中每一位，对于随机或伪随机值中对应数据的散列函数的结果都可以指示出各个位要附加到哪个份。在一种适

当的方法中，随机或伪随机值可以被生成为或扩展为原始数据大小的 8 倍，从而可以针对原始数据的每一位，对随机或伪随机值的对应字节执行散列函数。根据本发明可以使用以逐位级解析和拆分数据的任何其他适当的算法。应当进一步意识到，根据本发明，冗余数据可以附加到数据份上，例如，以上面描述的方式。

[0517] 在一种适当的方法中，解析和拆分不需要是随机或伪随机的。相反，可以使用用于解析和拆分数据的任何适当的确定性算法。例如，可以使用将原始数据分解为顺序的份作为解析和拆分算法。另一个示例是逐位地解析和拆分原始数据，以循环方式将每个相应位顺序地附加到数据份。应当进一步意识到，根据本发明，冗余数据可以附加到数据份上，例如，以上面描述的方式。

[0518] 在本发明的一个实施例中，在安全数据解析器生成原始数据的多个部分后，为恢复原始数据，所生成的部分中的特定的一个或多个可以是强制性的。例如，如果这些部分中的一个被用作认证份（例如，保存在物理令牌装置上），并且如果正在使用安全数据解析器的容错特征（即，少于所有的部分是必需的以恢复原始数据），则即使安全数据解析器可能能够访问原始数据的足够数量的部分，在恢复原始数据之前可能仍需要该存储在物理令牌装置上的认证份。应当理解，基于例如应用、数据类型、用户、任何其他适当的因素、或其任意组合，可以要求任何数量和类型的特定份。

[0519] 在一种适当的方法中，安全数据解析器或安全数据解析器的一些外部部件可以加密原始数据的一个或多个部分。可能需要提供和解密这些加密部分，以恢复原始数据。不同的加密部分可以用不同的加密密钥来加密。例如，该特征可以被用于实现更安全的“二人法则”，藉此第一用户需要使用第一加密来加密特定份，而第二用户需要使用第二加密密钥来加密特定份。为访问原始数据，两个用户都需要具有他们各自的加密密钥，并提供原始数据中他们各自的部分。在一种适当的方法中，公钥可以用于加密可以是恢复原始数据所需的强制份的一个或多个数据部分。私钥则可以用于解密该份以用于恢复原始数据。

[0520] 可以使用任何这样的适当的范例，其在少于所有的份被需要以恢复原始数据的情况下使用强制份。

[0521] 在本发明的一个适当的实施例中，到有限数量的数据份的数据分配可以被随机或伪随机地处理，从而从统计角度，任何特定数据份接收到特定数据单元的概率等于剩余份中任何一个将接收到该数据单元的概率。因此，每个数据份将具有近似相等数量的数据位。

[0522] 根据本发明的另一个实施例，有限数量的数据份中的每一个不需要具有从原始数据解析和拆分中接收数据单元的相等的概率。相反，某一个或多个份可以比其余份具有更高或更低的概率。因此，相对于其他份，某些份在位大小方面可能更大或者更小。例如，在两份的情况中，一个份可以具有 1％的接收数据单元的概率，而第二份具有 99％的概率。因此可知，一旦数据单元已经被安全数据解析器在两个份之间分配，第一份应该具有近似 1％的数据而第二份有 99％。根据本发明，可以使用任何适当的概率。

[0523] 应当理解，安全数据解析器也可以被编程为根据精确（或接近精确）的百分比将数据分配到各个份。例如，安全数据解析器可以被编程为将 80％的数据分配到第一份，将剩余的 20％数据分配到第二份。

[0524] 根据本发明的另一个实施例，安全数据解析器可以生成数据份，其中一个或多个具有预定大小。例如，安全数据解析器可以将原始数据拆分为数据部分，其中一个部分为精

确的256位。在一种适当的方法中，如果不可能生成具有所需大小的数据部分，则安全数据解析器可以补充该部分使其为正确大小。可以使用任何适当的大小。

[0525] 在一种适当的方法中，一个数据部分的大小可以是加密密钥、拆分密钥、任何其他适当的密钥、或任何其他适当的数据元素的大小。

[0526] 如前所讨论的，安全数据解析器在解析和拆分数据时使用密钥。为了清晰和简洁，这些密钥在此被称为“拆分密钥”。例如，前面介绍的会话主密钥为一种类型的拆分密钥。同样，如前所讨论的，拆分密钥可以在由安全数据解析器生成的数据份内被保护。用于保护拆分密钥的任何适当算法都可以用于在数据份当中保护它们。例如，Shamir 算法可以用于保护拆分密钥，藉此可以生成可用于重建拆分密钥的信息并附加到数据份。根据本发明，可以使用任何其他这样的适当算法。

[0527] 类似地，根据例如 Shamir 算法的任何适当的算法，任何适当的加密密钥可以在一个或多个数据份内被保护。例如，用于在解析和拆分之前加密数据集的加密密钥、用于在解析和拆分之后加密数据部分的加密密钥、或这两者，都可以使用例如 Shamir 算法或任何其他适当的算法来保护。

[0528] 根据本发明的一个实施例，全转换或不转换 (All or NothingTransform，AoNT)，例如全包转换，可以用于通过转换拆分密钥、加密密钥、任何其他适当的数据元素、或其任意组合来进一步保护数据。例如，根据本发明，用于在解析和拆分之前加密数据集的加密密钥可以通过 AoNT 算法而被转换。转换后的加密密钥然后可以根据例如 Shamir 算法或任何其他适当的算法在数据份之间被分配。如本领域技术人员已知的，为重建加密密钥，加密的数据集必须被恢复（例如，根据本发明，如果使用冗余，则不必使用所有数据份）以访问关于基于 AoNT 的转换的必要信息。当原始加密密钥被恢复时，其可以用于解密加密数据集以恢复原始数据集。应当理解，可以将本发明的容错特征与 AoNT 特征结合使用。即，冗余数据可以被包括在数据部分中，从而少于所有的数据部分为恢复加密数据集所必需。

[0529] 应当理解，AoNT 可以应用于加密密钥，其中该加密密钥用于在解析和拆分之后加密数据部分，作为在解析和拆分之前加密和 AoNT 与数据集相对应的各个加密密钥的替换或补充。同样，AoNT 可以应用于拆分密钥。

[0530] 在本发明的一个实施例中，根据本发明所使用的加密密钥、拆分密钥、或两者都可以使用例如工作组密钥来进一步加密，以向保护的数据集提供额外的安全等级。

[0531] 在本发明的一个实施例中，可以提供审计模块以在安全数据解析器被调用以拆分数据时进行跟踪。

[0532] 图 36 示出了根据本发明，使用安全数据解析器的部件的可能选项 3600。选项的每个组合在以下概括，并使用来自图 36 的适当的步骤号来标记。安全数据解析器本质上是模块化的，允许在图 36 中示出的每个功能块中使用任何已知的算法。例如，可以使用诸如 Blakely 之类的其他密钥拆分（例如，机密共享）算法来代替 Shamir，或 AES 加密可以用诸如 3 重 DES 之类的其他已知加密算法来代替。图 36 的示例中示出的标记仅示出了本发明的一个实施例中使用的一种可能的算法组合。应当理解，可以使用任何适当的算法或算法组合来代替所标记的算法。

[0533] 1)3610，3612，3614，3615，3616，3617，3618，3619

[0534] 在步骤 3610 使用之前加密过的数据，该数据最终可以被拆分为预定数量的份。如

果拆分算法需要密钥，则可以在步骤 3612 使用加密安全伪随机数生成器生成拆分加密密钥。拆分加密密钥在步骤 3615 被密钥拆分为具有容错的预定数量的份之前，可以在步骤 3614 可选地使用全转换或不转换 (AoNT) 转换为转换拆分密钥。然后在步骤 3616，数据可以被拆分为预定数量的份。在步骤 3617 可以使用容错方案以允许从少于总数量的份中再生数据。一旦创建了份，在步骤 3618，认证 / 完整性信息可以被嵌入到份中。在步骤 3619，每个份可以可选地被后加密。

[0535] 2)3111,3612,3614,3615,3616,3617,3618,3619

[0536] 在一些实施例中，可以使用由用户或外部系统提供的加密密钥来加密输入数据。在步骤 3611 提供外部密钥。例如，可以从外部密钥存储器提供密钥。如果拆分算法要求密钥，则在步骤 3612 可以使用密码安全伪随机数生成器来生成拆分加密密钥。拆分密钥在步骤 3615 被密钥拆分为具有容错的预定数量的份之前，在步骤 3614 可以可选地使用全转换或不转换 (AoNT) 而被转换为转换拆分加密密钥。然后在步骤 3616，数据被拆分为预定数量的份。在步骤 3617 可以使用容错方案来允许数据从少于总数量的份中再生。一旦创建了份，在步骤 3618，认证 / 完整性信息可以被嵌入到份中。在步骤 3619，每个份可以可选地被后加密。

[0537] 3)3612,3613,3614,3615,3612,3614,3615,3616,3617,3618,3619

[0538] 在一些实施例中，在步骤 3612 可以使用加密安全伪随机数生成器来生成加密密钥以转换数据。在步骤 3613 可以使用所生成的加密密钥来加密数据。在步骤 3614，加密密钥可以可选地使用全转换或不转换 (AoNT) 而被转换为转换加密密钥。然后在步骤 3615，转换加密密钥和 / 或所生成的加密密钥可以被拆分为具有容错的预定数量的份。如果拆分算法需要密钥，则在步骤 3612 可以使用加密安全伪随机数生成器来生成拆分加密密钥。拆分密钥在步骤 3615 被密钥拆分为具有容错的预定数量的份之前，在步骤 3614 可以可选地使用全转换或不转换 (AoNT) 而被转换为转换拆分加密密钥。然后在步骤 3616，数据可以被拆分为预定数量的份。在步骤 3617 可以使用容错方案来允许从少于总数量的份中再生数据。一旦创建了份，在步骤 3618，认证 / 完整性信息将被嵌入到份中。然后在步骤 3619，每个份可以被可选地后加密。

[0539] 4)3612,3614,3615,3616,3617,3618,3619

[0540] 在一些实施例中，数据可以被拆分为预定数量的份。如果拆分算法需要密钥，则在步骤 3612 可以使用加密安全伪随机数生成器来生成拆分加密密钥。拆分密钥在步骤 3615 被密钥拆分为具有容错的预定数量的份时，在步骤 3614 可以可选地使用全转换或不转换 (AoNT) 而被转换为转换拆分密钥。然后在步骤 3616，数据可以被拆分。在步骤 3617 可以使用容错方案以允许从少于总数量的份中再生数据。一旦创建了份，在步骤 3618，认证 / 完整性信息可以被嵌入到份中。在步骤 3619，每个份可以可选地被后加密。

[0541] 虽然以上四种选项组合被优选地用于本发明的一些实施例中，但是任何其他适当的特征、步骤或选项的组合都可以与安全数据解析器一起用于其它实施例。

[0542] 安全数据解析器可以通过有助于物理分离来提供灵活的数据保护。数据可以首先被加密，然后被拆分为具有“n 中 m 个”容错的份。这允许在少于总数量的份可用时再生原始信息。例如，一些份可能在传输中丢失或损坏。如以下更详细的讨论，该丢失或损坏的份可以基于附加到份的容错或完整性信息被重建。

[0543] 为了创建份，多个密钥可以可选地被安全数据解析器使用。这些密钥可以包括以下的一个或多个：

[0544] 预加密密钥：当选择预加密各份时，外部密钥可以被传递到安全数据解析器。该密钥可以在外部被生成和存储在密钥存储器中（或其他位置），并可以被用于可选地在数据拆分之前加密数据。

[0545] 拆分加密密钥：该密钥可以在内部被生成，并被安全数据解析器使用以在拆分之前加密数据。然后该密钥可以使用密钥拆分算法被安全地存储在各份中。

[0546] 拆分会话密钥：该密钥不与加密算法一起使用；相反，在选择随机拆分时其可以被用作数据分割算法的密钥。在使用随机拆分时，拆分会话密钥可以在内部生成，并由安全数据解析器用于将数据分割为份。该密钥可以使用密钥拆分算法被安全地存储在各份中。

[0547] 后加密密钥：在选择后加密份时，外部密钥可以被传递到安全数据解析器，并被用于后加密单独的份。该密钥可以在外部被生成和存储在密钥存储器中或其他适当位置。

[0548] 在一些实施例中，当以此方式使用安全数据解析器保护数据时，只有当所有所需的份和外部加密密钥都存在时，该信息才可以被重新组装。

[0549] 图 37 示出了在一些实施例中使用本发明的安全数据解析器的示例性总处理 3700。如上所述，安全数据解析器 3706 的两个适当的功能可以包括加密 3702 和备份 3704。这样，在一些实施例中，安全数据解析器 3706 可以与 RAID 或备份系统或硬件或软件加密引擎集成在一起。

[0550] 与安全数据解析器 3706 相关联的主密钥处理可以包括一个或多个预加密处理 3708、加密 / 转换处理 3710、密钥安全处理 3712、解析 / 分配处理 3714、容错处理 3716、份认证处理 3716、和后加密处理 3720。这些处理可以以多种适当的顺序或组合来执行，如图 36 所详细示出的。所使用的处理的组合和顺序可以取决于特定的应用或用途，期望的安全等级，是否期望可选的预加密、后加密、或两者，期望的冗余，底层或集成系统的能力或性能，或任何其他适当的因素或因素的组合。

[0551] 示例性处理 3700 的输出可以是两个或更多个份 3722。如上所述，在一些实施例中数据可以被随机（或伪随机）地分配至这些份中的每一个。在其他实施例中，可以使用确定性算法（或随机、伪随机、和确定性算法的某种适当组合）。

[0552] 除了对信息资产的个体保护，有时要求在不同的用户组或利益团体之间共享信息。则这可能有必要控制对用户组内个体的份的访问，或在那些用户之间共享仅允许该组的成员重新组装各个份的凭证。为此，在本发明的一些实施例中，可以对组成员配置工作组密钥。工作组密钥应当被保护和保持为机密，因为工作组密钥的泄露可能潜在地允许组外人员访问信息。用于工作组密钥的配置和保护的一些系统和方法在下面讨论。

[0553] 工作组密钥概念允许通过加密存储在份中的密钥信息来增强信息资产保护。一旦执行该操作，即使所有需要的份和外部密钥被发现，攻击者也没有希望不访问工作组密钥就重建信息。

[0554] 图 38 示出了用于在份内存储密钥和数据分量的示例性框图 3800。在图 3800 的示例中，可选的预加密和后加密步骤被省略，尽管这些步骤可以被包括在其他实施例中。

[0555] 拆分数据的简化处理包括在加密阶段 3802 使用加密密钥 3804 加密数据。根据本发明，加密密钥 3804 的部分然后可以被拆分并存储在份 3810 中。拆分加密密钥 3806 的部

分也可以存储在份 3810 中。使用拆分加密密钥，然后数据 3808 被拆分并存储在份 3810 中。

[0556] 为恢复数据，根据本发明，拆分加密密钥 3806 可以被取回并恢复。然后拆分操作可以被反向进行以恢复密文。加密密钥 3804 也可以被取回并恢复，并且密文可以使用加密密钥被解密。

[0557] 当使用工作组密钥时，以上处理可以被稍微地改变以使用工作组密钥保护加密密钥。然后在加密密钥被存储在份中之前，加密密钥可以用工作组密钥来保护。修改的步骤在图 39 的示例性框图 3900 中被示出。

[0558] 使用工作组密钥拆分数据的简化处理包括在阶段 3902 首先使用加密密钥加密数据。然后在阶段 3904，加密密钥可以使用工作组密钥被加密。然后该用工作组密钥加密的加密密钥可以被拆分为部分，并与份 3912 一起存储。拆分密钥 3908 也可以被拆分并存储在份 3912 中。最后，数据 3910 的部分使用拆分密钥 3908 被拆分并存储在份 3912 中。

[0559] 为恢复数据，根据本发明，拆分密钥可以被取回并恢复。然后根据本发明，拆分操作可以反向进行以恢复密文。加密密钥（使用工作组密钥加密）可以被取回并恢复。然后加密密钥可以使用工作组密钥被解密。最后，密文可以使用加密密钥被解密。

[0560] 有多种配置和保护工作组密钥的安全方法。为特定应用选择使用哪种方法取决于多个因素。这些因素可以包括所要求的安全等级、成本、便利、和工作组中的用户数量。在一些实施例中所使用的一些常用技术被提供如下：

[0561] 基于硬件的密钥存储

[0562] 基于硬件的方案通常为加密 / 解密密钥的安全性提供最强的保证。基于硬件的存储方案的示例包括防篡改密钥令牌装置，其在便携式装置（例如，智能卡 / 加密狗）或非便携密钥存储外围装置中存储密钥。这些装置被设计为防止由未授权方简单地复制密钥资料。密钥可以由可信任的权力机构生成并分配给用户，或在硬件内生成。此外，很多密钥存储系统提供多因素认证，其中使用密钥需要访问物理对象（令牌）和通行码或生物识别。

[0563] 基于软件的密钥存储

[0564] 虽然专用的基于硬件的存储可能被期望用于高安全性配置或应用，但其他配置可以选择直接在本地硬件（例如，盘、RAM 或非易失性 RAM 存储器，例如 USB 驱动器）上存储密钥。这对于内部攻击提供了较低的保护等级，或在一些情况下攻击者能够直接访问加密机器。

[0565] 为保护盘上的密钥，基于软件的密钥管理通常通过在源于其他认证度量的组合的密钥下以加密形式存储密钥来保护密钥，这些认证度量包括：口令和通行码、其他密钥的存在（例如，来自基于硬件的方案）、生物识别、或上述的任意适当组合。由这样的技术提供的安全等级可以从由一些操作系统（例如，MS Windows 和 Linux）提供的相对弱的密钥保护机制，延伸到使用多因素认证的更鲁棒的方案。

[0566] 本发明的安全数据解析器可以在多种应用和技术中被有利地使用。例如，电子邮件系统、RAID 系统、视频广播系统、数据库系统、或任何其他适当的系统可以以任何适当的等级集成安全数据解析器。如前讨论的，应当理解，安全数据解析器也可以被集成以保护和容错任何类型的通过任何传输介质（包括例如有线、无线、或物理传输介质）而移动的数据。如一个示例，因特网协议语音 (VoIP) 应用可以使用本发明的安全数据解析器以解决

与 VoIP 中经常出现的回声和延迟有关的问题。可以通过使用容错消除对丢包的网络重试的需要，其保证了包的递送，即使有预定数量的份丢失。数据包（例如，网络包）还可以以最小延迟和缓冲地被有效拆分和“即时”恢复，得到一种用于各种类型的移动数据的综合方案。安全数据解析器可以对网络数据包、网络语音包、文件系统数据块、或任何其他适当的信息单元起作用。除了与 VoIP 应用集成，安全数据解析器还可以与文件共享应用（例如，对等文件共享应用）、视频广播应用、电子投票或调查应用（其可以实现电子投票协议和盲签 (blind signature)，例如 Sensus 协议）、电子邮件应用、或任何其他需要或期望安全通信的网络应用集成。

[0567] 在一些实施例中，对移动的网络数据的支持可以通过本发明的安全数据解析器以两个不同阶段来提供——头生成阶段和数据分割阶段。简化的头生成处理 4000 和简化的数据分割处理 4010 分别在图 40A 和 40B 中示出。这些处理中的一个或两个可以对网络包、文件系统块、或任何其他适当的信息执行。

[0568] 在一些实施例中，在网络包流初始化时头生成处理 4000 可以被执行一次。在步骤 4002，可以生成随机（或伪随机）拆分加密密钥 K。然后在 AES 密钥包装步骤 4004，拆分加密密钥 K 可以可选地被加密。尽管 AES 密钥包装可以用在一些实施例中，但任何适当的密钥加密或密钥包装算法可以用在其它实施例中。AES 密钥包装步骤 4004 可以对整个拆分加密密钥 K 操作，或者拆分加密密钥可以被解析为若干块（例如，64 位的块）。然后如果期望，AES 密钥包装步骤 4004 可以对拆分加密密钥的块操作。

[0569] 在步骤 4006，机密共享算法（例如，Shamir）可以用于将拆分加密密钥 K 拆分为密钥份。然后每个密钥份可以被嵌入到输出份中之一（例如，在份的头部）。最后，份完整性模块和（可选的）后认证标签（例如，MAC）可以被附加到每个份的头块。每个头块可以被设计为适合于在单个的数据包内。

[0570] 在头生成完成（例如，使用简化的头生成处理 4000）之后，安全数据解析器可以使用简化的数据分割处理 4010 来进入数据分割阶段。在步骤 4012，使用拆分加密密钥 K 加密流中的每个输入的数据包或数据块。在步骤 4014，可以对从步骤 4012 得到的密文计算份完整性信息（例如，散列 H）。例如，可以计算 SHA-256 散列。然后在步骤 4016，根据本发明，数据包或数据块可以使用上述的数据分割算法中的一个被分割成为两个或更多个数据份。在一些实施例中，可以分割数据包或数据块，从而每个数据份包括加密的数据包或数据块的基本随机分配。然后完整性信息（例如，散列 H）可以被附加到每个数据份。在一些实施例中，可选的后认证标签（例如，MAC）也可以被计算并附加到每个数据份。

[0571] 每个数据份可以包括元数据，其可以是允许正确重建数据块或数据包所必需的。该信息可以被包括在份头中。元数据可以包括诸如加密密钥份、密钥标识、份临时随机数、签名 /MAC 值、和完整性块之类的信息。为最大化带宽效率，元数据可以以压缩二进制格式被存储。

[0572] 例如，在一些实施例中，份头包括明文头组块，其不被加密并可以包括诸如 Shamir 密钥份、每个会话临时随机数、每个份临时随机数、密钥标识符（例如，工作组密钥标识符和后认证密钥标识符）之类的元素。份头还可以包括加密头组块，其使用拆分加密密钥被加密。头中还可以包括完整性头组块，其可以包括对于任意数量的先前块（例如，先前的两个块）的完整性检查。任何其他适当的值或信息也可以被包括在份头中。

[0573] 如图41的示例性份格式4100中所示，头块4102可以与两个或更多个输出块4104相关联。每个头块，例如头块4102，可以被设计为适合于在单个网络数据包内。在一些实施例中，在头块4102从第一位置被传输到第二位置之后，然后可以传输输出块。可替换地，头块4102和输出块4104可同时并行传输。传输可以在一个或多个类似或不相类似的通信通道上发生。

[0574] 每个输出块可以包括数据部分4106和完整性/真实性部分4108。如上所述，每个数据份可以使用包括加密的、预分割数据的份完整性信息（例如，SHA-256散列）的份完整性部分被保护。为在恢复时校验输出块的完整性，安全数据解析器可以比较每个份的份完整性块，然后反转拆分算法。然后可以通过份散列来校验恢复的数据的散列。

[0575] 尽管以上描述了安全数据解析器的一些常用的应用，但是应当清楚地理解，本发明可以与任何网络应用集成以增加安全性、容错、匿名、或上述的任何适当组合。

[0576] 本发明的另一个方面是一种公共接口（例如，通用的应用程序接口（“API”)），以用于跨多个平台、接口、或平台和接口两者而安全地创建、存储、和管理加密密钥。公共接口可以由硬件、软件、固件、或硬件、软件、和固件的任意组合提供。例如，在一些实施例中，密钥提供者应用、实用程序、或机制可以在安全地存储或管理加密密钥的那些系统和安全数据解析器之间提供公共接口。密钥提供者应用、实用程序、或机制还可以允许各种密钥提供者和密钥存储器之间的互用性。

[0577] 可以实现关于公共接口的多个设计目标，包括可证明的安全性和对标准密码术的依赖。例如，加密密钥可以使用用于密钥管理的被证明的行业标准机制来被安全地处理和传递。任何密钥提供者应用或机制的完整性也可以通过不允许公共接口与底层密钥提供者应用、实用程序、或机制的特征或功能进行接口来维护。在一些实施例中，单个接口可以被用于所有密钥提供者。在一些实施例中，底层密钥提供者程序或安全数据解析器的改变可以不影响公共接口。密钥提供者或安全数据解析器引擎的接口改变可以对公共接口仅具有很小的影响。例如，公共接口（或提供公共接口的应用、实用程序、或机制）的设计可以高度模块化。个性的提供者二进制程序(binary)的合成和公共接口用以接口到这些二进制程序的机制，在本质上都可以被高度模块化。

[0578] 在一些实施例中，公共接口包括在应用或其他接口或平台之间生成、存储、取回和传递加密密钥的功能。这些功能可以包括下列示例性功能中的一个或多个：

[0579] OPEN(打开)-OPEN功能可以用于通过与应用或接口相关联的应用接口二进制程序来发起与目标应用或接口的通信。

[0580] CLOSE(关闭)-CLOSE功能可以用于停止与目标应用或接口的通信。

[0581] GENERATE KEY(生成密钥)-GENERATE KEY功能可以用于创建加密密钥。

[0582] RETRIEVE KEY(取回密钥)-RETRIEVE KEY功能可以用于访问要在安全数据解析器引擎的拆分或恢复功能中使用的密钥。

[0583] STORE KEY(存储密钥)-STORE KEY功能可以用于将密钥放入安全数据解析器引擎的密钥存储器内（或放在其他目标介质上，例如诸如USB闪存或智能卡之类的可移除介质）。

[0584] DELETE KEY(删除密钥)-DELETE KEY功能可以用于从安全数据解析器引擎的密钥存储器中移除密钥。

[0585] 在一些实施例中，每个应用或接口可以被配置为由使用个性的应用接口二进制程序的密钥提供者应用、实用程序、或机制来使用。该二进制程序（例如，程序）可以将来自公共接口的请求转换为进行请求的应用或接口的个性请求。这样，密钥提供者应用、实用程序或机制可以无缝地（并安全地）支持跨大量不同平台和接口的加密密钥创建、删除、存储、和管理。

[0586] 为使加密密钥可被安全数据解析器引擎访问，密钥可以首先被放置在安全解析器引擎的密钥存储器中，并被识别供使用。每个应用或接口可以具有在应用或接口的应用接口二进制程序中指定的个性化请求。因为每个应用被增加到密钥提供者应用、实用程序或机制中，所以其可以被提供有转换模块，其详细说明了公共接口和独特应用接口请求之间的转换。

[0587] 图42示出了用于使用公共接口来管理加密密钥的示例性处理4200。在步骤4202，管理加密密钥的请求可以从第一机制或接口被接收。例如，生成、取回、存储、或删除加密密钥的请求可以以第一接口格式被接收。该请求可以例如通过安全通信信道或经受保护的通信会话而被接收。如果该请求通过诸如因特网之类的网络被接收，则可以实施网络安全协议（例如 SSL、TLS、SSH、或 IPsec）。

[0588] 在步骤4204，所接收的请求可以被转换为公共接口请求。例如，该请求可以从第一接口格式被转换为上述的一个或多个公共接口功能。在步骤4206，公共接口请求可以被认证。例如，用户或发起的网络地址可以使用任何认证协议或加密握手而被加密认证。该请求也可以通过校验该请求是有效公共接口格式或源于授权源而被认证。例如，在一些实施例中，有效请求可以被加密签名，在该种情况下，步骤4206可以包括校验与该请求相关联的加密签名。这可以帮助防止其中无效形成的公共接口请求可能尝试绕开认证的第三方攻击。安全数据解析器（或执行安全数据解析器的系统或机制）也可维护授权客户或进行请求的接口的表。客户或进行请求的接口可以使用任何适当的认证协议而被认证。

[0589] 在步骤4208，可以确定公共接口请求是否已经被认证。例如，在一些实施例中，成功认证之后，可以提供认证令牌。对于来自相同的接口或进行请求的客户的连续请求，可以提交认证令牌来代替发起新的认证会话。认证令牌也可以与某个截止日期、截止时间、或两者相关联。这样，认证令牌可以在发布之后的某个预定时间长度之后到期。密钥提供者应用、实用程序、或机制可以通过从系统时钟或计时器访问当前时间来实施与认证令牌相关联的所有的截止日期和时间。在其他实施例中，每个请求可以被认证，而不理会有效和未过期的认证令牌的存在。如果该请求还未被认证，则示例性处理4200可返回步骤4202。

[0590] 如果在步骤4208，已确定请求已经被认证，则在步骤4210可以访问安全数据解析器的加密密钥。如上所述，加密密钥可以以各种方式被存储或保护。例如，安全数据解析器的所有或一些密钥可以被保护在标准密钥存储器、单独的硬件密钥存储器、智能卡、定制密钥存储器、单独的数据库表中，或保护在一个或多个加密份内。

[0591] 在访问安全数据解析器的加密密钥后，在步骤4212可以执行公共接口请求。例如，可以执行上述的一个或多个公共接口功能。这些功能可以生成新的密钥、取回已有的密钥、存储密钥、或从安全数据解析器的加密密钥的集合中删除已有的密钥，等等。任何其他适当的命令或动作也可以在步骤4212被执行。在步骤4214，确定公共接口请求是否包括任何返回参数。例如，一些请求可以返回加密密钥、至加密密钥的存储器地址或指针、或任何

其他适当的信息（例如，成功或失败返回代码）。如果在步骤 4214 没有返回参数，则示例性处理 4200 可以返回到步骤 4202 并等待新的请求。

[0592] 如果一个或多个参数要被返回，则在步骤 4216，返回参数可以被转换为与第一接口兼容的格式。例如，应用接口二进制程序可以将返回参数从公共接口格式转换为进行请求的客户的接口格式。在步骤 4218，返回参数可以通过安全通信通道或通信会话被传输至第一接口。如果是通过诸如因特网之类的网络接收请求，则一个或多个返回参数可以通过实施网络安全协议（例如 SSL、TLS、SSH、或 IPsec）的网络连接被传输。

[0593] 此外，考虑到此处的公开，对于本领域的技术人员来说，其他的组合、增加、替换和修改都将是显而易见的。因此，本发明并不旨在被优选实施例的反应所限制，而是提供参考附加的权利要求来限定。

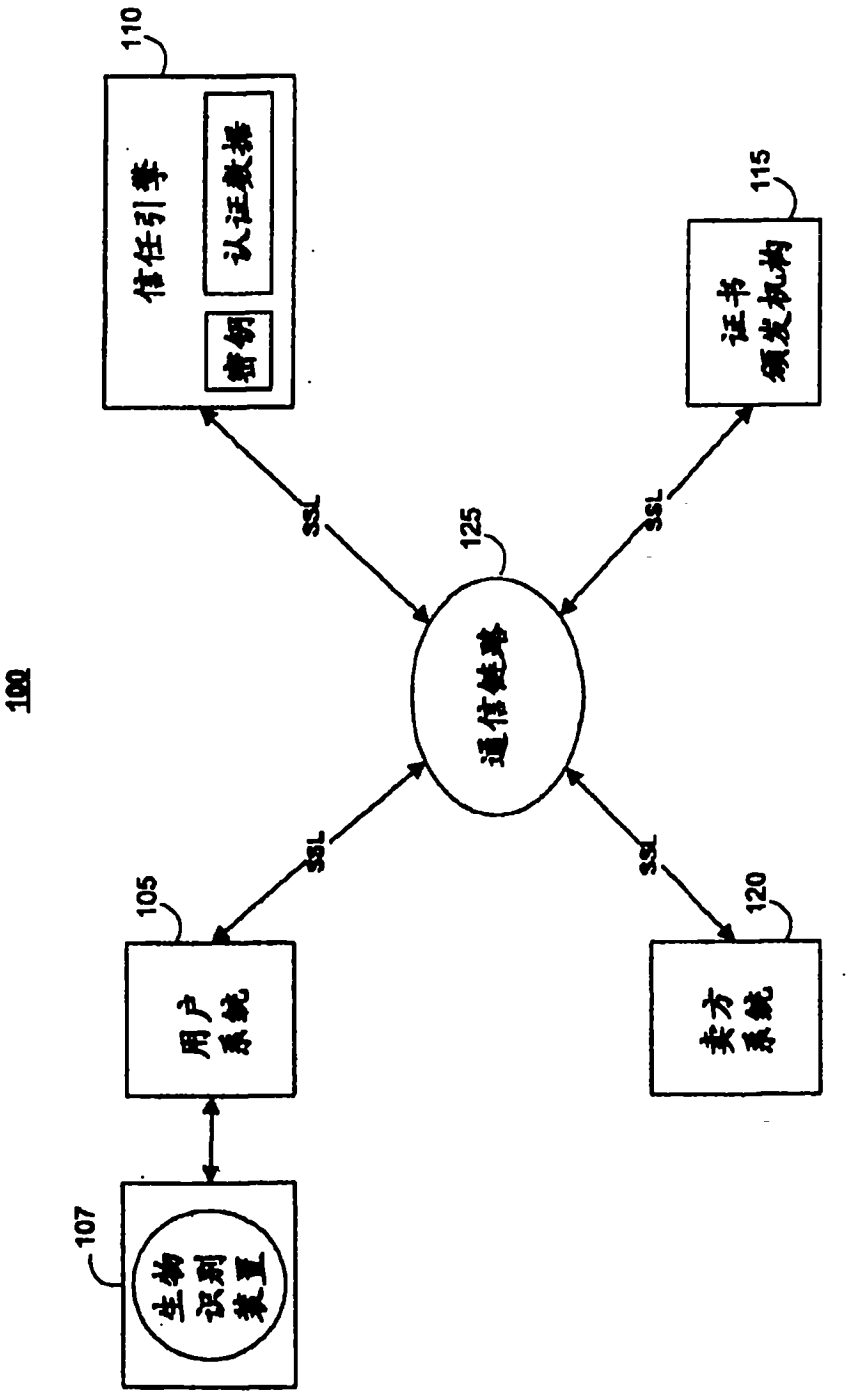

100
110
信任引擎
认证数据
密钥
115
证书颁发机构
SSL
SSL
125
通信链路
SSL
SSL
105
用户系统
120
卖方系统
107
生物识别装置

图 1

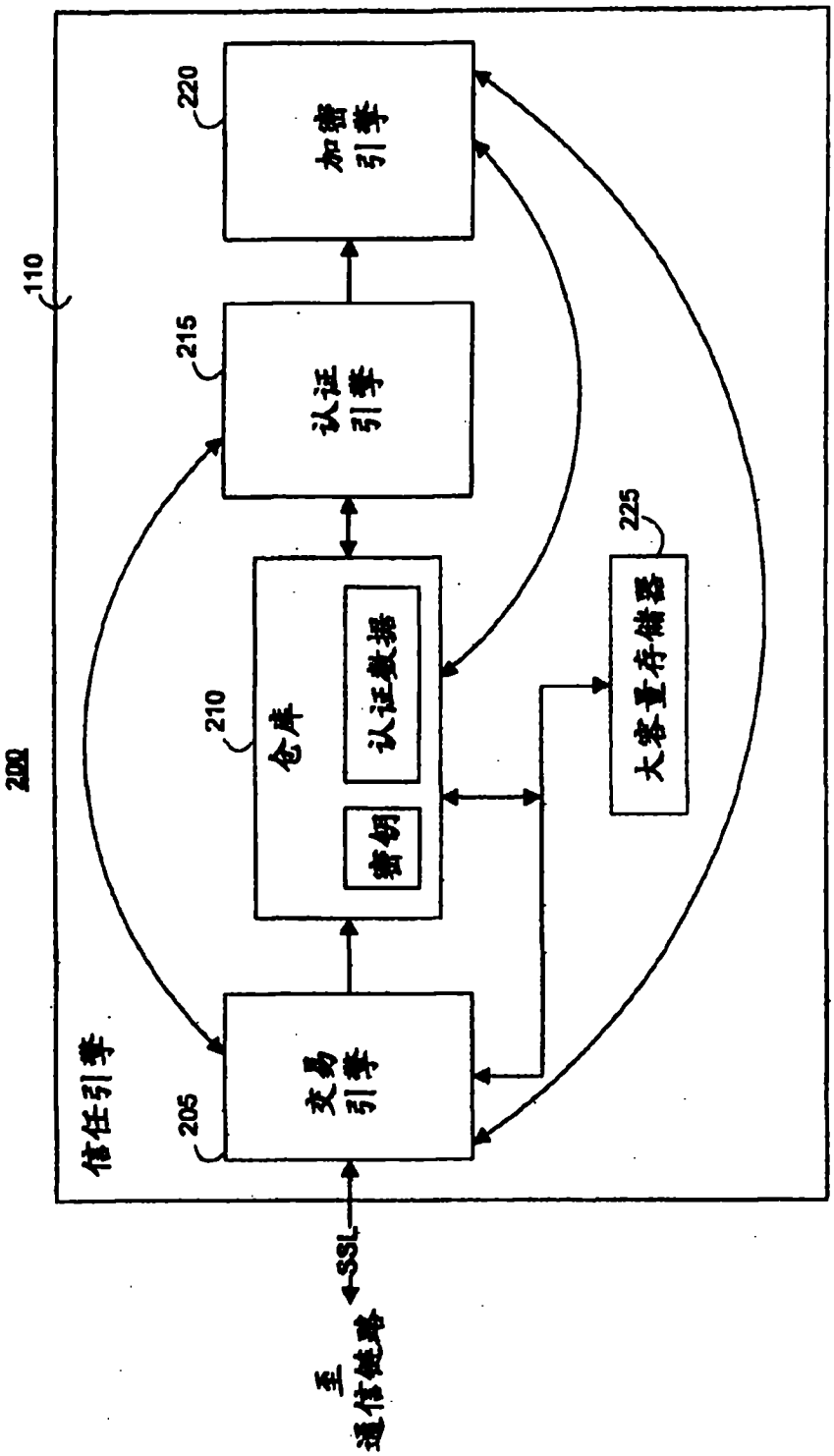

220
加密引擎
110
215
认证引擎
225
210
仓库
认证数据
密钥
大容量存储器
202
信任引擎
205
交易引擎
SSL
至通信链路

图 2

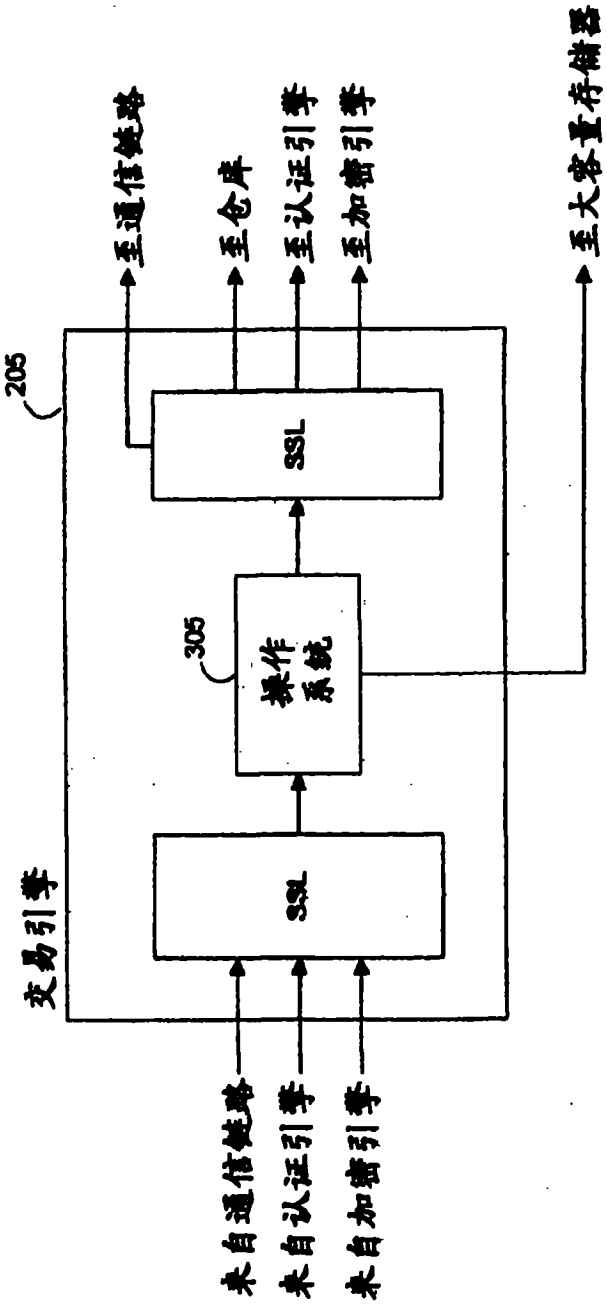

至通信链路
至仓库
至认证引擎
至加密引擎
至大容量存储器
205
SSL
305
操作系统
SSL
交易引擎
来自通信链路
来自认证引擎
来自加密引擎

图 3

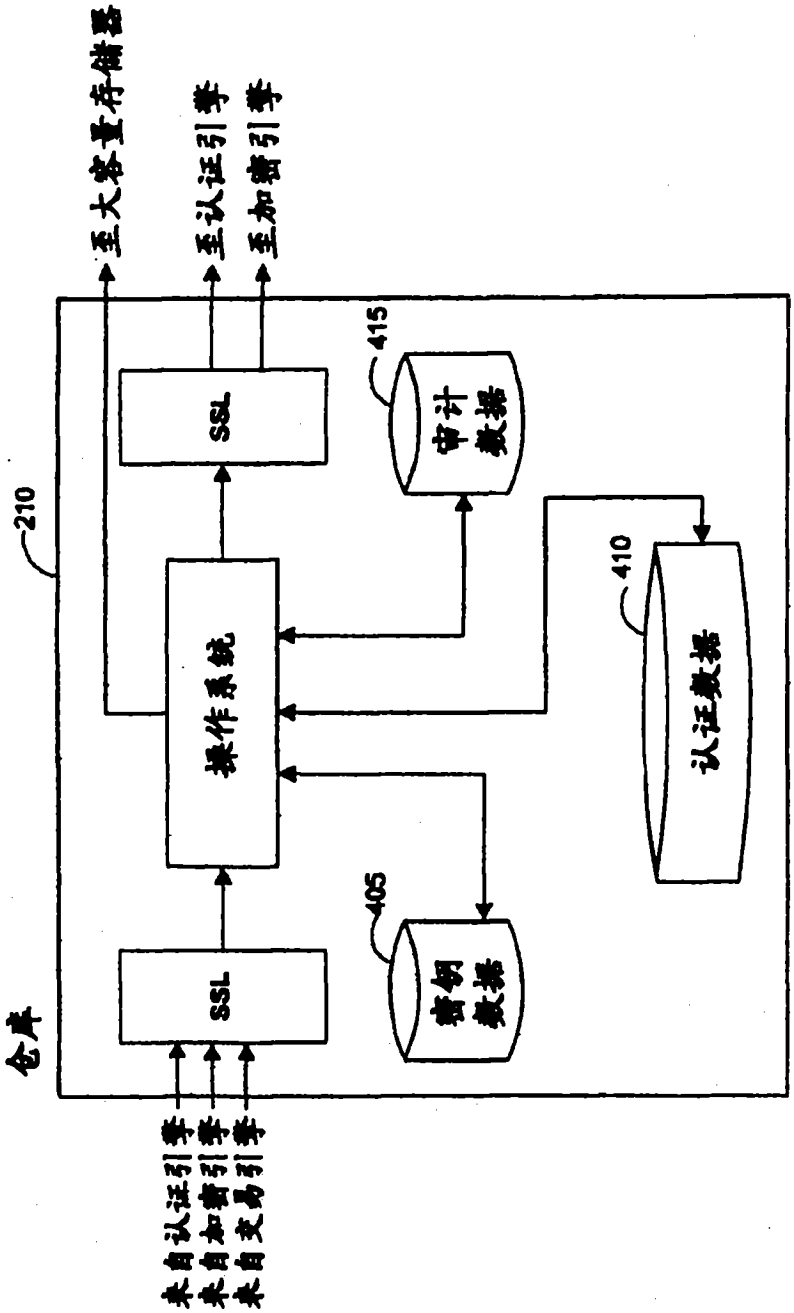

至大容量存储器
至认证引擎
至加密引擎
SSL
415
审计数据
210
操作系统
410
认证数据
405
密钥数据
SSL
仓库
来自认证引擎
来自加密引擎
来自支易引擎

图 4

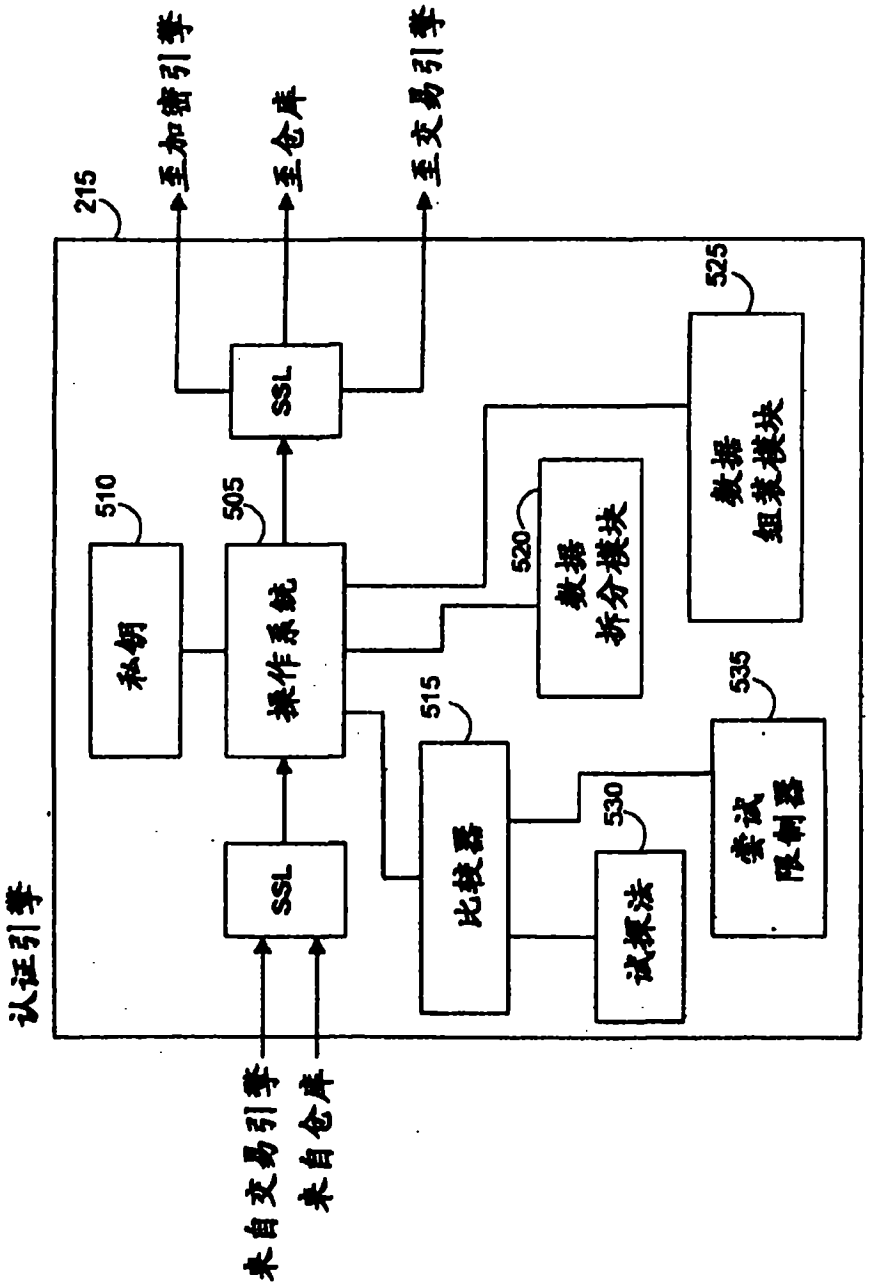

认证引擎
215
至加密引擎
至仓库
至交易引擎
SSL
525
数据
组装模块
510
私钥
505
操作系统
520
数据
拆分模块
515
比较器
530
试探法
535
尝试
限制器
SSL
来自交易引擎
来自仓库

图 5

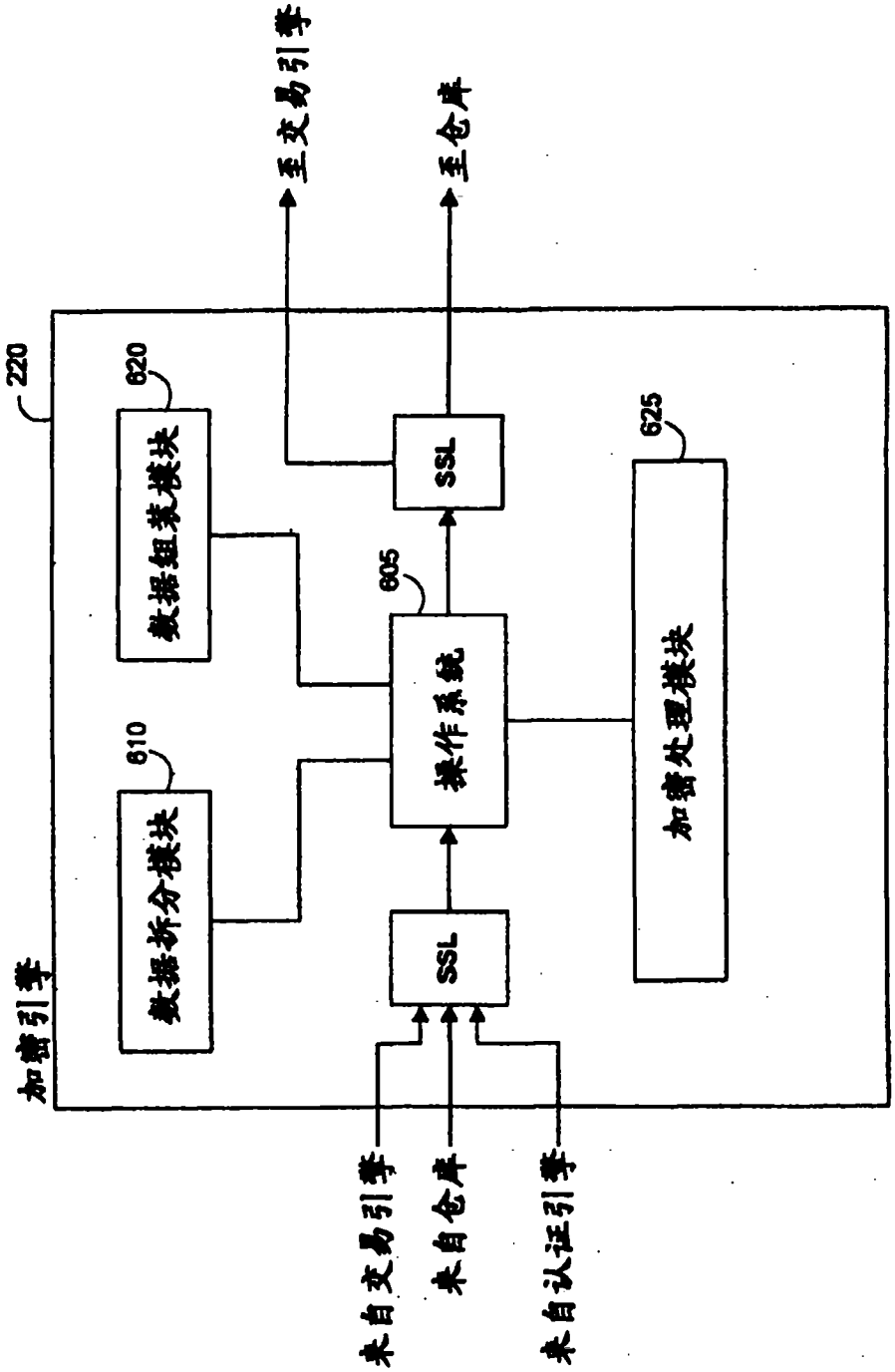

至交易引擎
至仓库
220
620
数据组装模块
SSL
625
605
操作系统
加密处理模块
610
数据拆分模块
SSL
加密引擎
来自交易引擎
来自仓库
来自认证引擎

图 6

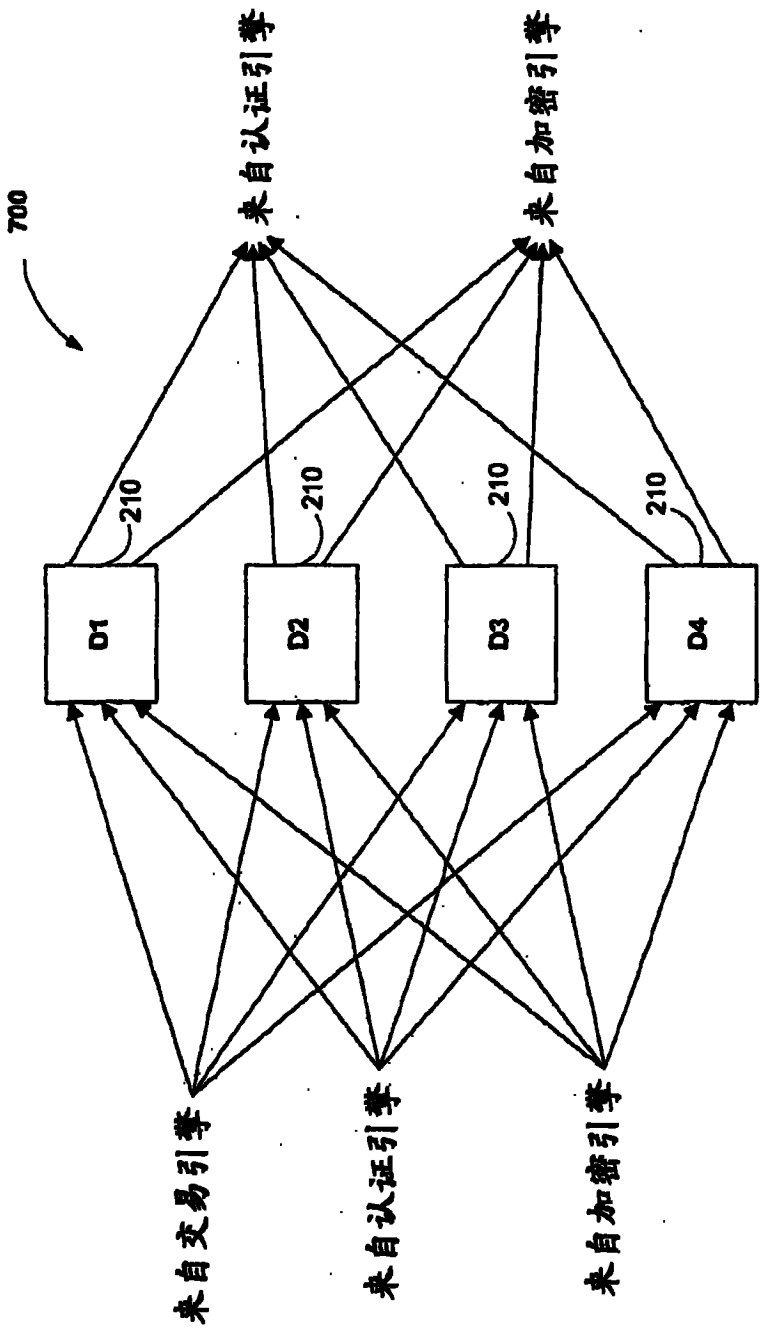

700
来自认证引擎
来自加密引擎
210
210
210
210
D1
D2
D3
D4
来自交易引擎
来自认证引擎
来自加密引擎

图 7

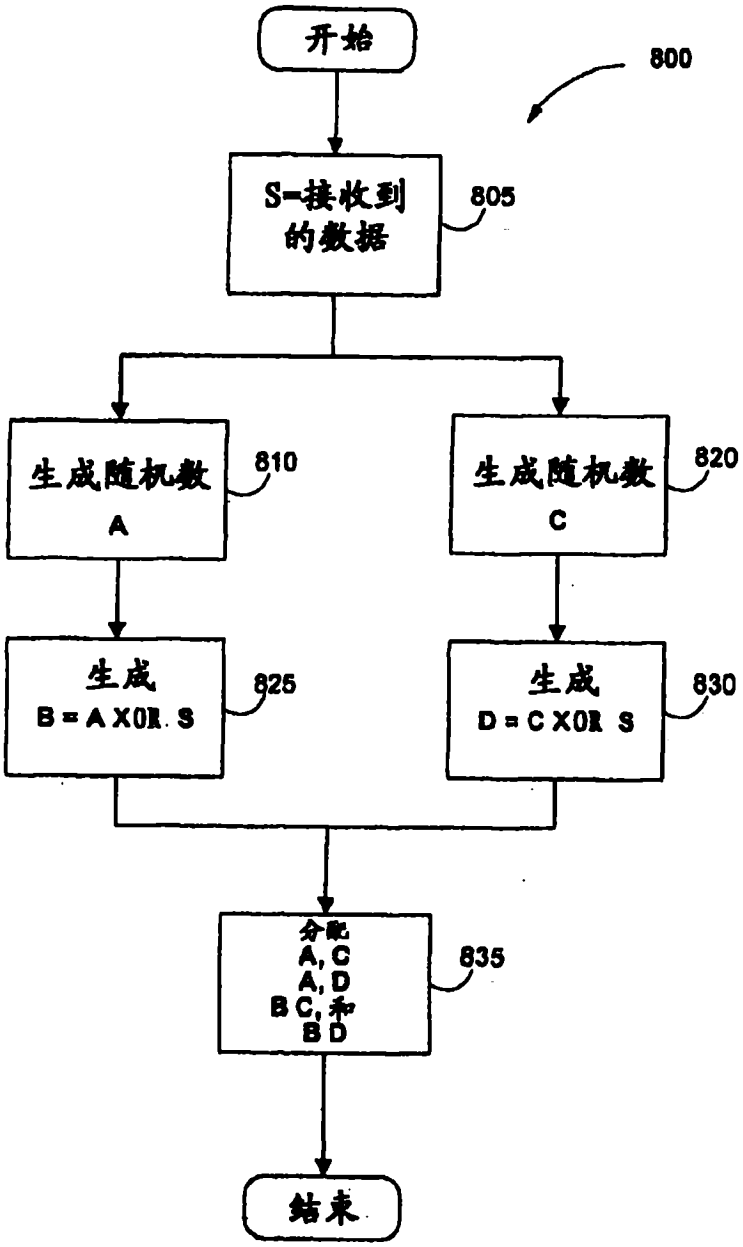

开始
800
S=接收到的数据
805
生成随机数A
810
生成随机数C
820
生成B=A XOR S
825
生成D=C XOR S
830
分配A,C A,D B C,和B D
835
结束

图 8

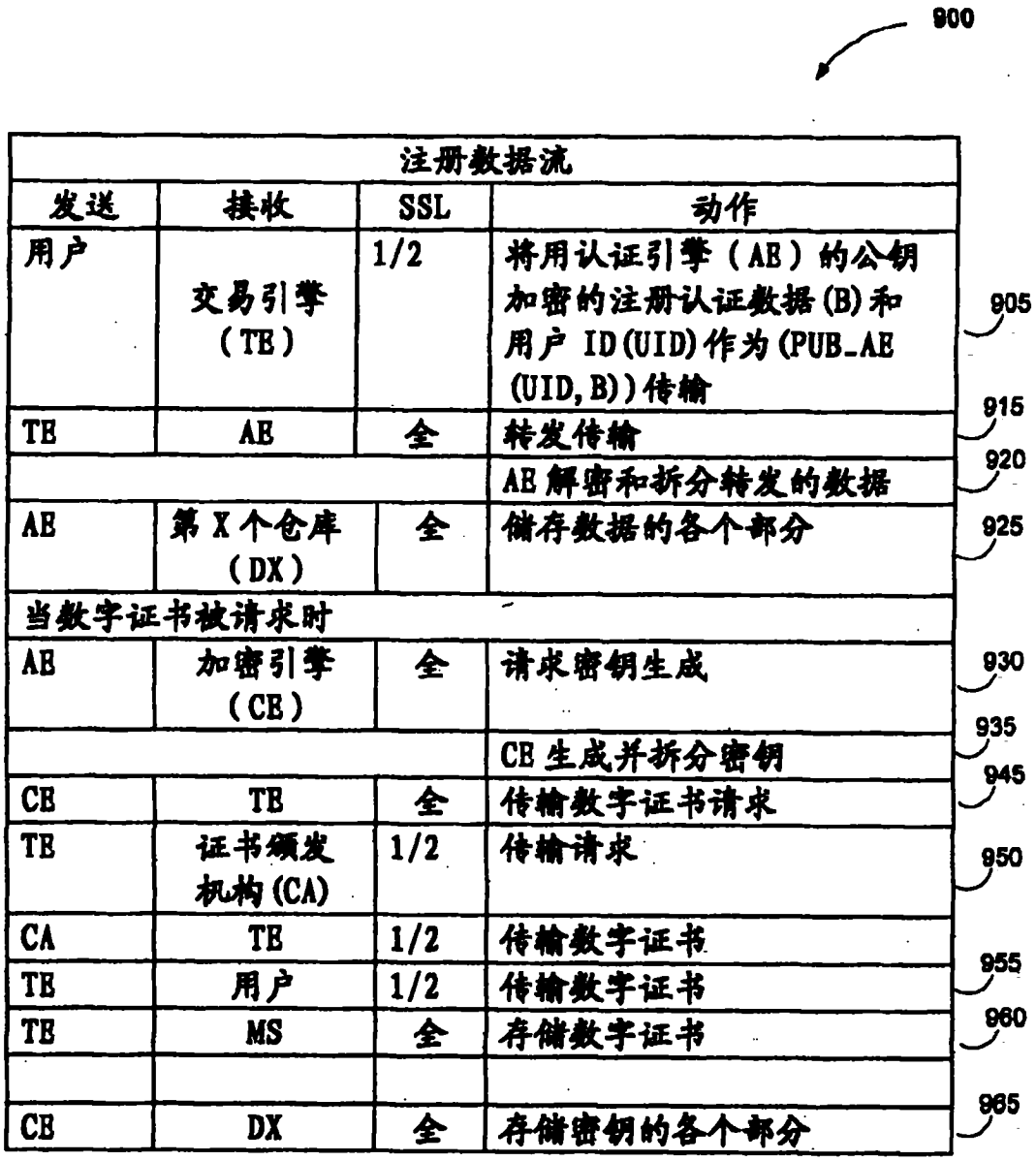

900
注册数据流
发送
接收
SSL
动作
用户
交易引擎（TE）
1/2
将用认证引擎（AE）的公钥加密的注册认证数据(B)和用户 ID(UID)作为(PUB_AE(UID,B))传输
905
TE
AE
全
转发传输
915
AE 解密和拆分转发的数据
920
AE
第 X 个仓库（DX）
全
储存数据的各个部分
925
当数字证书被请求时
AE
加密引擎（CE）
全
请求密钥生成
930
CE 生成并拆分密钥
935
CE
TE
全
传输数字证书请求
945
TE
证书颁发机构(CA)
1/2
传输请求
950
CA
TE
1/2
传输数字证书
TE
用户
1/2
传输数字证书
955
TE
MS
全
存储数字证书
960
CE
DX
全
存储密钥的各个部分
965

图 9，面 A

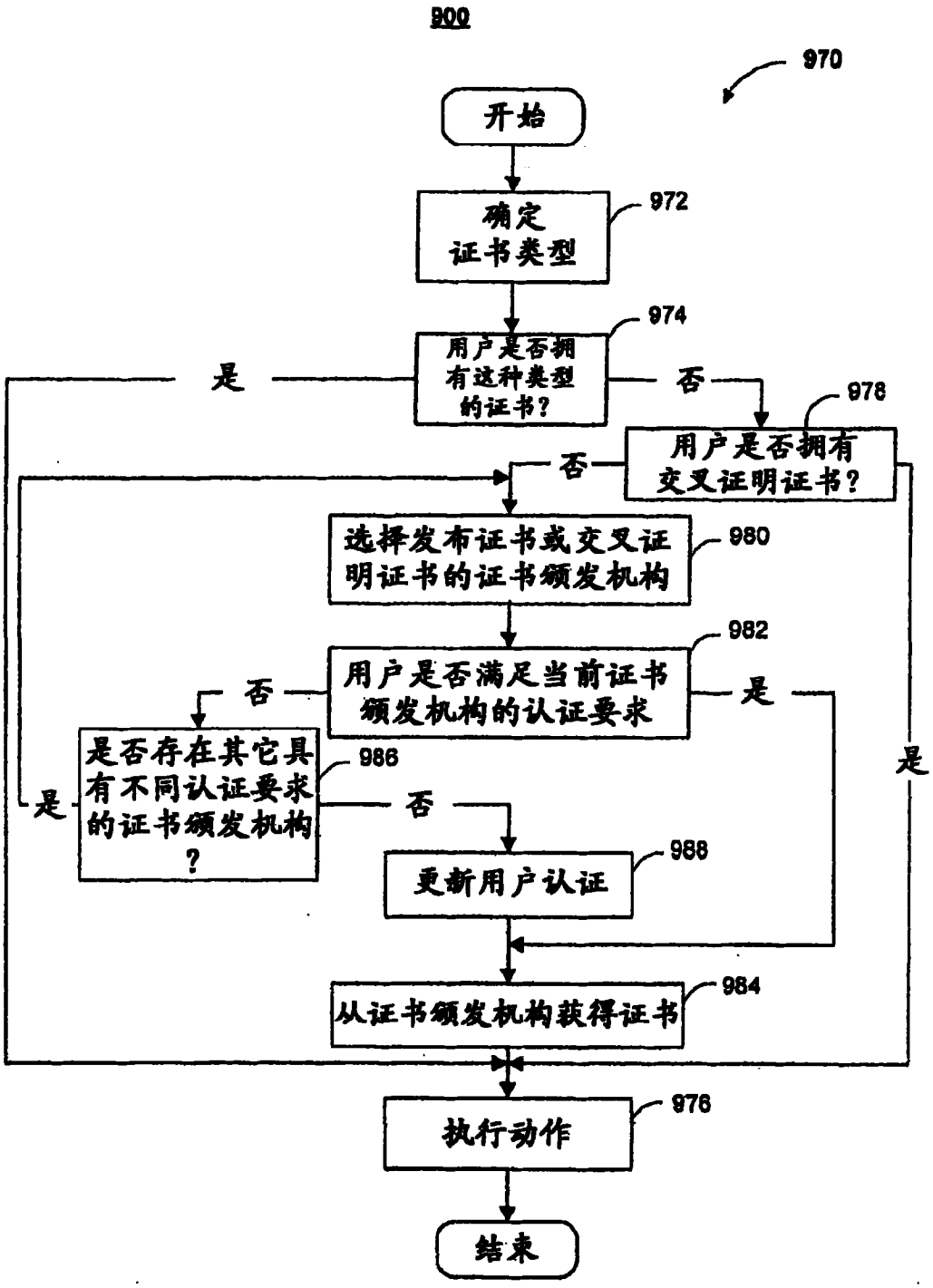

900
970
开始
确定证书类型
972
用户是否拥有这种类型的证书?
974
是
否
用户是否拥有交叉证明证书?
978
否
选择发布证书或交叉证明证书的证书颁发机构
980
用户是否满足当前证书颁发机构的认证要求
982
否
是
是否存在其它具有不同认证要求的证书颁发机构?
986
是
否
更新用户认证
988
从证书颁发机构获得证书
984
是
执行动作
976
结束

图 9,面 B

1000

| | 认证数据流 | | | |
|---|---|---|---|---|
| | 发送 | 接收 | SSL | 动作 |
| 1005 | 用户 | 卖方 | 1/2 | 交易发生，例如选择购买 |
| 1010 | 卖方 | 用户 | 1/2 | 传输交易 ID(TID)和认证请求(AR) |
| | | | | 从用户收集认证数据（B'） |
| 1015 | 用户 | TE | 1/2 | 将包装在认证引擎（AE）的公钥中的TID和B'作为(PUB_AE(TID, B'))传输 |
| 1020 | TE | AE | 全 | 转发传输 |
| | | | | 请求和收集注册认证数据（B） |
| 1025 | 卖方 | 交易引擎（TE） | 全 | 传输 TID, AR |
| 1030 | TE | 大容量存储器(MS) | 全 | 在数据库中创建记录 |
| 1035 | TE | 第 X 个仓库（DX） | 全 | UID, TID |
| 1040 | DX | AE | 全 | 将 TID 和存储在注册处的认证数据部分（BX）作为(PUB_AE(TID, BX))传输 |
| 1045 | | | | AE 组装 B 并与 B' 比较 |
| 1050 | AE | TE | 全 | TID,被填充的AR |
| 1055 | TE | 卖方 | 全 | TID, 是/否 |
| | TE | 用户 | 1/2 | TID, 确认消息 |

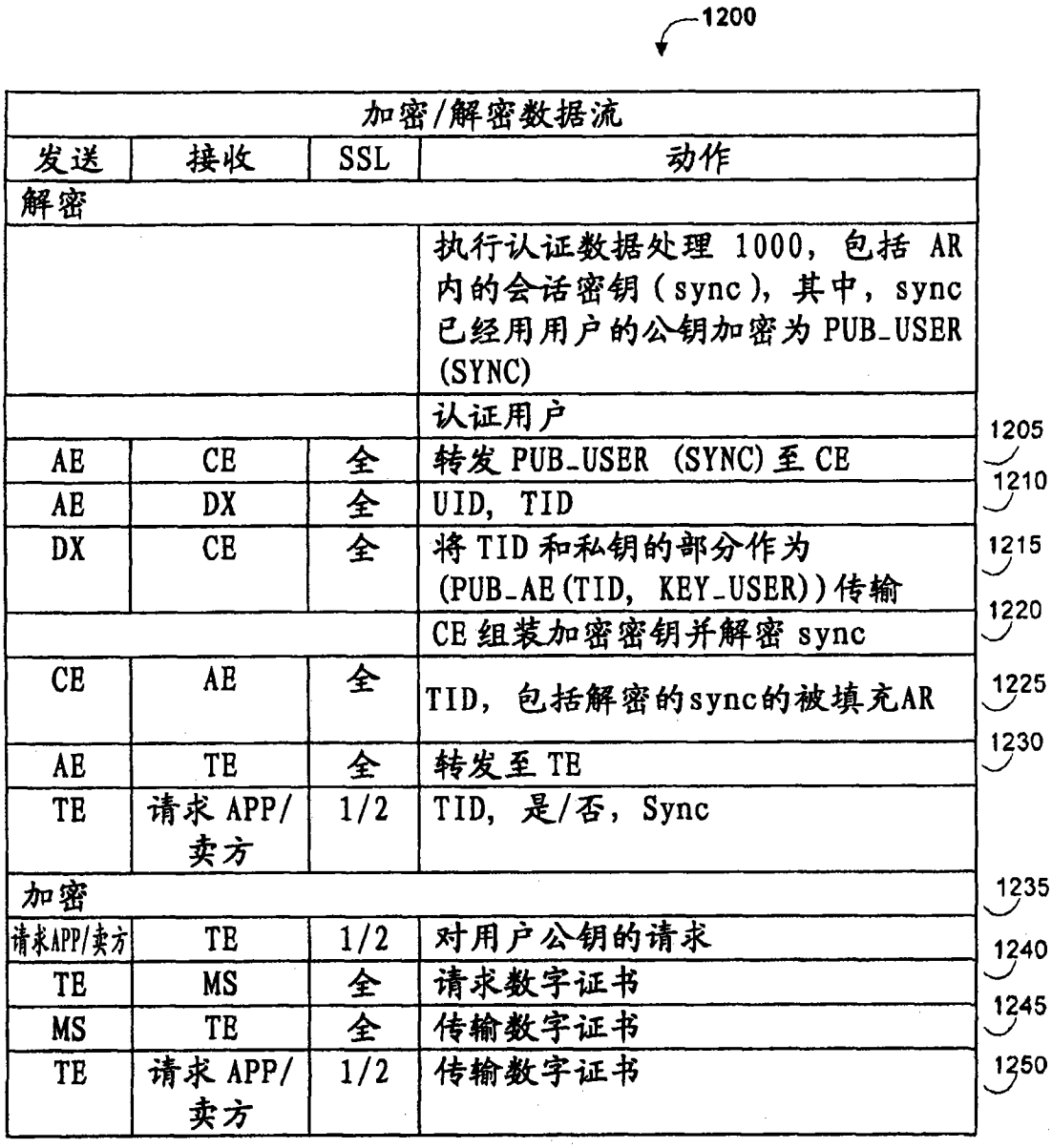
图 10

1100

| 签名数据流 | | | |
|---|---|---|---|
| 发送 | 接收 | SSL | 动作 |
| 用户 | 卖方 | 1/2 | 交易发生，例如达成协定 |
| 卖方 | 用户 | 1/2 | 传输交易识别号(TID)，认证请求(AR)，和协定或消息(M) |
| | | | 从用户收集当前认证数据(B')和用户所接收的消息的散列(h(M')) |
| 用户 | TE | 1/2 | 将包装在认证引擎(AE)的公钥中的TID，B'，AR和h(M')作为(PUB_AE(TID，B'，h(M')))传输 |
| TE | AE | 全 | 转发传输 |
| | | | 收集注册认证数据 |
| 卖方 | 交易引擎(TE) | 全 | 传输UID，TID，AR和消息的散列(h(M')) |
| TE | 大容量存储器(MS) | 全 | 在数据库中创建记录 |
| TE | 第X个仓库(DX) | 全 | UID，TID |
| DX | AE | 全 | 将TID和存储在注册处的认证数据部分(BX)作为(PUB_AE(TID，BX))传输 |
| | | | 原始卖方消息被传输到AE |
| TE | AE | 全 | 传输h(M) |
| | | | AE组装B，与B'比较，并将h(M)与h(M')比较 |
| AE | 加密引擎(CE) | 全 | 请求数字签名和要签名的消息，例如散列的消息 |
| AE | DX | 全 | TID，签署UID |
| DX | CE | 全 | 传输对应于签名方的加密密钥的部分 |
| | | | CE组装密钥和签名 |
| CE | AE | 全 | 传输签名方的数字签名(S) |
| AE | TE | 全 | TID，被填充的AR，h(M)，和S |
| TE | 卖方 | 全 | TID，收据=(TID，是/否，和S)，以及信任引擎的数字签名，例如，用信任引擎的私钥(Priv_TE(h(receipt)))加密的收据的散列 |
| TE | 用户 | 1/2 | TID，确认消息 |

1103 1105 1110 1115 1120 1125 1130 1135 1140

图 11




| 加密/解密数据流 | | | |
|---|---|---|---|
| 发送 | 接收 | SSL | 动作 |
| 解密 | | | |
| | | | 执行认证数据处理 1000，包括 AR 内的会话密钥（sync），其中，sync 已经用用户的公钥加密为 PUB_USER (SYNC) |
| | | | 认证用户 |
| AE | CE | 全 | 转发 PUB_USER (SYNC) 至 CE |
| AE | DX | 全 | UID，TID |
| DX | CE | 全 | 将 TID 和私钥的部分作为 (PUB_AE(TID，KEY_USER)) 传输 |
| | | | CE 组装加密密钥并解密 sync |
| CE | AE | 全 | TID，包括解密的sync的被填充AR |
| AE | TE | 全 | 转发至 TE |
| TE | 请求 APP/卖方 | 1/2 | TID，是/否，Sync |
| 加密 | | | |
| 请求APP/卖方 | TE | 1/2 | 对用户公钥的请求 |
| TE | MS | 全 | 请求数字证书 |
| MS | TE | 全 | 传输数字证书 |
| TE | 请求 APP/卖方 | 1/2 | 传输数字证书 |

图 12

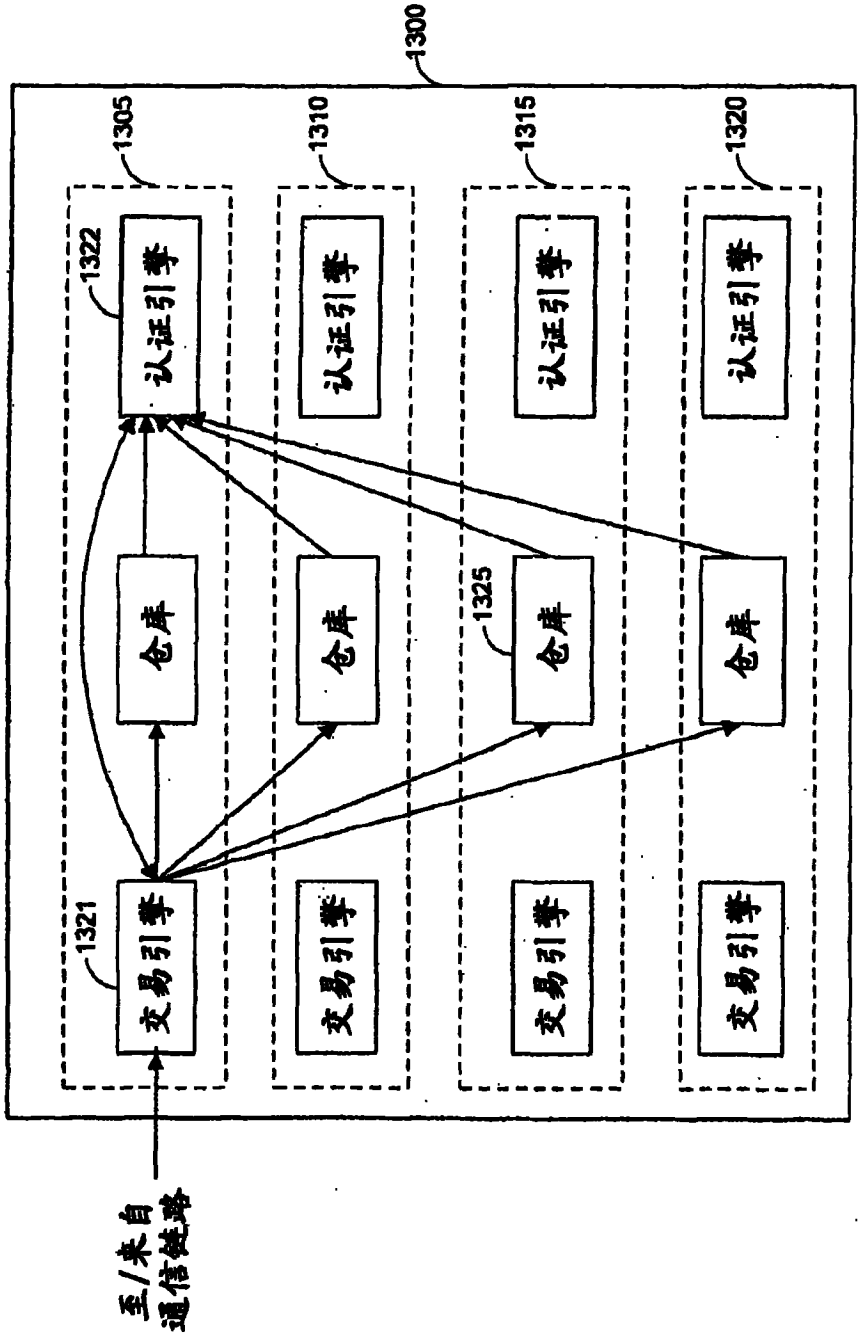



图 13

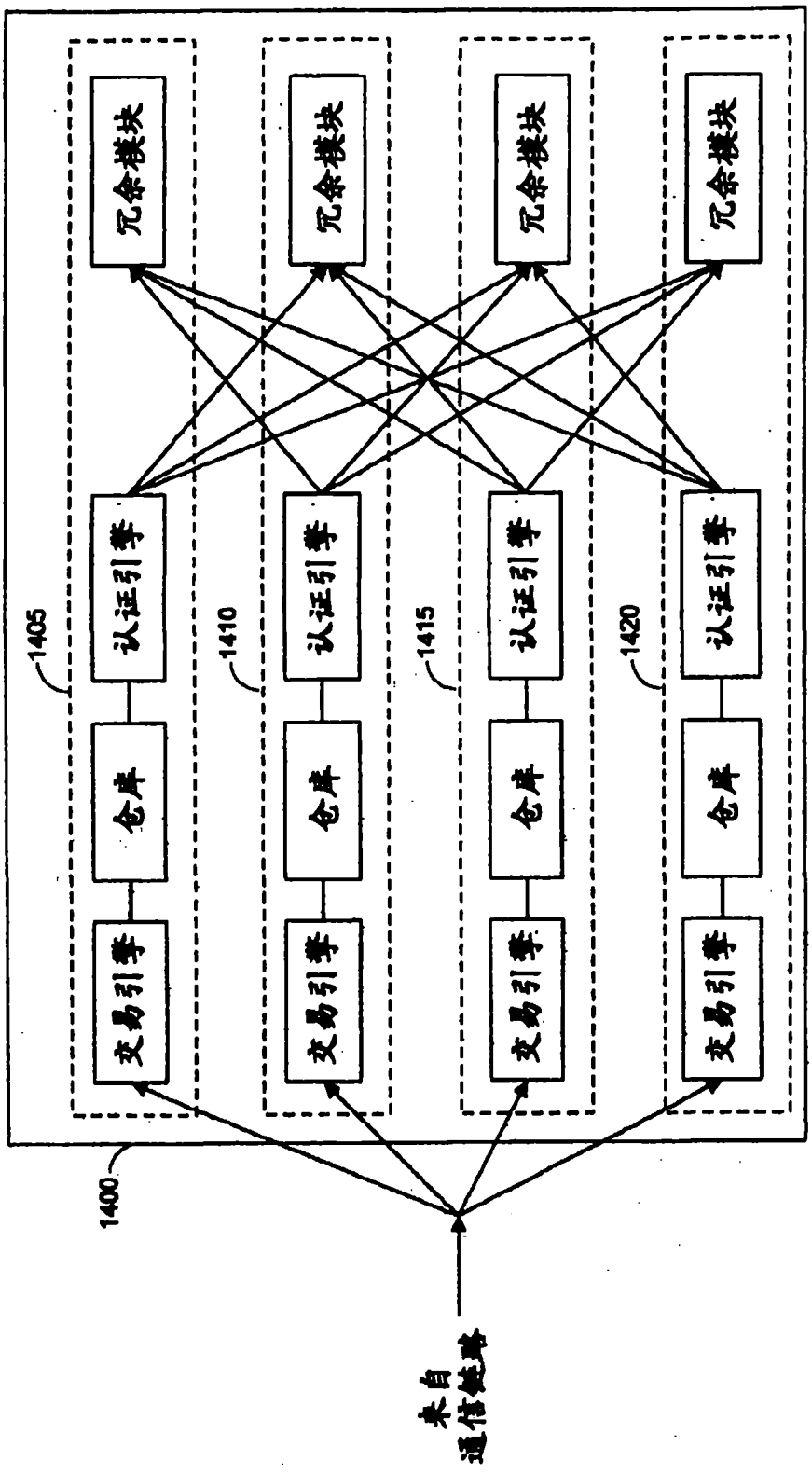

冗余模块
冗余模块
冗余模块
冗余模块
认证引擎
认证引擎
认证引擎
认证引擎
仓库
仓库
仓库
仓库
交易引擎
交易引擎
交易引擎
交易引擎
1405
1410
1415
1420
1400
来自
通信链路

图 14

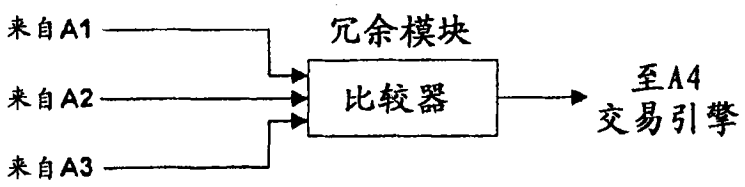



图 15

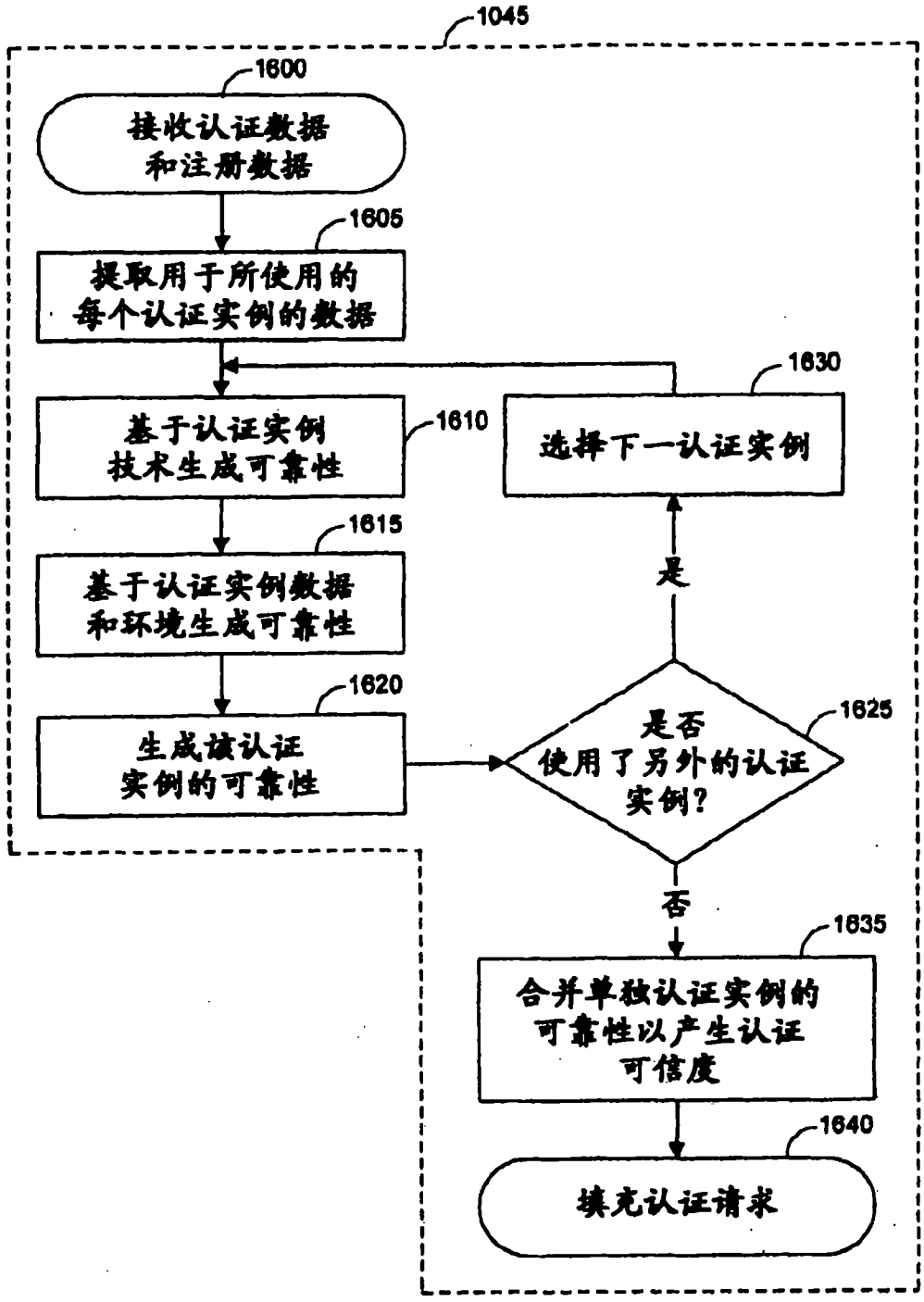

1045
1600
接收认证数据
和注册数据
1605
提取用于所使用的
每个认证实例的数据
1610
基于认证实例
技术生成可靠性
1615
基于认证实例数据
和环境生成可靠性
1620
生成该认证
实例的可靠性
1625
是否
使用了另外的认证
实例?
是
1630
选择下一认证实例
否
1635
合并单独认证实例的
可靠性以产生认证
可信度
1640
填充认证请求

图 16

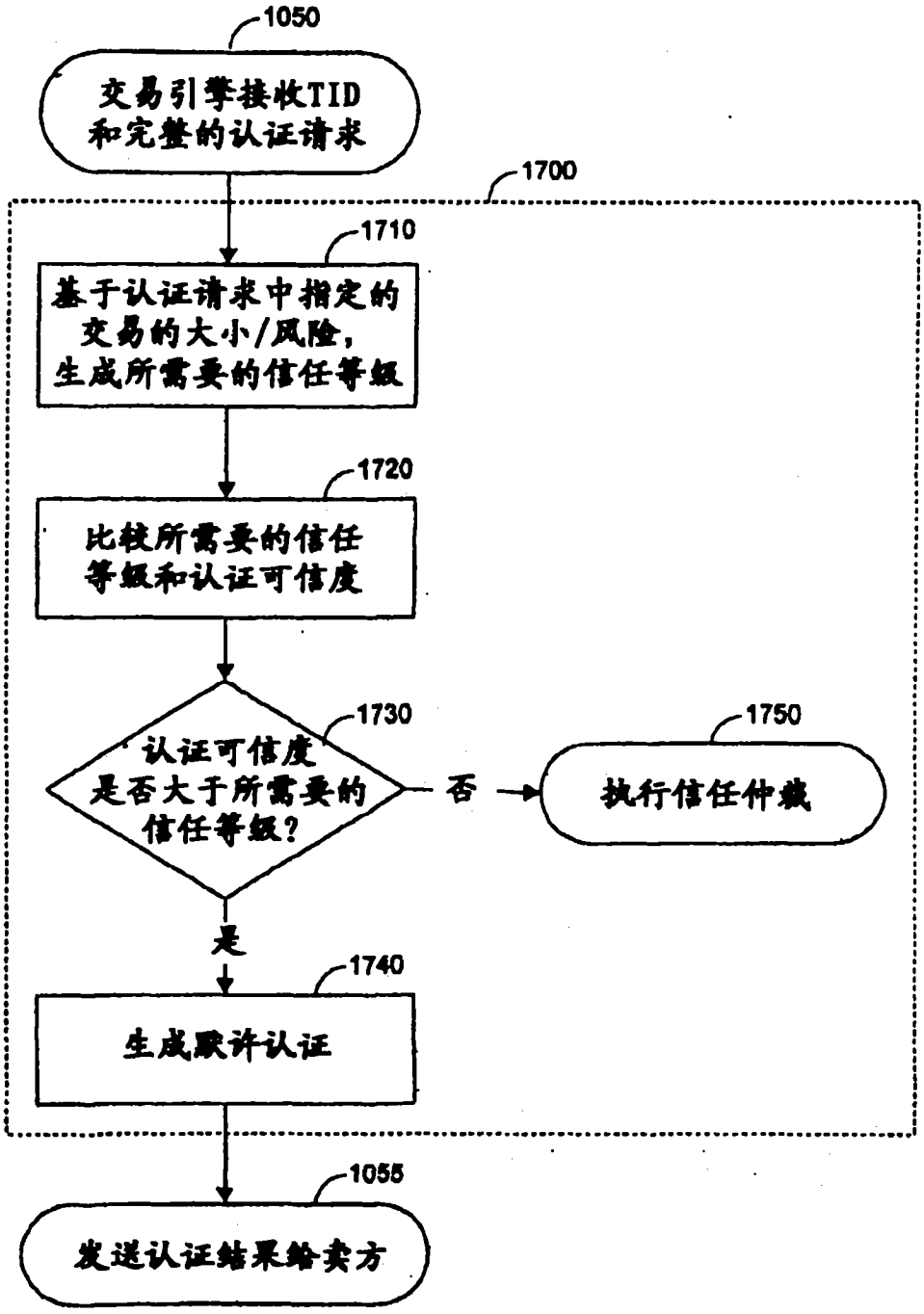

1050
交易引擎接收TID
和完整的认证请求
1700
1710
基于认证请求中指定的
交易的大小/风险，
生成所需要的信任等级
1720
比较所需要的信任
等级和认证可信度
1730
认证可信度
是否大于所需要的
信任等级?
否
1750
执行信任仲裁
是
1740
生成默许认证
1055
发送认证结果给卖方

图 17

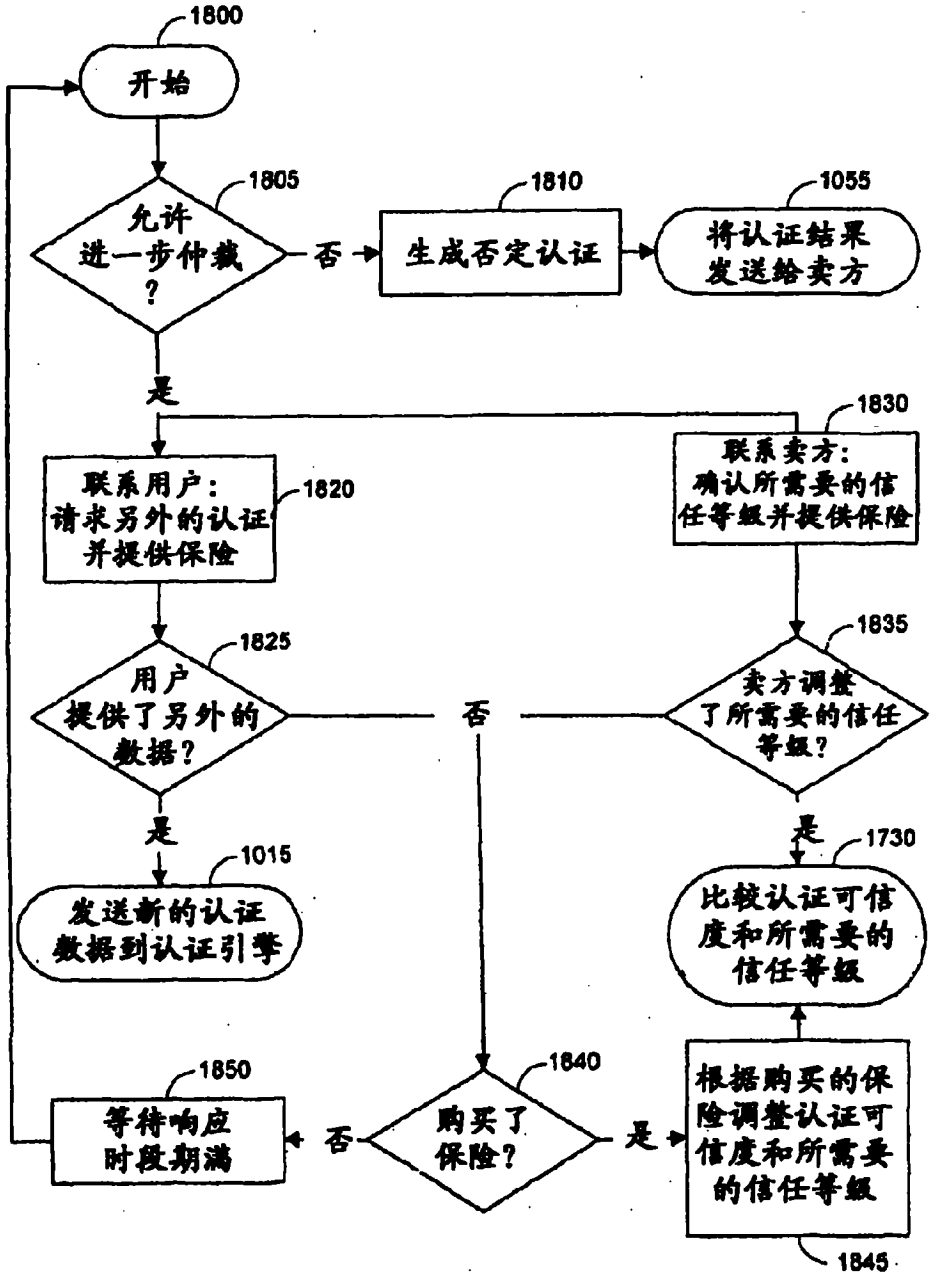

1800
开始
1805
允许进一步仲裁？
否
1810
生成否定认证
1055
将认证结果发送给卖方
是
1830
联系卖方：确认所需要的信任等级并提供保险
联系用户：请求另外的认证并提供保险
1820
1835
卖方调整了所需要的信任等级？
否
1825
用户提供了另外的数据？
是
1730
比较认证可信度和所需要的信任等级
是
1015
发送新的认证数据到认证引擎
1850
等待响应时段期满
否
1840
购买了保险？
是
根据购买的保险调整认证可信度和所需要的信任等级
1845

图 18

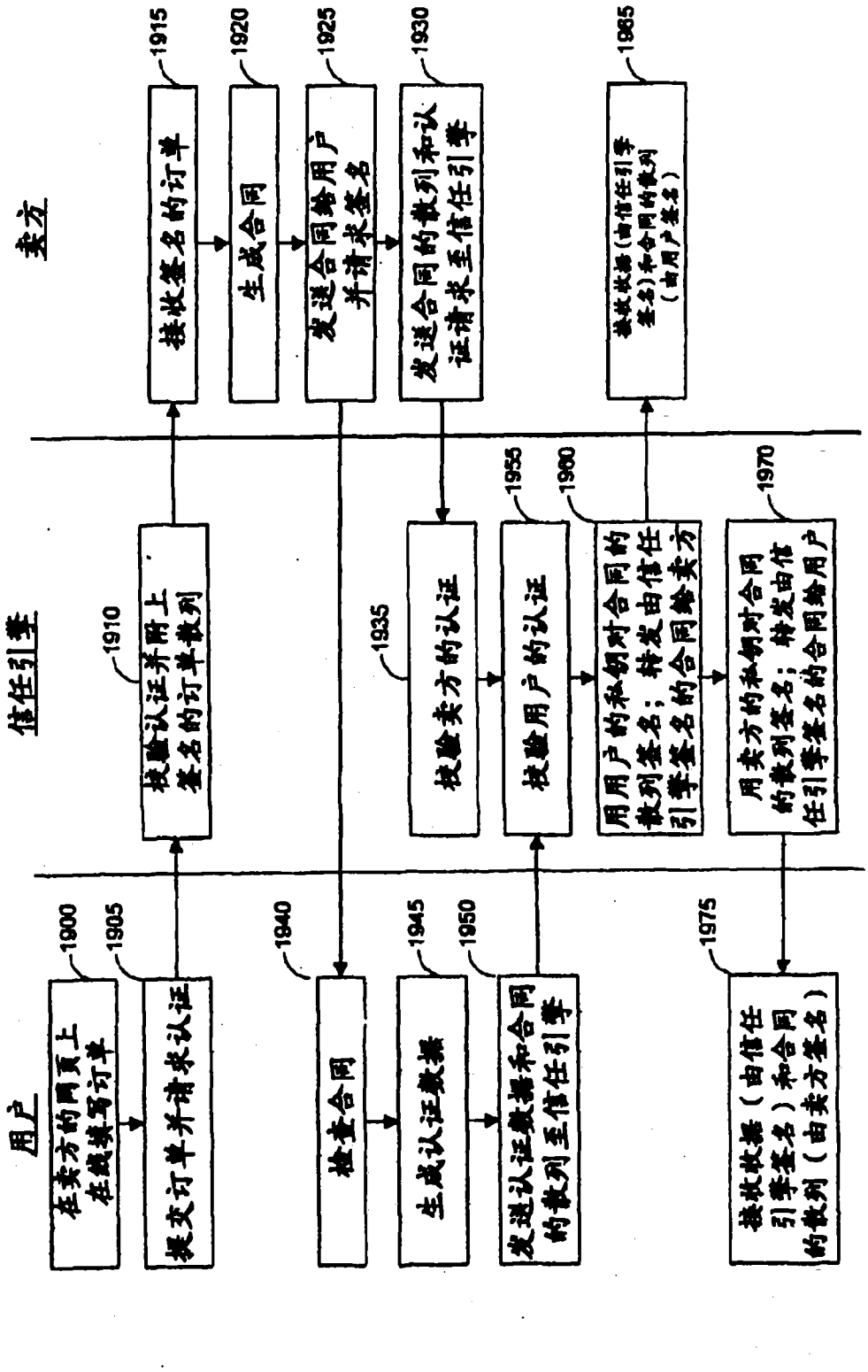

卖方
信任引擎
用户
1915
接收签名的订单
1920
生成合同
1925
发送合同给用户并请求签名
1930
发送合同的散列和认证请求至信任引擎
1985
接收收据（由信任引擎签名）和合同的散列（由用户签名）
1910
校验认证并附上签名的订单散列
1935
校验卖方的认证
1955
校验用户的认证
1960
用用户的私钥对合同的散列签名；将发由信任引擎签名的合同给卖方
1970
用卖方的私钥对合同的散列签名；将发由信任引擎签名的合同给用户
1900
在卖方的网页上在线填写订单
1905
提交订单并请求认证
1940
检查合同
1945
生成认证数据
1950
发送认证数据和合同的散列至信任引擎
1975
接收收据（由信任引擎签名）和合同的散列（由卖方签名）

图 19

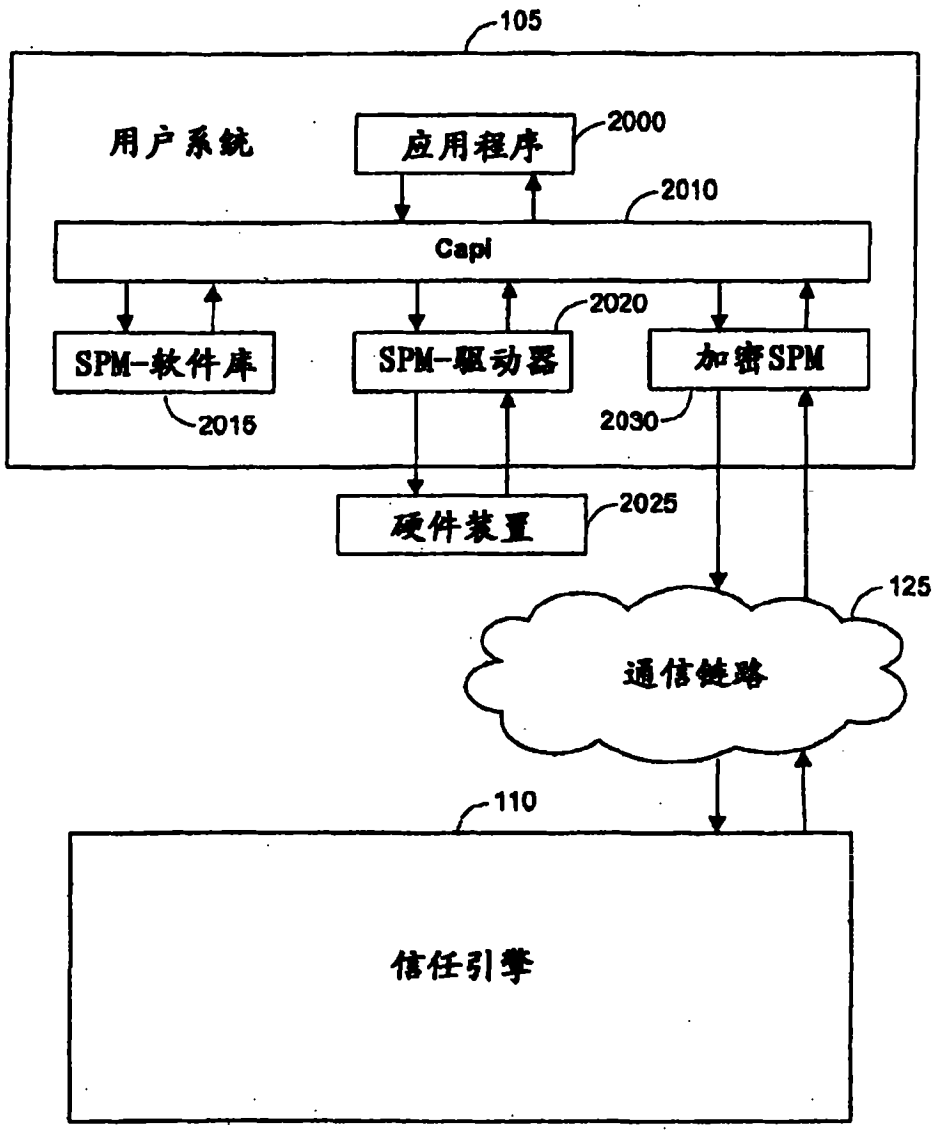

105
用户系统
应用程序
2000
2010
Capi
2020
SPM-软件库
SPM-驱动器
加密SPM
2015
2030
硬件装置
2025
125
通信链路
110
信任引擎

图 20

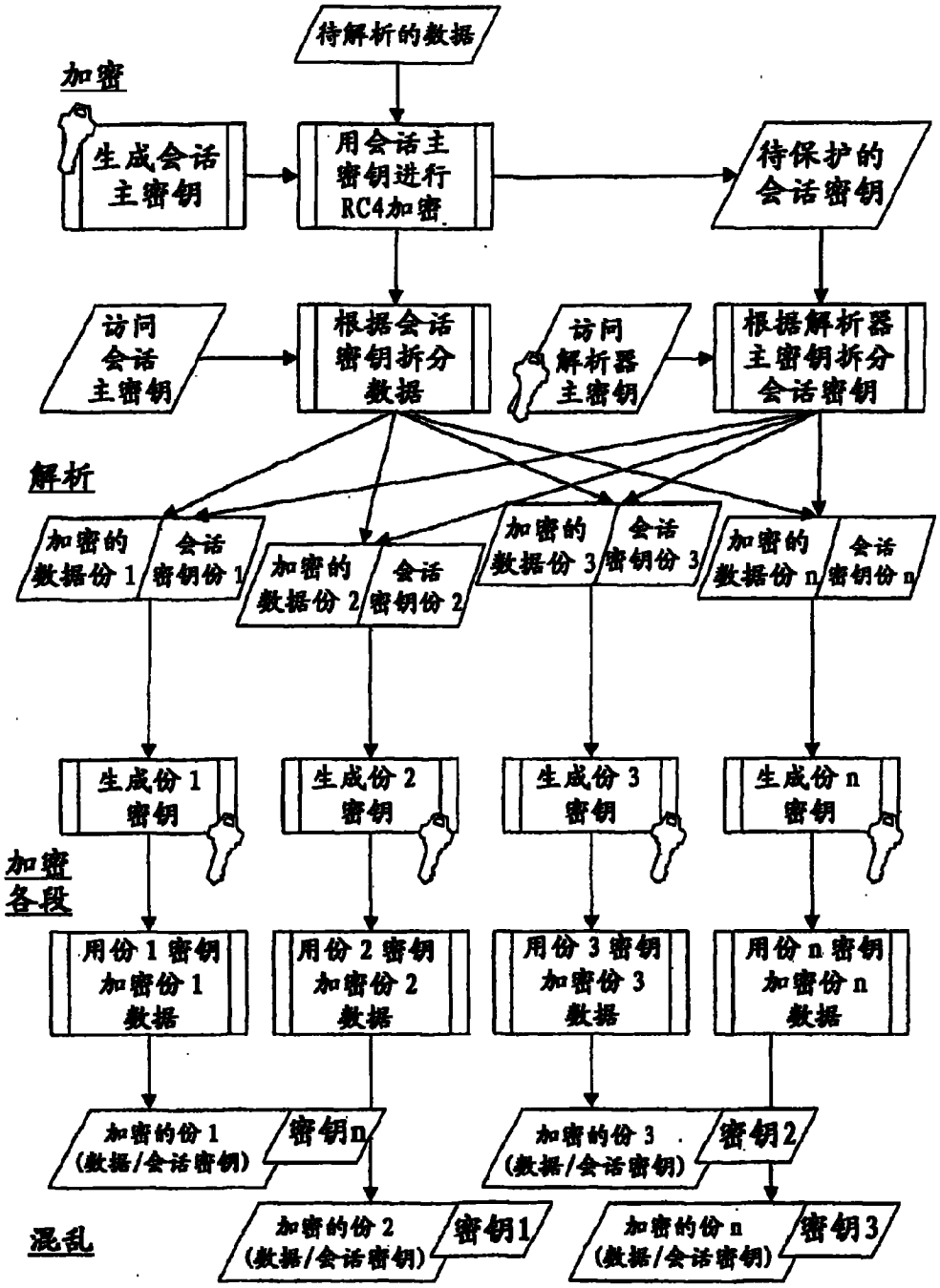

待解析的数据
加密
生成会话主密钥
用会话主密钥进行RC4加密
待保护的会话密钥
访问会话主密钥
根据会话密钥拆分数据
访问解析器主密钥
根据解析器主密钥拆分会话密钥
解析
加密的数据份 1
会话密钥份 1
加密的数据份 2
会话密钥份 2
加密的数据份 3
会话密钥份 3
加密的数据份 n
会话密钥份 n
生成份 1 密钥
生成份 2 密钥
生成份 3 密钥
生成份 n 密钥
加密各段
用份 1 密钥加密份 1 数据
用份 2 密钥加密份 2 数据
用份 3 密钥加密份 3 数据
用份 n 密钥加密份 n 数据
加密的份 1 (数据/会话密钥)
密钥n
加密的份 3 (数据/会话密钥)
密钥2
加密的份 2 (数据/会话密钥)
密钥1
加密的份 n (数据/会话密钥)
密钥3
混乱

图 21

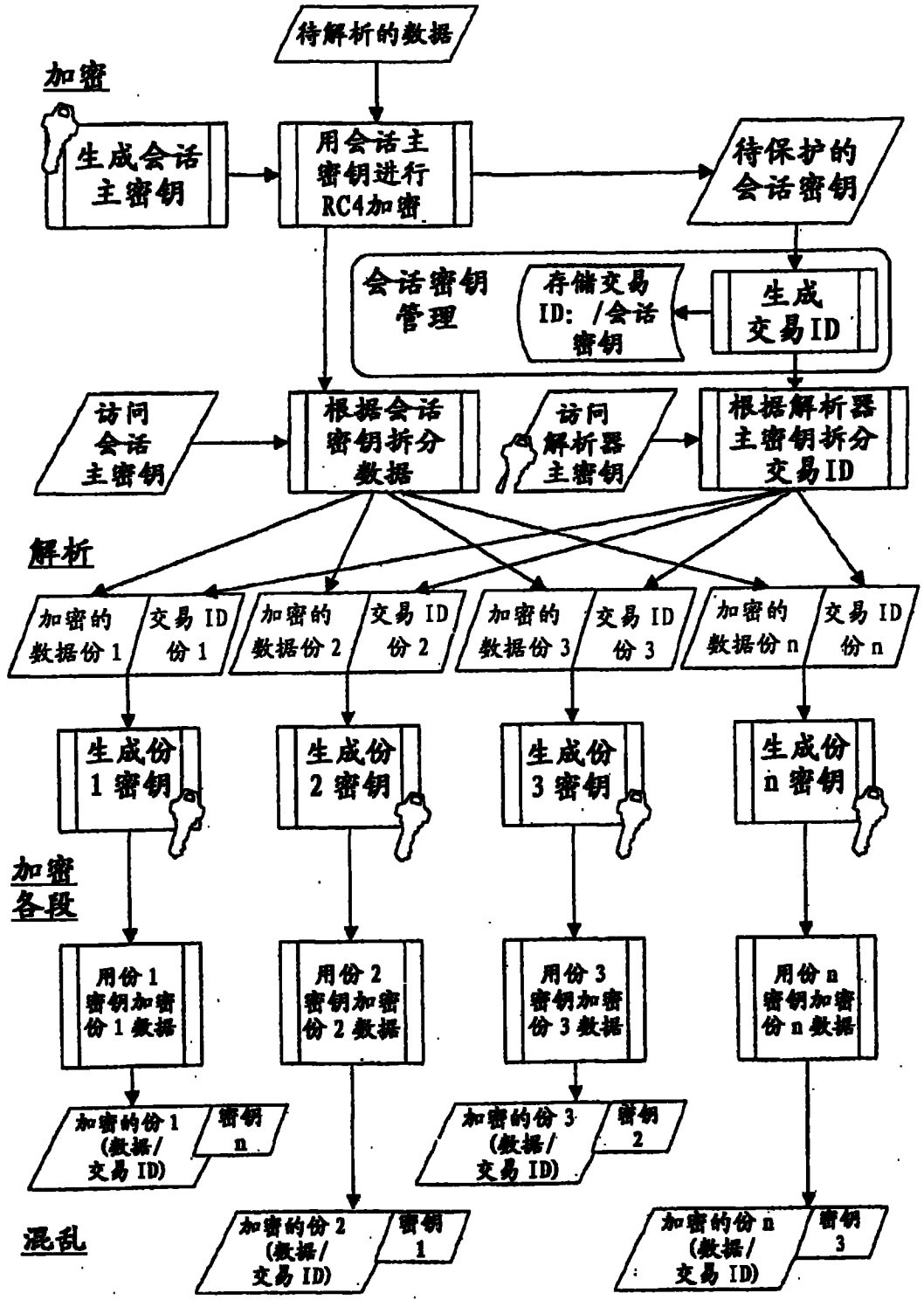

待解析的数据
加密
生成会话主密钥
用会话主密钥进行RC4加密
待保护的会话密钥
会话密钥管理
存储交易ID: /会话密钥
生成交易ID
访问会话主密钥
根据会话密钥拆分数据
访问解析器主密钥
根据解析器主密钥拆分交易ID
解析
加密的数据份1
交易ID份1
加密的数据份2
交易ID份2
加密的数据份3
交易ID份3
加密的数据份n
交易ID份n
生成份1密钥
生成份2密钥
生成份3密钥
生成份n密钥
加密各段
用份1密钥加密份1数据
用份2密钥加密份2数据
用份3密钥加密份3数据
用份n密钥加密份n数据
加密的份1(数据/交易ID)
密钥n
加密的份3(数据/交易ID)
密钥2
混乱
加密的份2(数据/交易ID)
密钥1
加密的份n(数据/交易ID)
密钥3

图 22

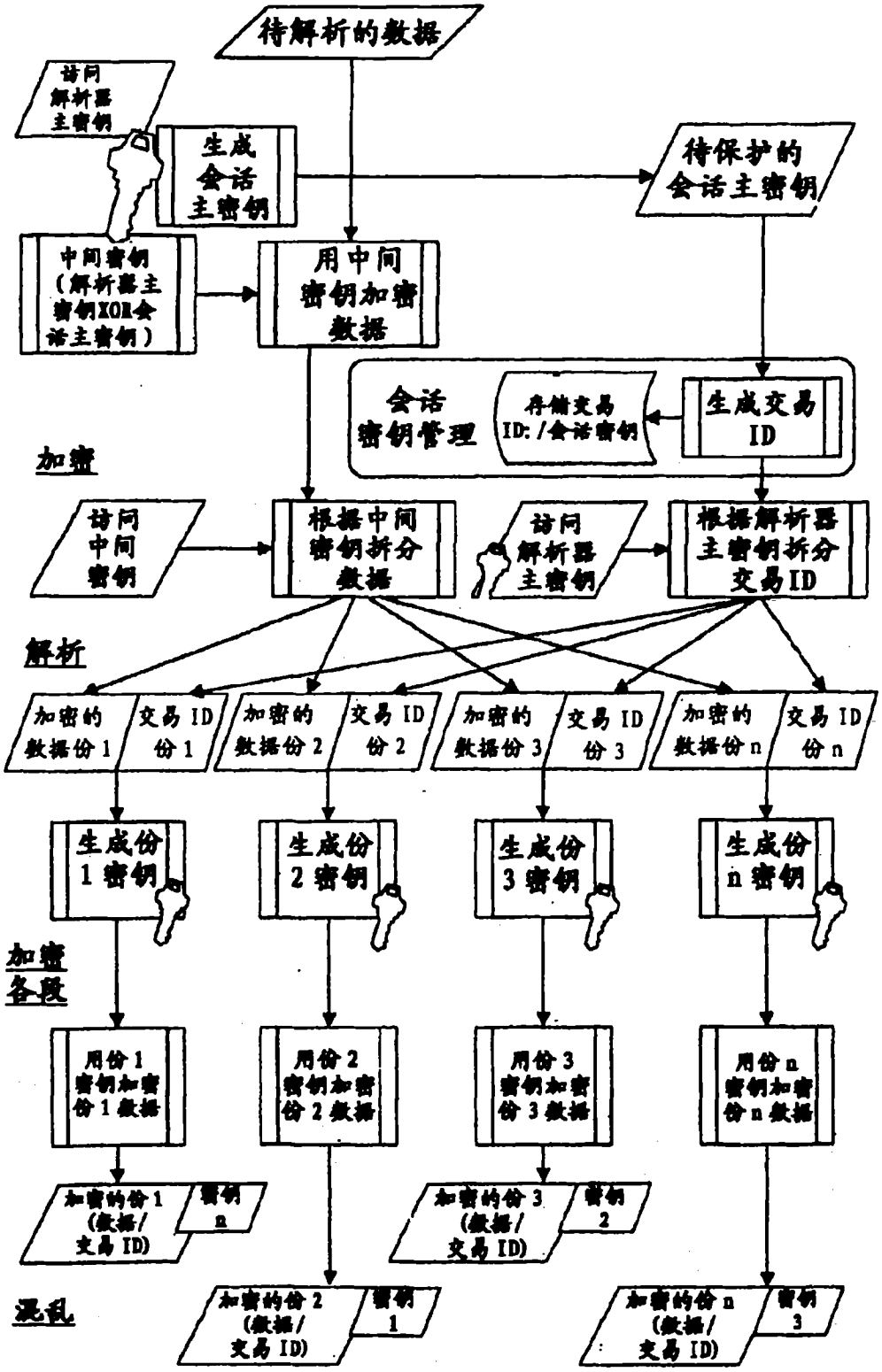

待解析的数据
访问解析器主密钥
生成会话主密钥
待保护的会话主密钥
中间密钥（解析器主密钥XOR会话主密钥）
用中间密钥加密数据
会话密钥管理
存储交易ID: /会话密钥
生成交易ID
加密
访问中间密钥
根据中间密钥拆分数据
访问解析器主密钥
根据解析器主密钥拆分交易ID
解析
加密的数据份1
交易ID份1
加密的数据份2
交易ID份2
加密的数据份3
交易ID份3
加密的数据份n
交易ID份n
生成份1密钥
生成份2密钥
生成份3密钥
生成份n密钥
加密各段
用份1密钥加密份1数据
用份2密钥加密份2数据
用份3密钥加密份3数据
用份n密钥加密份n数据
加密的份1（数据/交易ID）
密钥n
加密的份3（数据/交易ID）
密钥2
混乱
加密的份2（数据/交易ID）
密钥1
加密的份n（数据/交易ID）
密钥3

图 23

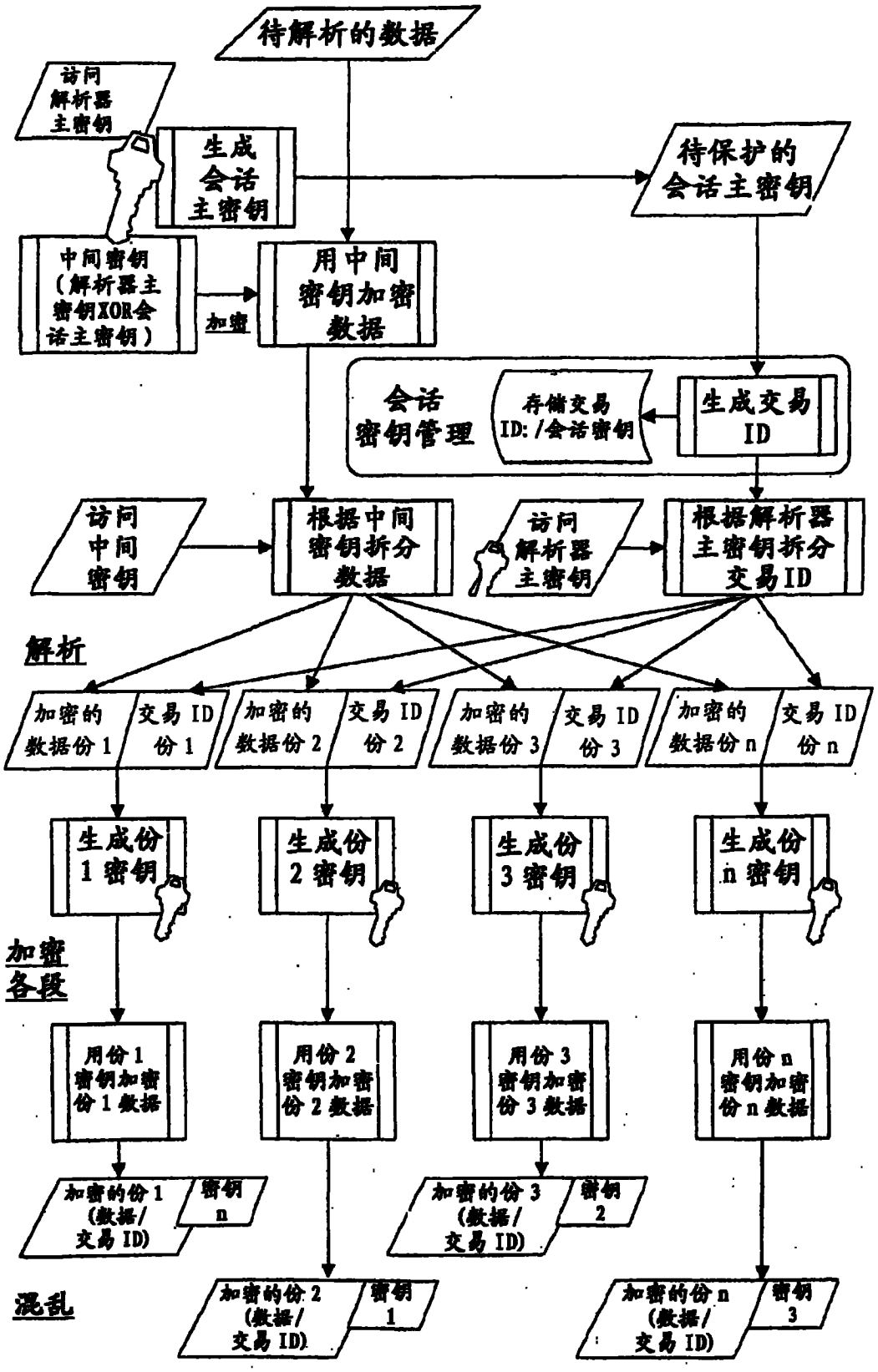

待解析的数据
访问解析器主密钥
生成会话主密钥
待保护的会话主密钥
中间密钥（解析器主密钥XOR会话主密钥）
加密
用中间密钥加密数据
会话密钥管理
存储交易ID: /会话密钥
生成交易ID
访问中间密钥
根据中间密钥拆分数据
访问解析器主密钥
根据解析器主密钥拆分交易ID
解析
加密的数据份1
交易ID份1
加密的数据份2
交易ID份2
加密的数据份3
交易ID份3
加密的数据份n
交易ID份n
生成份1密钥
生成份2密钥
生成份3密钥
生成份n密钥
加密各段
用份1密钥加密份1数据
用份2密钥加密份2数据
用份3密钥加密份3数据
用份n密钥加密份n数据
加密的份1（数据/交易ID）
密钥n
加密的份3（数据/交易ID）
密钥2
混乱
加密的份2（数据/交易ID）
密钥1
加密的份n（数据/交易ID）
密钥3

图 24

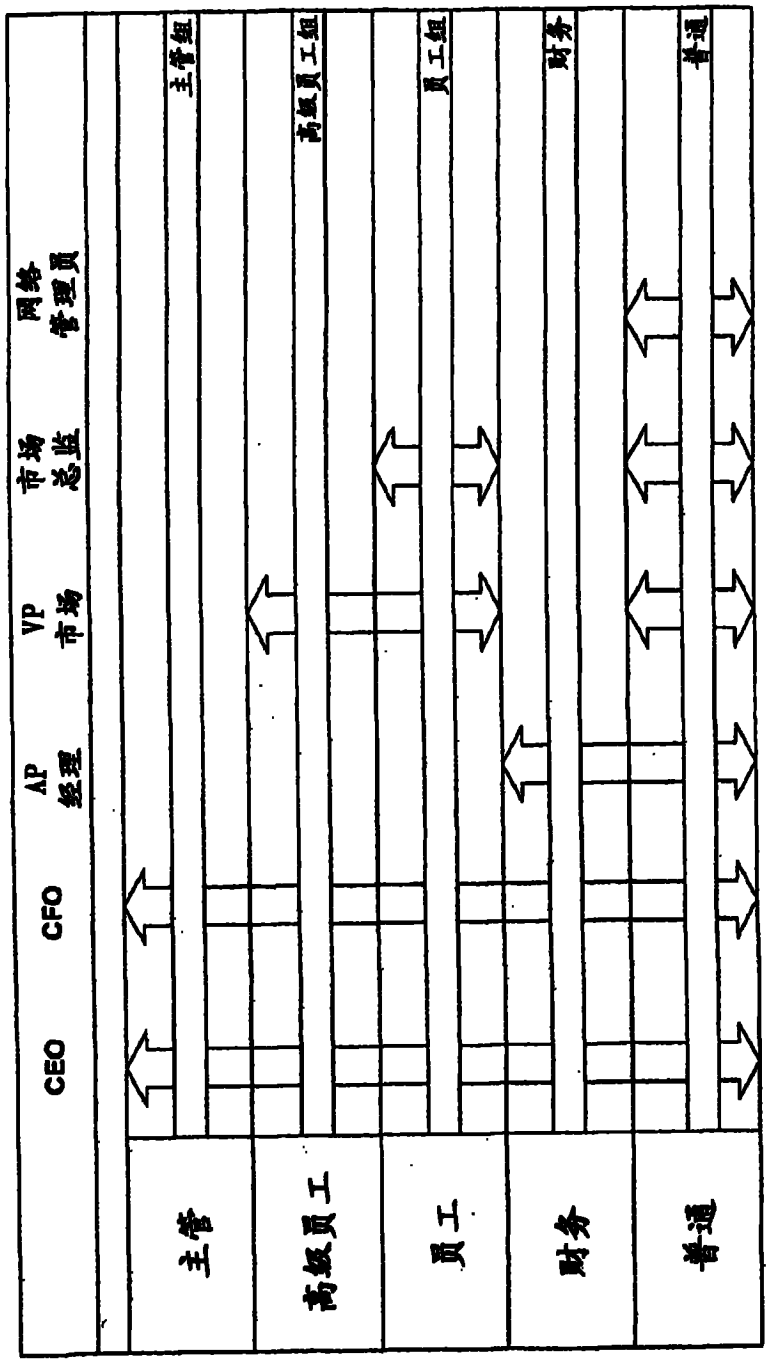

CEO
CFO
AP 经理
VP 市场
市场 总监
网络 管理员
主管组
高级员工组
员工组
财务
普通
主管
高级员工
员工
财务
普通

图 25

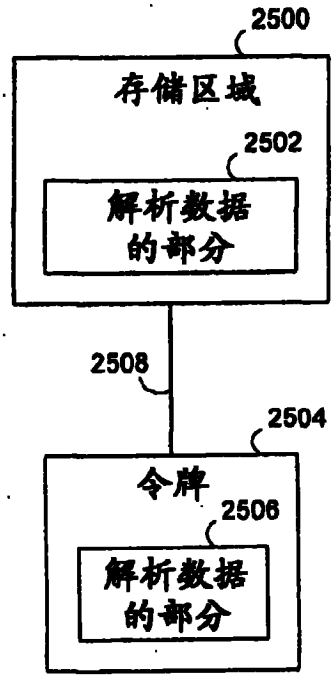

2500
存储区域
2502
解析数据的部分
2508
2504
令牌
2506
解析数据的部分

图 26

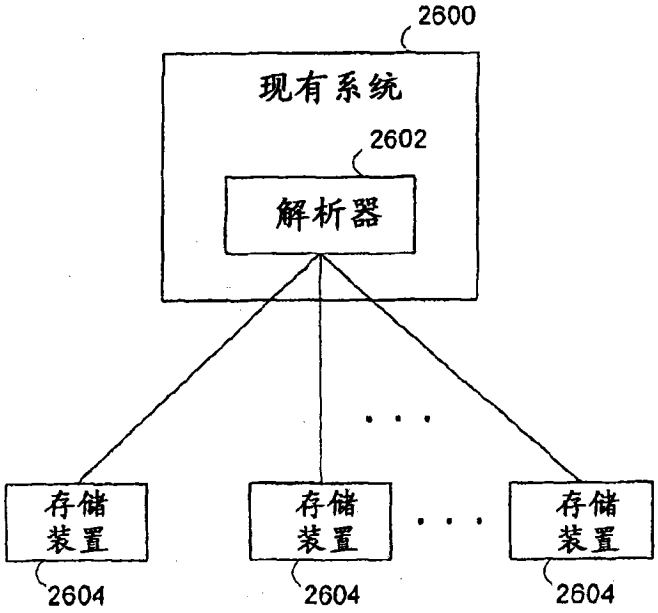

2600
现有系统
2602
解析器
存储装置
2604
存储装置
2604
存储装置
2604

图 27

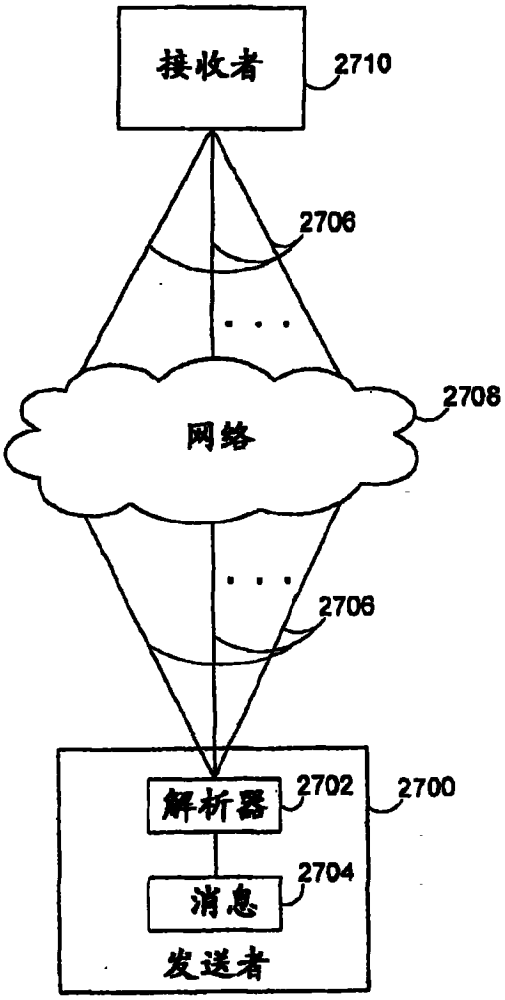

接收者
2710
2706
网络
2708
2706
解析器
2702
2700
消息
2704
发送者

图 28

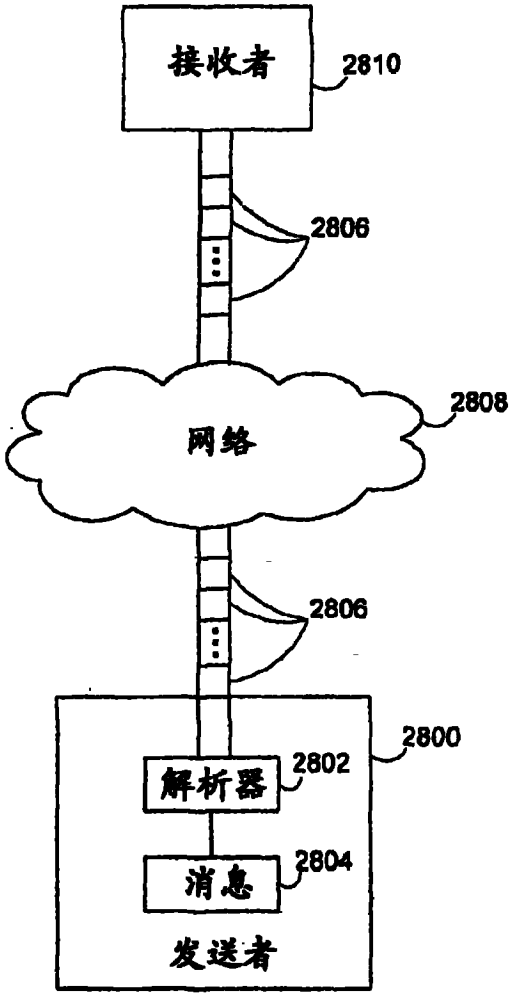

接收者
2810
2806
网络
2808
2806
2800
解析器
2802
消息
2804
发送者

图 29

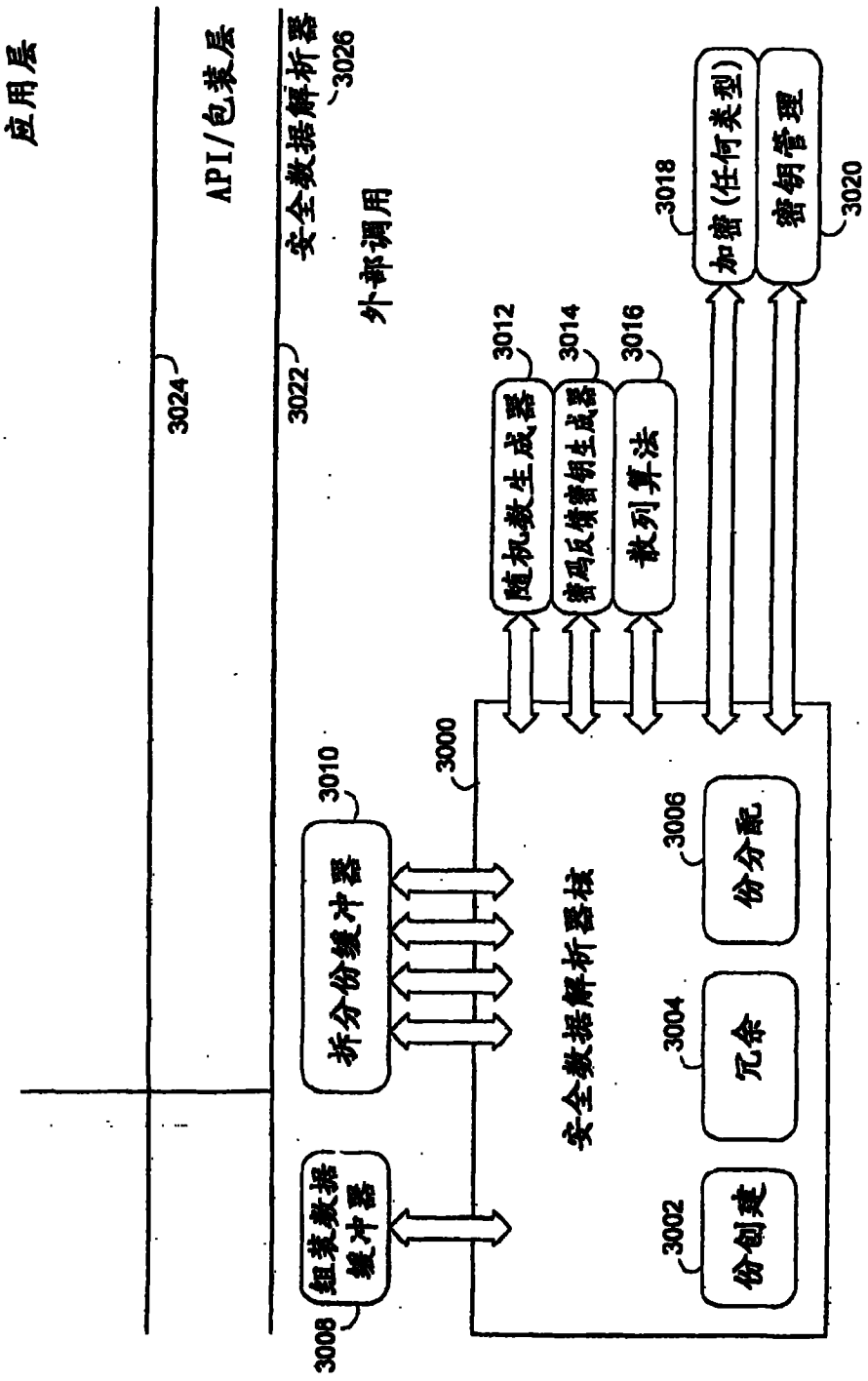

应用层
API/包装层
安全数据解析器
3026
外部调用
3024
3022
3012
3014
3016
3018
3020
随机数生成器
密码反馈密钥生成器
数列算法
加密(任何类型)
密钥管理
3000
3010
拆分份缓冲器
3006
份分配
安全数据解析器核
3004
冗余
3002
份创建
组装数据缓冲器
3008

图 30

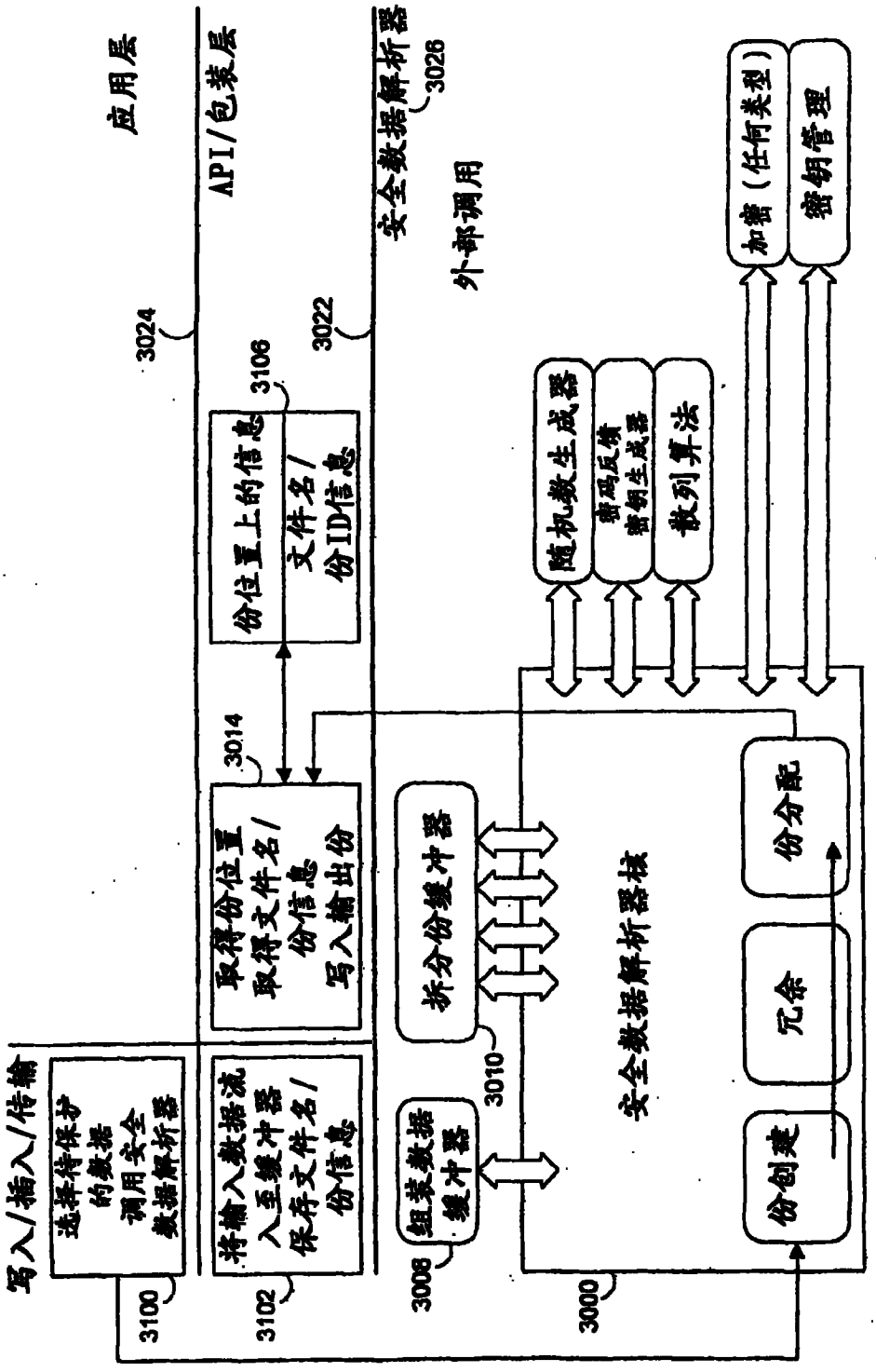

应用层
API/包装层
安全数据解析器
3028
外部调用
3024
3022
3106
份位置上的信息
文件名/份ID信息
3014
取得份位置
取得文件名/份信息
写入输出份
写入/输入/传输
选择待保护的数据调用安全数据解析器
将输入数据流入至缓冲器保存文件名/份信息
3100
3102
拆分份缓冲器
3010
组装数据缓冲器
3008
安全数据解析器核
3000
随机数生成器
密码反馈密钥生成器
散列算法
加密（任何类型）
密钥管理
份分配
冗余
份创建

图 31

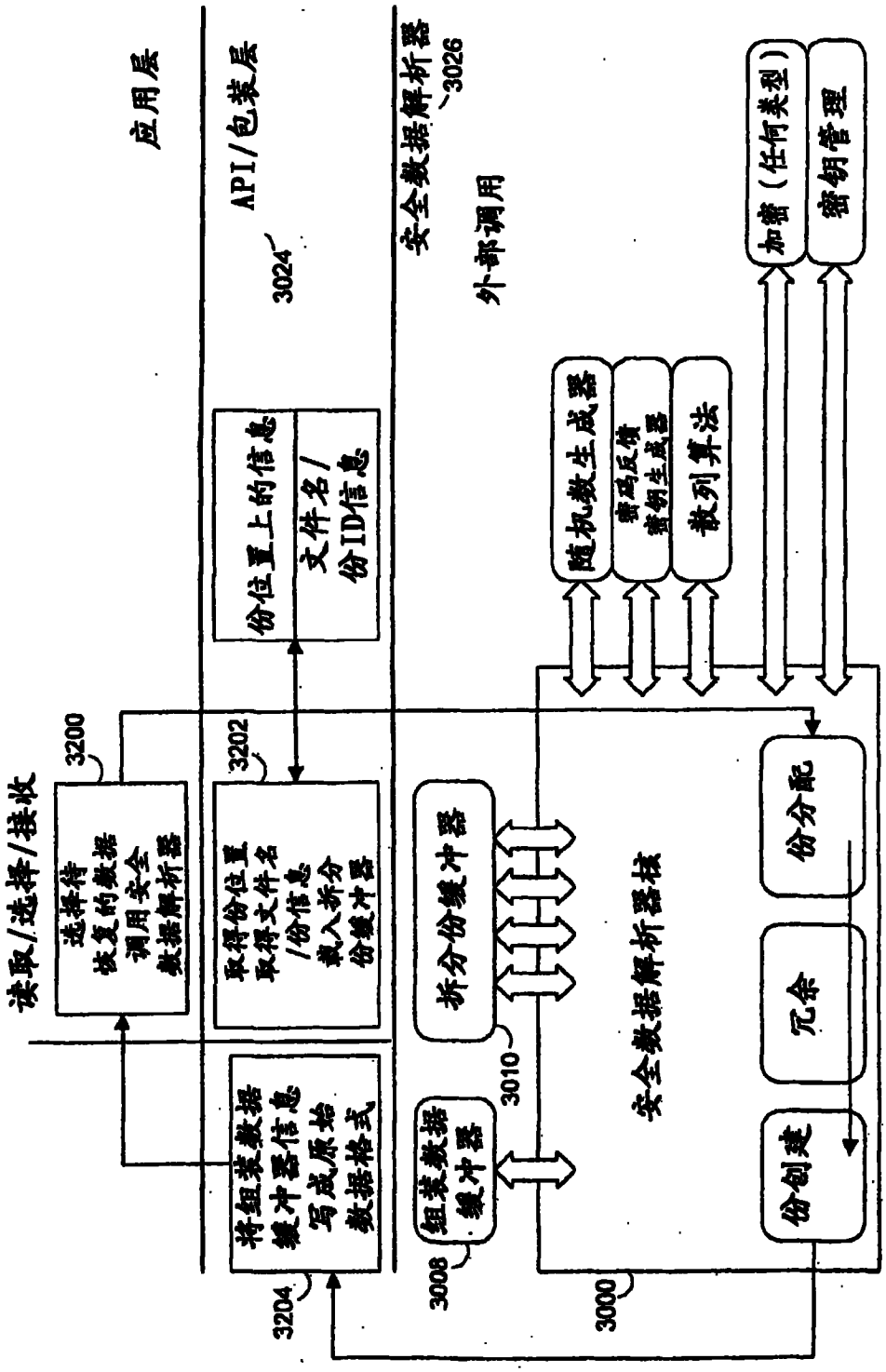

应用层
API/包装层
3024
安全数据解析器
3026
外部调用
读取/选择/接收
3200
选择待恢复的数据调用安全数据解析器
3202
取得份位置取得文件名/份信息载入拆分份缓冲器
份位置上的信息
文件名/份ID信息
将组装数据缓冲器信息写成原始数据格式
3204
组装数据缓冲器
3008
拆分份缓冲器
3010
安全数据解析器核
3000
随机数生成器
密码反馈密钥生成器
散列算法
加密（任何类型）
密钥管理
份分配
冗余
份创建

图 32

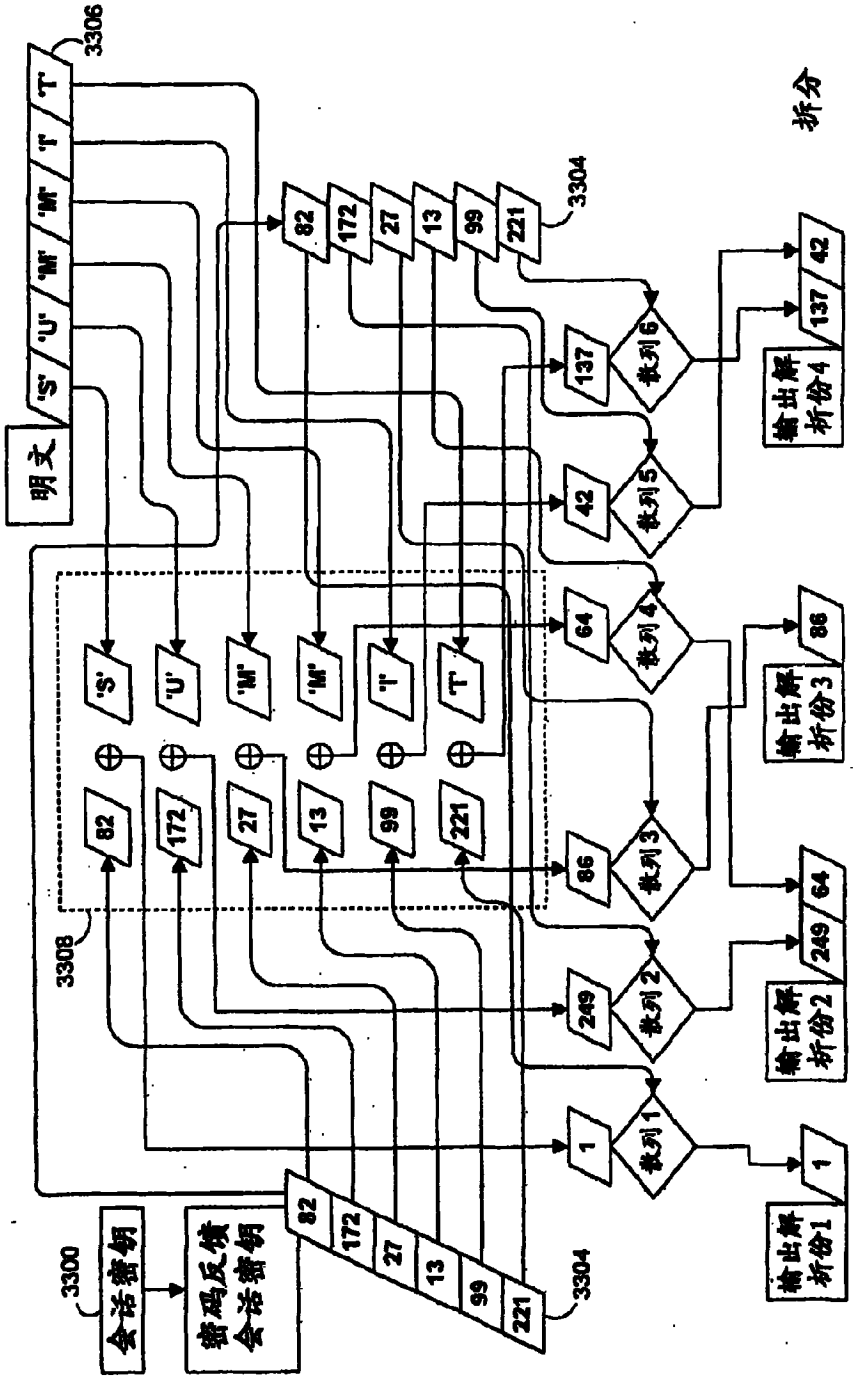

明文
'S'
'U'
'W'
'M'
'T'
'T'
3306
82
172
27
13
99
221
3304
会话密钥
密码反馈会话密钥
3300
3308
1
249
86
64
42
137
拆分
输出解折份1
输出解折份2
输出解折份3
输出解折份4

图 33

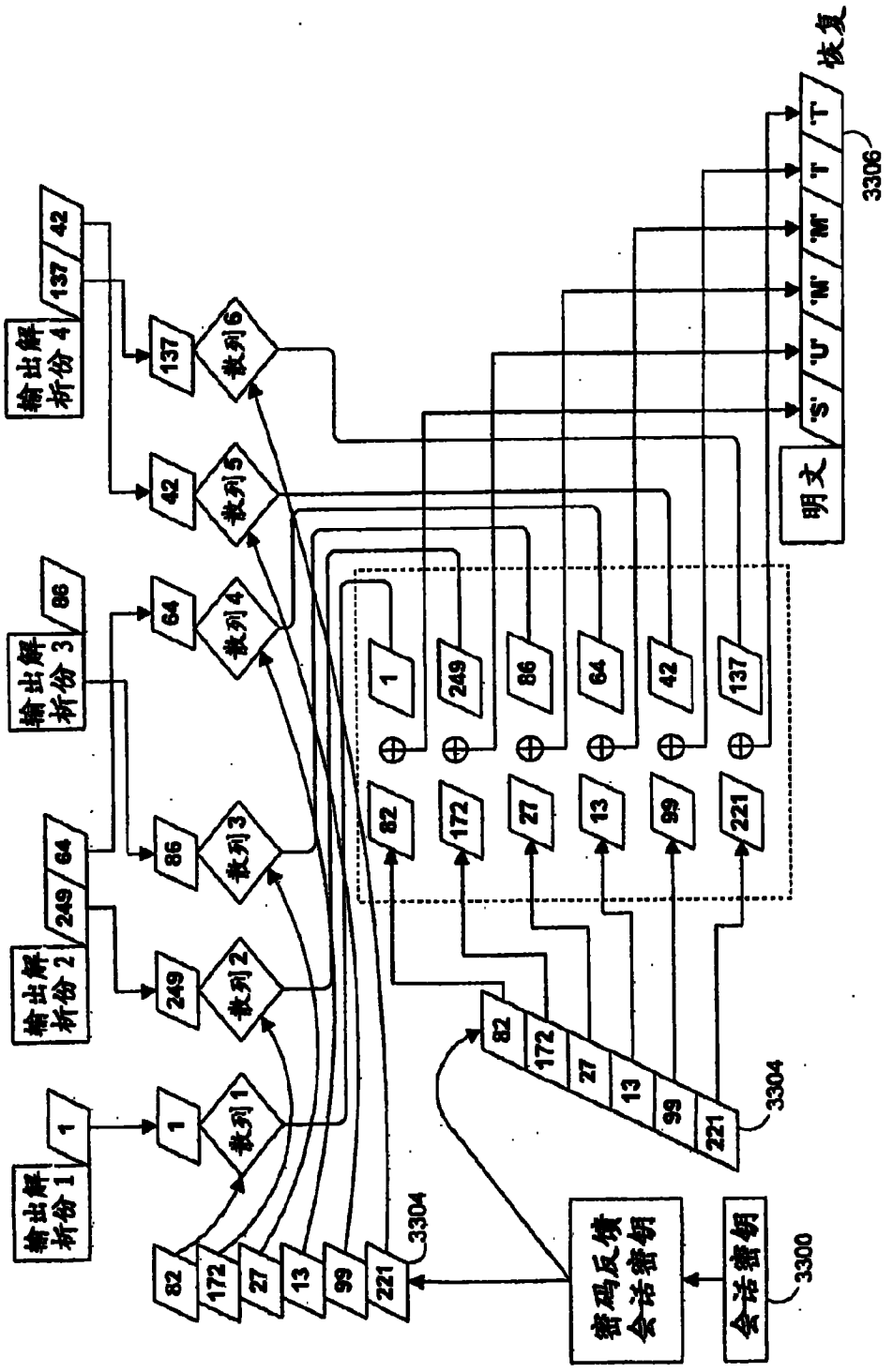

输出解析份 1
输出解析份 2
输出解析份 3
输出解析份 4
散列1
散列2
散列3
散列4
散列5
散列6
明文
恢复
'S'
'U'
'N'
'W'
'T'
'T'
3306
3304
3300
密码反馈会话密钥
会话密钥

图 34

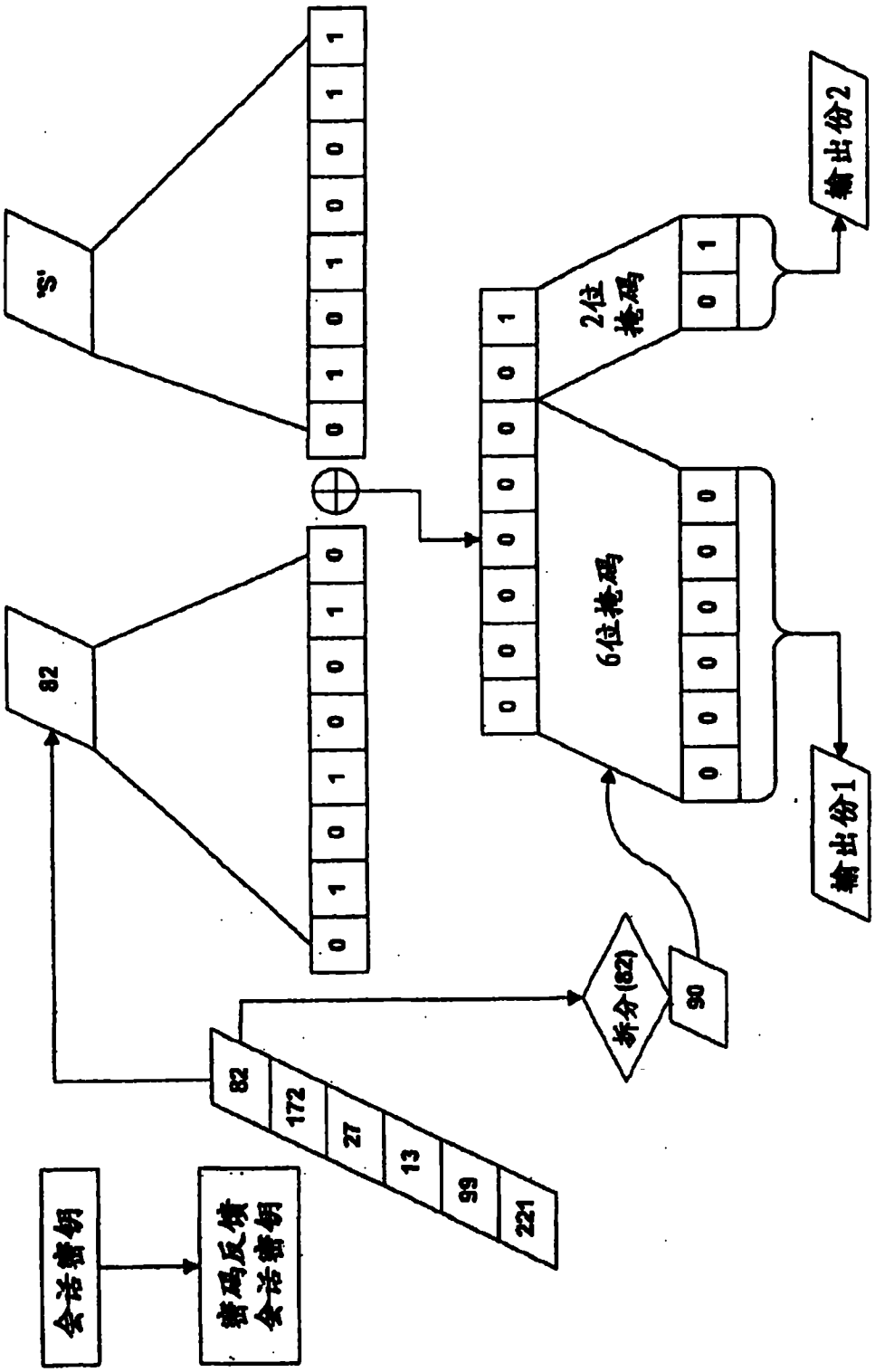

'S'
82
2位掩码
6位掩码
输出份2
输出份1
拆分(82)
90
82
172
27
13
99
221
会话密钥
密码反馈会话密钥

图 35

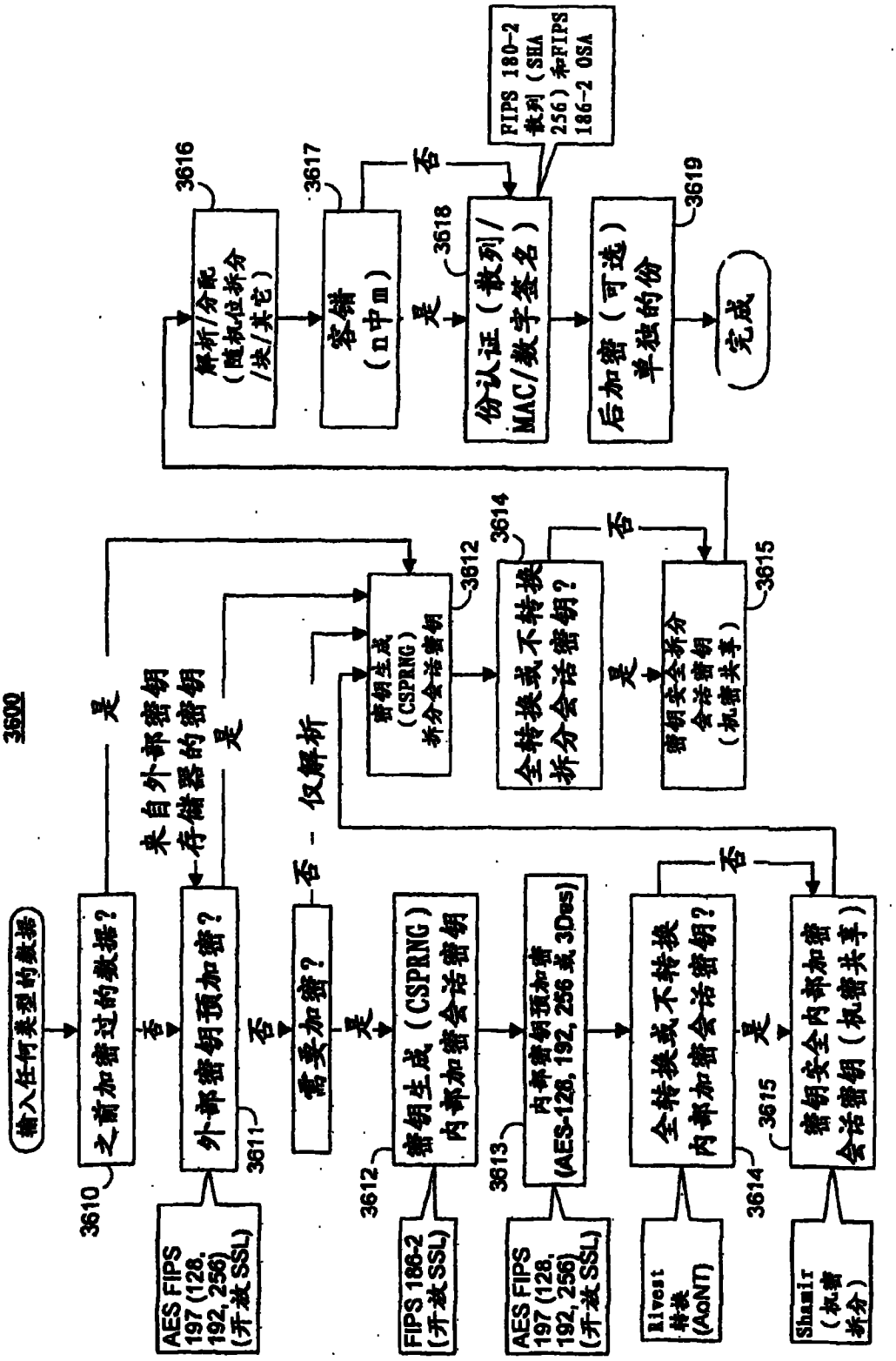

3600
输入任何类型的数据
之前加密过的数据？
否
是
3610
AES FIPS 197 (128, 192, 256) (开放SSL)
外部密钥预加密？
否
是
来自外部密钥存储器的密钥
3611
需要加密？
否 - 仅解析
是
密钥生成（CSPRNG）内部加密会话密钥
3612
FIPS 186-2 (开放SSL)
内部密钥预加密 (AES-128, 192, 256或3Des)
3613
AES FIPS 197 (128, 192, 256) (开放SSL)
全转换或不转换内部加密会话密钥？
否
是
3614
Rivest 转换 (AoNT)
密钥安全内部加密会话密钥（机密共享）
3615
Shamir (机密拆分)
密钥生成（CSPRNG）拆分会话密钥
3612
全转换或不转换拆分会话密钥？
否
是
3614
密钥安全拆分会话密钥（机密共享）
3615
FIPS 180-2 散列（SHA 256）和FIPS 186-2 OSA
否
份认证（散列/MAC/数字签名）
是
3618
后加密（可选）单独的份
3619
完成
解析/分配（随机位拆分/块/其它）
3616
容错（n中m）
3617

图 36

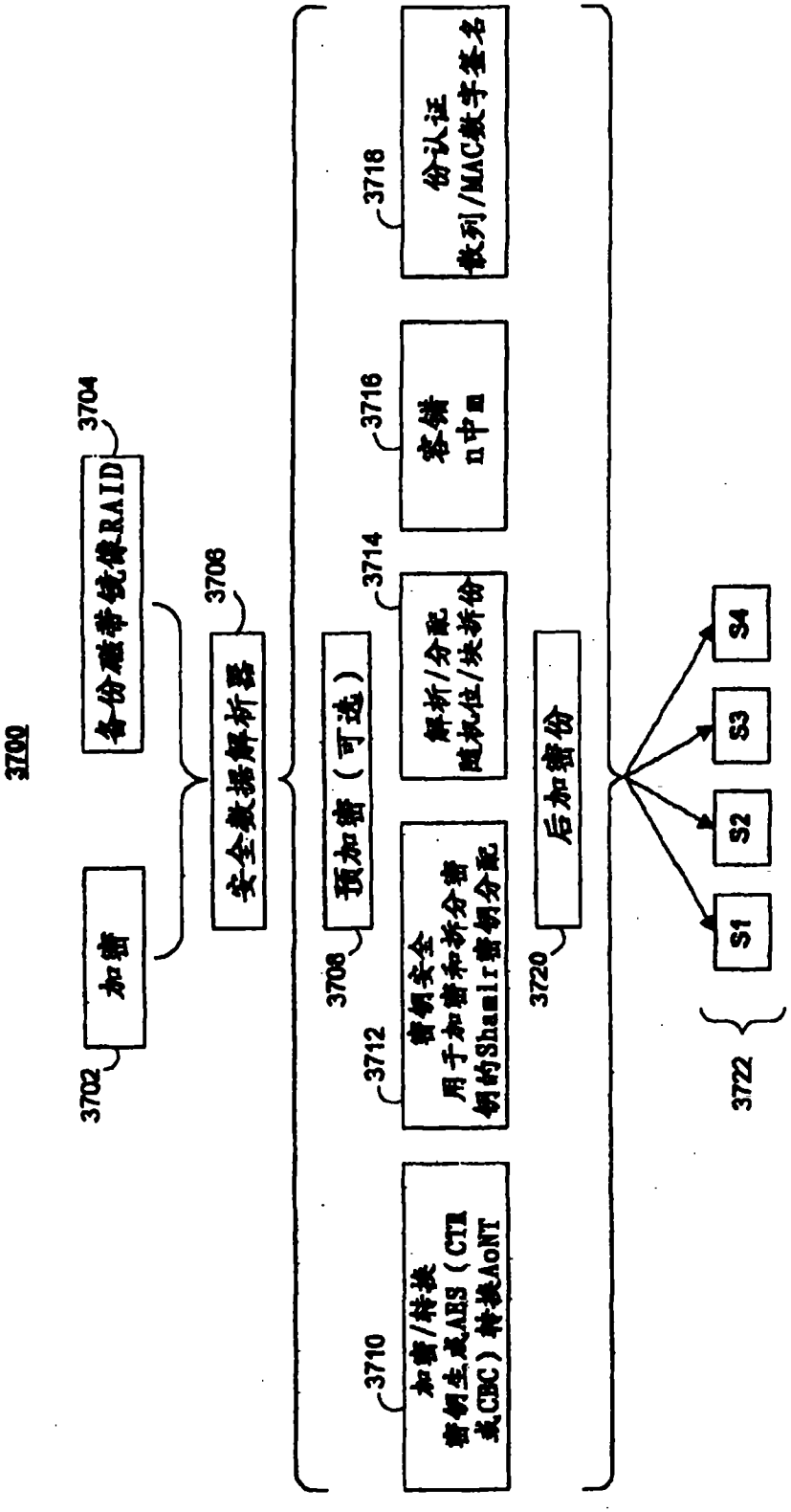

3700
3702
加密
3704
备份磁带镜像RAID
3706
安全数据解析器
3708
预加密（可选）
3710
加密/转换
密钥生成ABS（CTR
或CBC）转换AoNT
3712
密钥安全
用于加密和拆分密
钥的Shamir密钥分配
3714
解析/分配
随机性/块拆分
3716
客端
n中n
3718
份认证
数列/MAC数字签名
3720
后加密份
S1
S2
S3
S4
3722

图 37

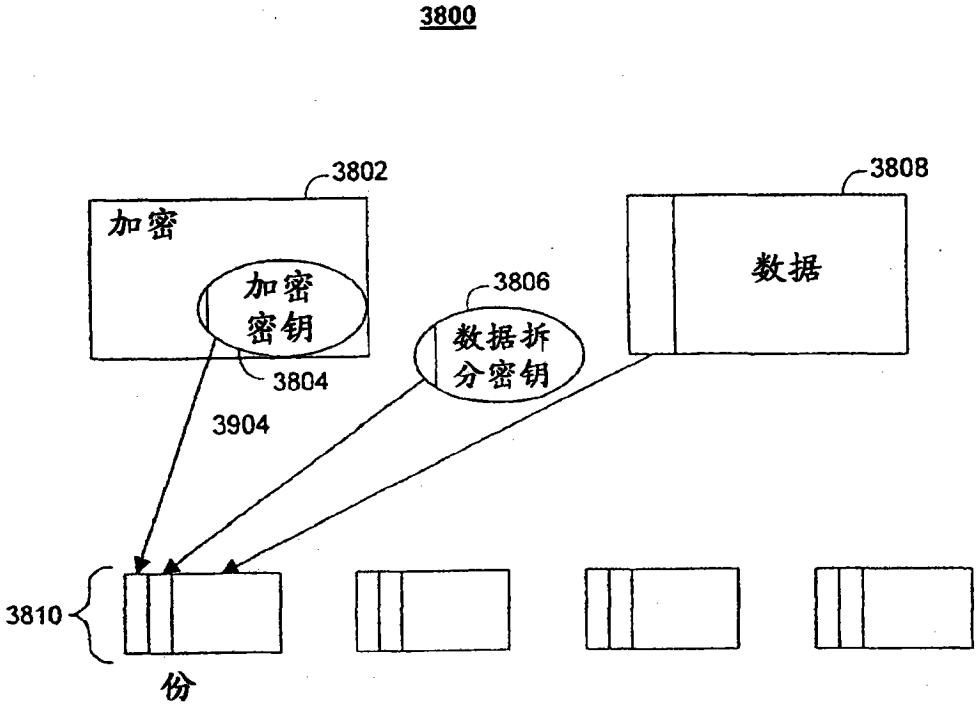



图 38

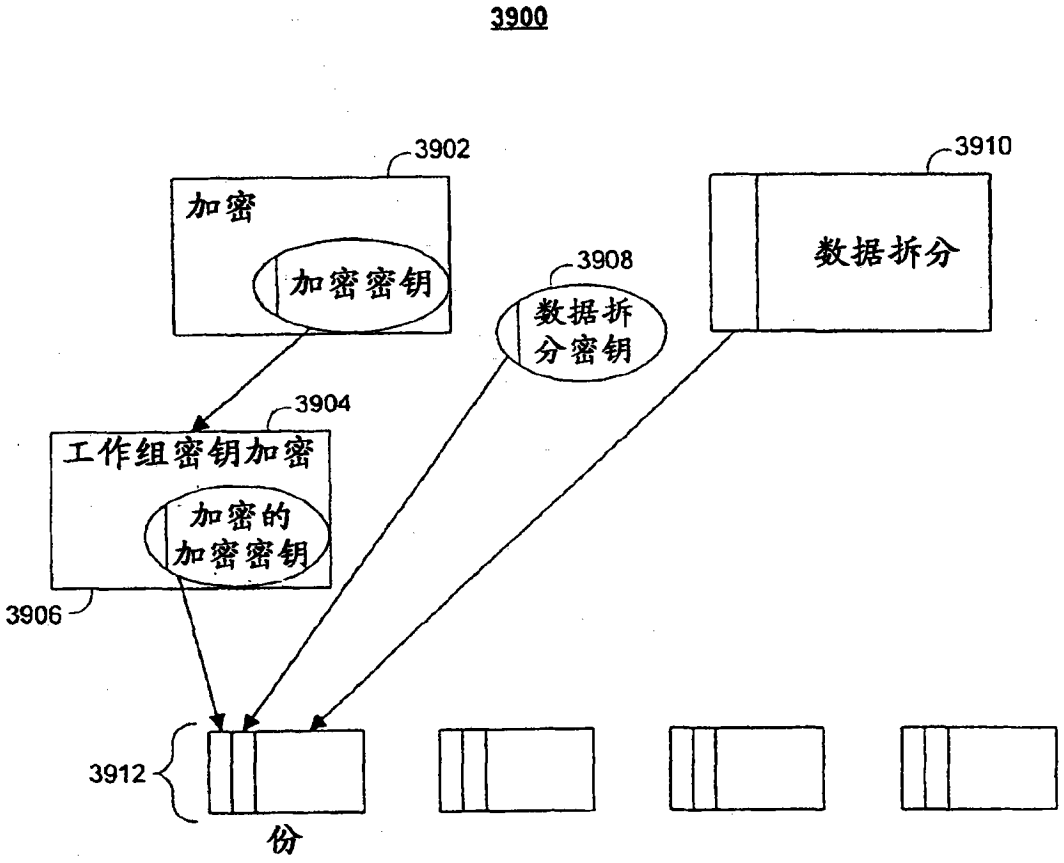

3900
3902
加密
加密密钥
3910
数据拆分
3908
数据拆分密钥
3904
工作组密钥加密
加密的加密密钥
3906
3912
份

图 39

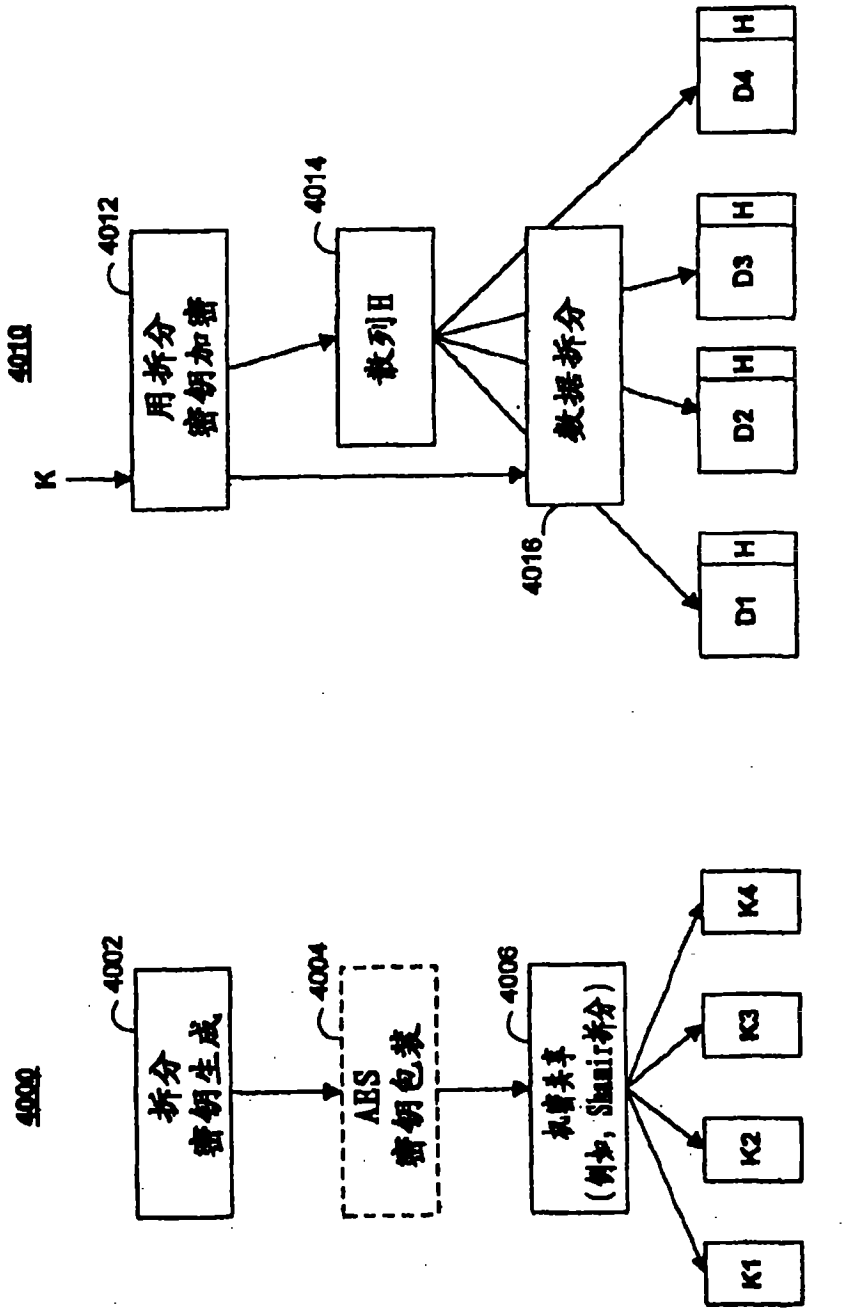

4010
K
4012
用拆分
密钥加密
4014
散列H
数据拆分
4016
H
D4
H
D3
H
D2
H
D1
4000
4002
拆分
密钥生成
4004
AES
密钥包装
4006
K4
K3
K2
K1

图 40A

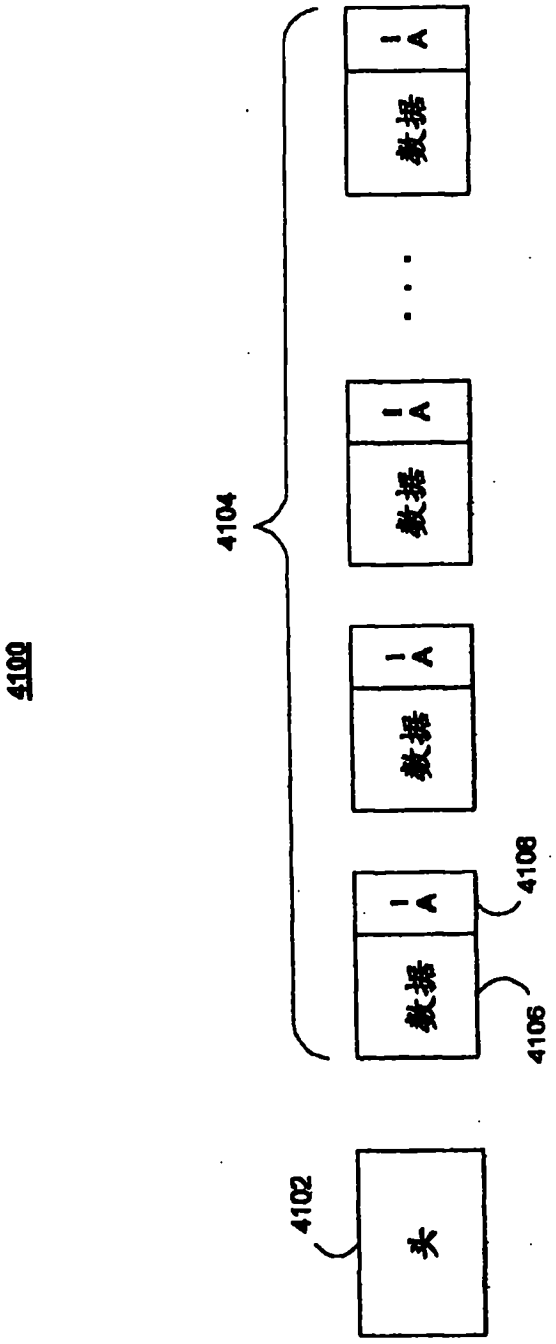



图 41

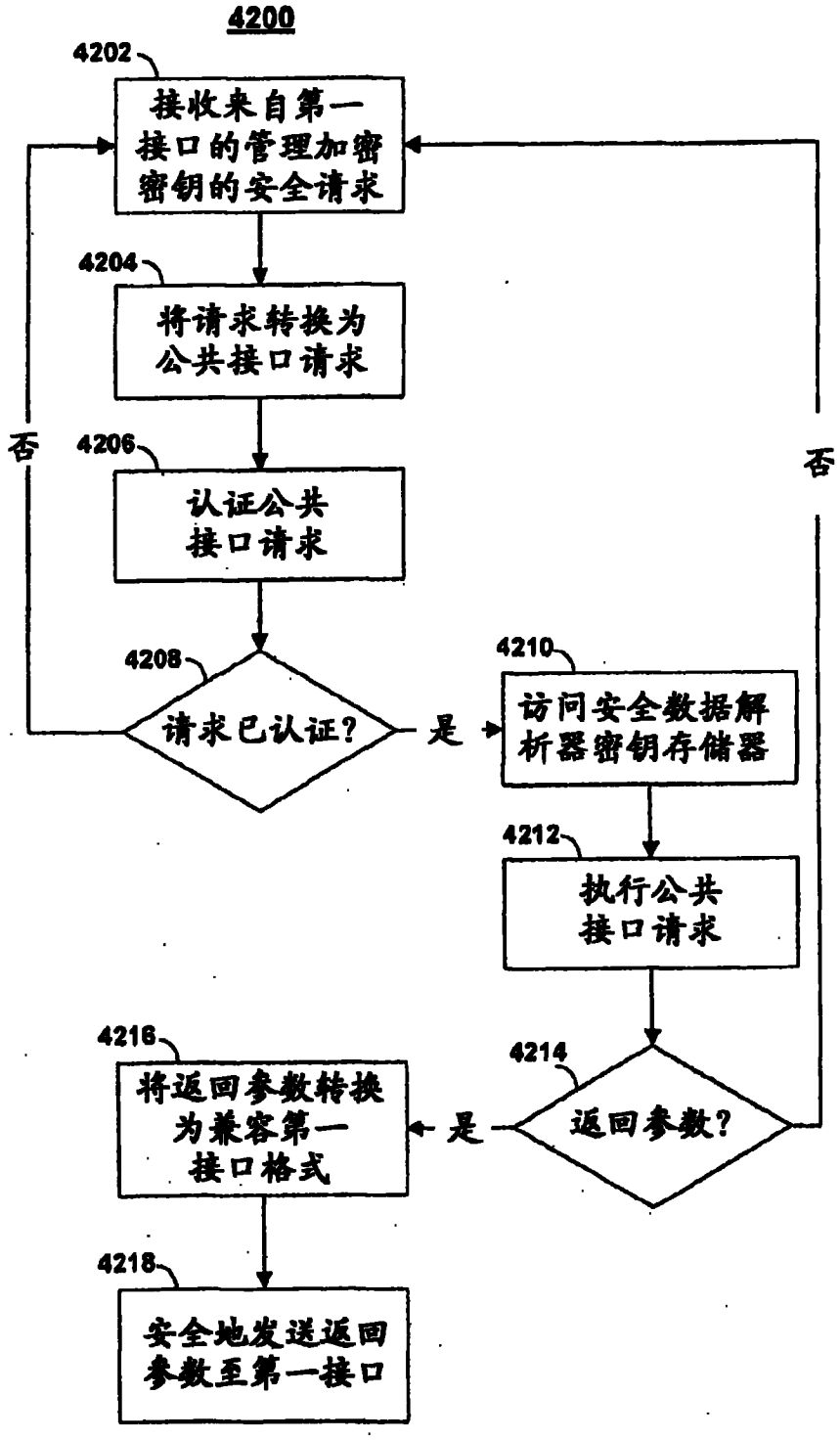

4200
4202
接收来自第一接口的管理加密密钥的安全请求
4204
将请求转换为公共接口请求
4206
认证公共接口请求
4208
请求已认证?
否
是
4210
访问安全数据解析器密钥存储器
4212
执行公共接口请求
4214
返回参数?
否
是
4216
将返回参数转换为兼容第一接口格式
4218
安全地发送返回参数至第一接口

图 42