(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 102664728 A
(43)申请公布日 2012.09.12

(21)申请号 201210121997.7

(22)申请日 2004.06.10

(30)优先权数据
10/458,928 2003.06.11 US

(62)分案原申请数据
200480022710.9 2004.06.10

(71)申请人 安全第一公司
地址 美国加利福尼亚

(72)发明人 里克·奥西尼 约翰·万扎特
马克·奥哈雷 罗格·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 9/06(2006.01)

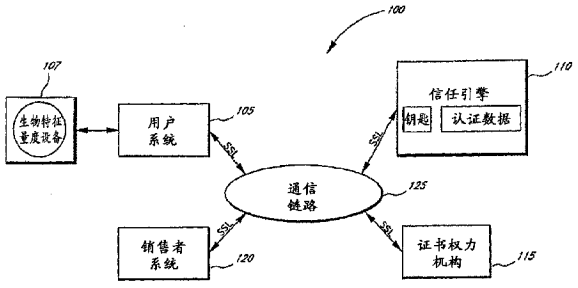
权利要求书 5 页 说明书 49 页 附图 22 页

(54)发明名称

安全数据解析器方法和系统

(57)摘要

本发明涉及安全数据解析器方法和系统。本发明提供了用于保护敏感数据免遭未经授权的访问或使用的方法和系统。本发明的方法和系统可用于多种设置中，其中包括公众一般可获得的商用设置，这种设置就用户数目而言可能极大也可能极小。本发明的方法和系统还可用于更私用的设置中，例如用于公司或政府机构中，以及用在公司、政府机构或任何其他实体之间。



CN 102664728 A

1. 一种用于保护数据的方法，包括：

a) 对数据集合进行加密以提供加密后的数据集合；

b) 从所述加密后的数据集合产生两部分或更多部分数据，其中所述两部分或更多部分数据均包含所述加密后的数据集合的基本上随机化分布；

c) 对来自步骤 b) 的一部分或多部分数据进行加密；以及

d) 将来自步骤 c) 的所述两部分或更多部分数据存储在一个或多个数据仓库的一个或多个位置，由此能够从来自步骤 b) 的所述两部分或更多部分数据中的至少两个部分恢复所述数据集合，其中恢复所述数据集合包括：

对来自步骤 c) 的所述一部分或多部分数据进行解密，

通过重新组合在步骤 b) 中被基本上随机化分布的所述两部分或更多部分数据中的所述至少两个部分的数据，来重构所述加密后的数据集合，以及

将所述加密后的数据集合解密为所述数据集合。

2. 如权利要求 1 所述的方法，其中从所述加密后的数据集合产生两部分或更多部分数据包括产生至少四部分数据。

3. 如权利要求 1 所述的方法，其中在步骤 d) 的所述存储之前步骤 b) 和步骤 c) 被重复一次或多次，并且其中步骤 c) 的所述加密是用与步骤 a) 中的加密算法不同的加密算法来执行的。

4. 如权利要求 1 所述的方法，其中存储所述两部分或更多部分数据包括将所述两部分或更多部分数据存储在相同数据仓库的不同位置上。

5. 如权利要求 1 所述的方法，其中存储所述两部分或更多部分数据包括将所述两部分或更多部分数据存储在不同数据仓库上。

6. 如权利要求 1 所述的方法，其中存储所述两部分或更多部分数据包括将所述两部分或更多部分数据存储于在不同地理位置的不同数据仓库上。

7. 如权利要求 1 所述的方法，其中步骤 c) 的所述加密提供加密钥匙，并且其中所述加密钥匙和步骤 c) 中利用所述加密钥匙加密的数据在步骤 d) 中被存储在一起。

8. 如权利要求 1 所述的方法，其中步骤 c) 中的所述加密提供加密钥匙，并且其中所述加密钥匙和步骤 c) 中利用所述加密钥匙加密的数据在步骤 d) 中被分离地存储。

9. 如权利要求 1 所述的方法，其中步骤 a) 的所述数据集合包括从以下群组中选择的数据，该群组由加密钥匙数据、文本、视频、音频、图像、生物特征量度和数字数据组成。

10. 一种用于保护数据的方法，包括：

a) 从数据集合产生两部分或更多部分数据，其中所述两部分或更多部分数据均包含所述数据集合的基本上随机化分布；

b) 对步骤 a) 的一部分或多部分数据进行加密；以及

c) 将所述两部分或更多部分数据存储在一个或多个数据仓库的一个或多个位置，由此能够从所述两部分或更多部分数据中的至少两个部分恢复所述数据集合，其中恢复所述数据集合包括：

对来自步骤 b) 的所述一部分或多部分数据进行解密，

通过重新组合在步骤 a) 中被基本上随机化分布的所述两部分或更多部分数据中的所述至少两个部分的数据，来重构所述数据集合。

11. 如权利要求 10 所述的方法，其中从所述数据集合产生两部分或更多部分数据包括产生至少四部分数据。

12. 如权利要求 10 所述的方法，其中在步骤 c) 的所述存储之前步骤 a) 和步骤 b) 被重复一次或多次，并且其中步骤 b) 的所述加密是用不同的加密算法来重复的。

13. 如权利要求 10 所述的方法，其中存储所述两部分或更多部分数据包括将所述两部分或更多部分数据存储在相同数据仓库的不同位置上。

14. 如权利要求 10 所述的方法，其中存储所述两部分或更多部分数据包括将所述两部分或更多部分数据存储在不同数据仓库上。

15. 如权利要求 10 所述的方法，其中存储所述两部分或更多部分数据包括将所述两部分或更多部分数据存储于在不同地理位置的不同数据仓库上。

16. 如权利要求 10 所述的方法，其中步骤 b) 的所述加密提供加密钥匙，并且其中所述加密钥匙和步骤 b) 中利用所述加密钥匙加密的数据在步骤 c) 中被存储在一起。

17. 如权利要求 10 所述的方法，其中步骤 b) 中的所述加密提供加密钥匙，并且其中所述加密钥匙和步骤 b) 中利用所述加密钥匙加密的数据在步骤 c) 中被分离地存储。

18. 如权利要求 10 所述的方法，其中步骤 a) 的所述数据集合包括从以下群组中选择的数据，该群组由加密钥匙数据、文本、视频、音频、图像、生物特征量度和数字数据组成。

19. 如权利要求 10 所述的方法，其中步骤 b) 中的所述加密是使用从由 RS1、RC4™ 和 OTP 组成的群组中选择的加密算法执行的。

20. 一种用于保护数据的方法，包括 ：

a) 生成加密主钥并且利用所述加密主钥对数据集合进行加密 ；

b) 根据一个分离模式从所述加密主钥和所述加密后的数据集合产生两部分或更多部分数据，并将加密主钥部分附加到加密后的数据集合部分后，其中所述两部分或更多部分数据包括所述加密后的数据集合中的数据的基本上随机化分布 ；

c) 为来自步骤 b) 的数据部分生成一个或多个加密钥匙，并且利用所述一个或多个加密钥匙对所述数据部分进行加密 ；以及

d) 将来自步骤 c) 的加密数据部分和来自步骤 c) 的所述加密密钥存储在至少一个数据仓库上，由此能够从所述两部分或更多部分数据中的至少两个部分恢复所述数据集合，其中恢复所述数据集合包括 ：

对加密数据部分进行解密，

通过重新组合在步骤 b) 中被基本上随机化分布的所述两部分或更多部分数据中的所述至少两个部分的数据，来重构所述加密后的数据集合，以及

将所述加密后的数据集合解密为所述数据集合。

21. 一种用于保护数据的方法，包括 ：

a) 生成加密主钥并且利用所述加密主钥对数据集合进行加密 ；

b) 根据一个分离模式从所述加密主钥和所述加密后的数据集合产生两部分或更多部分数据，并将加密主钥部分存储在一个或多个数据仓库的一个或多个位置上，其中所述两部分或更多部分数据包括所述加密后的数据集合中的数据的基本上随机化分布 ；

c) 为步骤 b) 的加密后的数据集合部分生成一个或多个加密钥匙，并且利用所述加密钥匙对所述数据部分进行加密 ；以及

d) 将来自步骤 c) 的加密数据部分和来自步骤 c) 的所述加密密钥存储在至少一个数据仓库的至少一个位置上，其中所述数据仓库与步骤 b) 的数据仓库不同，由此能够从所述两部分或更多部分数据中的至少两个部分恢复所述数据集合，其中恢复所述数据集合包括：

对来自步骤 c) 的加密数据部分进行解密，

通过重新组合在步骤 b) 中被基本上随机化分布的所述两部分或更多部分数据中的所述至少两个部分的数据，来重构所述加密后的数据集合，以及

将所述加密后的数据集合解密为所述数据集合。

22. 如权利要求 21 所述的方法，其中步骤 d) 中的所述加密数据部分的存储是在一个数据仓库的两个或更多个不同位置上的。

23. 如权利要求 21 所述的方法，其中步骤 d) 中的所述加密数据部分的存储是在两个或更多个数据仓库上的。

24. 如权利要求 21 所述的方法，其中步骤 d) 中的所述加密钥匙的存储是在一个数据仓库的两个或更多个不同位置上的。

25. 如权利要求 21 所述的方法，其中步骤 d) 中的所述加密钥匙的存储是在两个或更多个不同数据仓库上的。

26. 如权利要求 21 所述的方法，其中根据步骤 d)，在步骤 c) 中生成并在步骤 c) 中用于对数据集合进行加密的所述加密钥匙与使用所述加密密钥进行加密的加密后的数据集合一起存储在一个或多个数据仓库上。

27. 如权利要求 21 所述的方法，其中根据步骤 d)，在步骤 c) 中生成并在步骤 c) 中用于对数据集合进行加密的所述加密钥匙与使用所述加密密钥进行加密的加密后的数据集合存储在一个或多个数据仓库的不同位置。

28. 如权利要求 21 所述的方法，其中步骤 b) 的加密后数据被分离成四个或更多个部分。

29. 如权利要求 21 所述的方法，其中步骤 b) 的所述加密主钥被分离成四个或更多个部分。

30. 如权利要求 21 所述的方法，其中步骤 b) 和步骤 c) 被重复一次或多次，并且，任选地，其中步骤 c) 的所述加密是利用与步骤 a) 中使用的加密算法不同的加密算法来执行的。

31. 一种用于保护数据的系统，包括：

a) 数据分配模块，用于将数据集合的数据分配成至少两个数据部分，其中所述数据以基本上随机的方式被分配成所述至少两个数据部分；

b) 密码处理模块，用于对所述数据集合进行加密；以及

c) 数据组装模块，用于从所述至少两个数据部分组装所述数据集合，其中所述数据组装模块被配置为：

通过重新组合被基本上随机分配的所述至少两个数据部分中的数据，来重构加密后的数据集合，

将所述加密后的数据集合解密为所述数据集合，并且

存储解密后的数据集合。

32. 如权利要求 31 所述的系统，其中所述密码处理模块对所述至少两个数据部分进行加密。

33. 一种用于保护数据的方法，包括：

a) 对数据集合进行加密以提供加密后的数据集合；

b) 根据唯一钥匙值的内容，从所述加密后的数据集合产生两部分或更多部分数据，其中所述加密后的数据集合在所述两部分或更多部分数据之间基本上随机分布；

c) 对来自步骤 b) 的一个或多个数据部分进行加密；以及

d) 将来自步骤 c) 的加密数据部分存储在一个或多个数据仓库的一个或多个位置上，由此能够至少从所述数据部分的子集恢复所述数据集合，其中恢复所述数据集合包括：

对来自步骤 c) 的所述一个或多个数据部分进行解密，

通过重新组合在步骤 b) 中被基本上随机分布的所述两部分或更多部分数据中的所述至少两个部分的数据，来重构所述加密后的数据集合，以及

将所述加密后的数据集合解密为所述数据集合。

34. 如权利要求 33 所述的方法，其中步骤 b) 的产生步骤根据唯一钥匙值的内容产生四部分或更多部分数据。

35. 如权利要求 33 所述的方法，其中所述数据部分包括一个或多个数据比特。

36. 一种用于保护数据的方法，包括：

a) 将数据集合分割成 N 个数据单元；

b) 选择 X 份用于数据单元存储；

c) 生成与所述 X 份相对应的 N 个基本随机数；

d) 将所述随机数分配给所述数据单元；并且

e) 使用电子存储器将所述数据单元和所述随机数存储在与所述随机数相对应的份中，由此能够从所述 X 份的至少子集恢复所述数据集合，其中恢复所述数据集合包括通过根据基本随机数重新组合来自所述 X 份的至少子集的数据单元，来重构所述数据集合。

37. 如权利要求 36 所述的方法，其中所述数据单元包括至少一个比特。

38. 一种用于保护数据集合的方法，包括：

从所述数据集合产生至少两部分数据，其中所述至少两部分数据中的每一个分别包含所述数据集合的相应子集的基本上随机分布；以及

将所述至少两部分数据存储在至少一个数据仓库的至少一个位置，由此通过重新组合基本上随机分布的所述至少两部分数据中的至少两个部分的数据，能够从所述至少两部分数据中的所述至少两个部分恢复所述数据集合。

39. 如权利要求 38 所述的方法，其中产生所述至少两部分数据包括从所述数据集合产生所述至少两部分数据，其中所述至少两部分数据均包含所述数据集合中的数据比特的基本上随机化分布。

40. 如权利要求 38 所述的方法，其中产生所述至少两部分数据包括从所述数据集合产生所述至少两部分数据，其中所述至少两部分数据均包含所述数据集合中的数据字节的基本上随机化分布。

41. 如权利要求 38 所述的方法，其中产生所述至少两部分数据包括从所述数据集合产生所述至少两部分数据，其中所述至少两部分数据均包含所述数据集合中的数据块的基本上随机化分布。

42. 如权利要求 38 所述的方法，其中产生所述至少两部分数据包括使用冗余从所述数

据集合产生所述至少两部分数据。

43. 一种用于保护数据集合的方法，该方法包括：

从所述数据集合中基本上随机地选择第一组数据单元；

从所述数据集合中基本上随机地选择第二组数据单元，其中第一组数据单元和第二组数据单元中的每一组包含的数据单元少于所述数据集合中的全部数据单元；

将第一组数据单元和第二组数据单元分别存储，由此能够从第一组数据单元的至少一部分和第二组数据单元的至少一部分恢复所述数据集合。

44. 如权利要求 43 所述的方法，其中将第一组数据单元和第二组数据单元分别存储包括将第一组数据单元和第二组数据单元存储在同一数据仓库的不同位置上。

45. 如权利要求 43 所述的方法，其中将第一组数据单元和第二组数据单元分别存储包括将第一组数据单元和第二组数据单元存储在不同的数据仓库上。

46. 如权利要求 43 所述的方法，其中将第一组数据单元和第二组数据单元分别存储包括将第一组数据单元和第二组数据单元存储于在不同地理位置的不同数据仓库上。

安全数据解析器方法和系统

[0001] 本申请是国际申请日为 2004 年 6 月 10 日的、名称为“安全数据解析器方法和系统”的发明专利申请 No.200480022710.9(PCT/US2004/018426) 的分案申请。

[0002] 引用相关申请

[0003] 本发明根据 35U.S.C. §120 要求 2003 年 6 月 11 日递交的美国专利申请序列号 10/458,928 的优先权，该美国专利申请序列号 10/458,928 是 2000 年 9 月 20 日递交的同时待决的非临时申请序列号 09/666,519 的部分延续申请，该非临时申请序列号 09/666,519 根据 35U.S.C§199(e) 要求 1999 年 9 月 20 日递交的题为“SECURE SITE FOR INTERNET TRANSCA TIONS(用于互联网事务的安全站点)”的美国临时申请 No.60/154,734 和 2000 年 4 月 27 日递交的题为“SECURE SITE FOR INTERNET TRANSCA TIONS(用于互联网事务的安全站点)”的美国临时申请 No.60/200,396 的优先权。

技术领域

[0004] 本发明一般地涉及用于保护数据不被未经授权地访问或使用的系统。

背景技术

[0005] 在当今社会，个人和企业越来越多地经由计算机系统从事活动。这些计算机系统，其中包括专用和非专用计算机网络，通常存储、归档和传输各种类型的敏感信息。从而，越来越需要确保经由这些系统存储和传输的数据无法被读取或被以其他方式危害。

[0006] 一种用于保护计算机系统的常用解决方案是提供登录和口令功能。但是，口令管理已被证实是代价相当高昂的，因为很大一部分帮助桌面调用都涉及口令问题。此外，口令提供的安全性很小，这是因为它们一般被存储在易于例如通过强力攻击来不适当地访问的文件中。

[0007] 另一种保护计算机系统的解决方案是提供密码基础设施。密码术一般而言是指通过将数据变换或加密成不可读的格式来保护数据。只有拥有加密钥匙者才能将数据解密成可用格式。密码术被用于识别用户，例如认证，以允许访问特权，例如授权，以便创建数字证书和签名等等。一种流行的密码系统是公钥系统，其使用两个钥匙：所有人都知道的公钥和只有其个人或企业拥有者才知道的私钥。一般而言，用一个钥匙加密的数据是用另一个来解密的，并且两个钥匙都不能用另一个来重新创建。

[0008] 不幸的是，即使是前述典型公钥密码系统也仍是高度依赖于用户来获得安全性的。例如，密码系统例如通过用户的浏览器向用户发布私钥。然后不够老练的用户一般将该私钥存储在硬盘驱动器上，而该硬盘驱动器一般是可通过诸如因特网这样的开放计算机系统来被他人访问的。另一方面，用户可能为包含其私钥的文件选择拙劣的名称，例如“key.”。前述和其他行为的结果是使得一个或多个钥匙易遭受危害。

[0009] 除了前述危害外，用户可能将其私钥保存在配置有归档或备份系统的计算机系统上，这可能导致私钥复本传播经过多个计算机存储设备或其他系统。这种安全性漏洞通常被称为“钥匙迁移 (key migration)”。与钥匙迁移类似，许多应用至多只通过简单的登录

和口令访问就提供对用户私钥的访问。如前所述，登录和口令访问通常不提供足够的安全性。

[0010] 用于增大前述密码系统的安全性的一种解决方案是将生物特征量度 (biometric) 包括为认证或授权的一部分。生物特征量度一般包括可测量的物理特性，例如能通过自动化系统来检查的指纹或语音，该自动系统例如是指纹样式或语音样式的样式匹配或识别。在这种系统中，用户的生物特征量度和 / 或钥匙可被存储在移动计算设备上，例如智能卡、笔记本电脑、个人数字助理或移动电话，从而允许生物特征量度或钥匙能在移动环境中被使用。

[0011] 前述移动生物特征量度密码系统仍有多种缺陷。例如，移动用户可能丢失或损坏智能卡或便携式计算设备，从而使其对可能的重要数据的访问完全被切断。或者，某个恶意者可能窃取移动用户的智能卡或便携式计算设备，并使用它来有效地窃取移动用户的数字证书。另一方面，便携式计算设备可能被连接到开放系统，例如因特网，并且正如口令那样，存储生物特征量度的文件可能易遭受由于用户对安全性或恶意入侵者的疏忽而造成的危害。

发明内容

[0012] 基于前述内容，需要提供这样一种密码系统，其安全性是独立于用户的，同时仍支持移动用户。

[0013] 因此，本发明的一个方面是提供一种方法，用于几乎保护任何类型的数据免遭未经授权的访问或使用。该方法包括将要保护的数据解析、分割或分离成两个或更多个部分的一个或多个步骤。该方法还包括对要保护的数据进行加密。数据加密可以在数据的第一解析、分割或分离之前或之后执行。此外，可以为一部分或多部分数据重复加密步骤。类似地，可以为一部分或多部分数据重复解析、分割或分离步骤。该方法还任选地包括将已加密的解析、分割或分离后的数据存储在一个位置或多个位置中。此方法还任选地包括将受保护数据重新构成或重新组装成其原始形式，以供授权访问或使用。此方法可被结合到任何能够执行该方法的所需步骤的计算机、服务器、引擎等的操作之中。

[0014] 本发明的另一个方面提供了一种系统，用于几乎保护任何类型的数据免遭未经授权的访问或使用。此系统包括数据分割模块、密码处理模块，并且任选地包括数据组合模块。在一个实施例中，该系统还可包括可存储安全数据的一个或多个数据存储设施。

[0015] 因此，本发明的一个方面是提供一种安全服务器或信任引擎，其具有以服务器为中心的钥匙，或者换言之，将密钥和用户认证数据存储在服务器上。根据此实施例，用户访问信任引擎以便执行认证和密码功能，例如但不限于认证、授权、数字签署和生成、存储以及证书检索、加密、类似公证人的或类似委托书的动作等等。

[0016] 本发明的另一个方面是提供一种可靠的或受信任的认证过程。此外，在可信的肯定认证之后，可采取多种不同动作，这些动作包括提供密码技术，到系统或设备认证和访问，以允许使用或控制多种电子设备之一。

[0017] 本发明的另一个方面是在密钥和认证数据不被丢失、窃取或危害的环境中提供密钥和认证数据，从而有利地避免了需要不断重新发布和管理新钥匙和认证数据。根据本发明的另一个方面，信任引擎允许用户将一个钥匙对用于多个活动、销售者和 / 或认证请求。

根据本发明的另一个方面，信任引擎至少执行密码处理的一个步骤，例如但不限于服务器方的加密、认证、或签署，从而允许客户端或用户只拥有最少量的计算资源。

[0018] 根据本发明的另一个方面，信任引擎包括一个或多个仓库 (depository)，用于存储每个密钥和认证数据的多个部分。这些部分是通过数据分割过程产生的，该过程阻止了在没有来自一个仓库中的多于一个位置或来自多个仓库的预定部分的情况下进行重建。根据另一个实施例，多个仓库可以是地理上远程的，以便进行欺诈的雇员或一个仓库处的其他受到危害的系统将不会提供对用户的钥匙或认证数据的访问。

[0019] 根据另一个实施例，认证过程有利地允许信任引擎并行处理多个认证活动。根据另一个实施例，信任引擎可有利地跟踪失败访问尝试，从而限制恶意入侵者尝试破坏系统的次数。

[0020] 根据另一个实施例，信任引擎可包括多个实例，其中每个信任引擎可彼此预测和共享处理负载。根据另一个实施例，信任引擎可包括冗余模块，用于轮询多个认证结果以确保多于一个系统认证了用户。

[0021] 因此，本发明的一个方面包括一种可以从远程访问的安全密码系统，用于存储任何类型的数据，其中包括但不限于与多个用户相关联的多个私用密钥。密码系统将多个用户中的每一个与来自多个私用密钥的一个或多个不同钥匙相关联，并且在不向用户发表多个私钥密钥的情况下，利用相关联的一个或多个钥匙为每个用户执行密码功能。密码系统包括具有至少一个服务器的仓库系统，所述服务器存储要保护的数据，例如多个私用密钥和多个注册认证数据。每个注册认证数据标识多个用户之一，并且多个用户中的每一个与来自多个私用密钥的一个或多个不同钥匙相关联。密码系统还可包括认证引擎，其将由多个用户之一接收到的认证数据和与所述多个用户之一相对应的从仓库系统接收来的注册认证数据相比较，从而产生认证结果。密码系统还可包括密码引擎，在认证结果指示所述多个用户之一的正确标识的情况下，该引擎利用从仓库系统接收来的相关联的一个或多个不同钥匙代表多个用户之一执行密码功能。密码还可包括事务引擎，该引擎被连接以将来自多个用户的数据路由到仓库服务器系统、认证引擎和密码引擎。

[0022] 本发明的另一个方面包括安全密码系统，该系统是任选地可从远程访问的。密码系统包括仓库系统，该仓库系统具有至少一个服务器，所述服务器存储至少一个私钥和任何其他数据，例如但不限于与多个注册认证数据，其中每个注册认证数据标识多个可能的用户之一。密码系统还可任选地包括认证引擎，该引擎将由用户接收到的认证数据和与所述用户相对应的从仓库系统接收来的注册认证数据相比较，从而产生认证结果。密码系统还可包括密码引擎，在认证结果指示所述用户的正确标识的情况下，该引擎至少利用可能从仓库系统接收到的所述私钥来代表所述用户执行密码功能。密码还可任选地包括事务引擎，该引擎被连接以将来自用户的数据路由到其他引擎或系统，例如但不限于仓库服务器系统、认证引擎和密码引擎。

[0023] 本发明的另一个方面包括一种辅助密码功能的方法。该方法包括将来自多个用户的一个用户与来自多个私用密钥的一个或多个钥匙相关联，所述多个私用密钥被存储在安全位置上，例如安全服务器上。该方法还包括接收来自用户的认证数据，并且将认证数据和与用户相对应的认证数据相比较，从而验证用户的身份。该方法还包括在不向用户发表所述一个或多个钥匙的情况下利用所述一个或多个钥匙来执行密码功能。

[0024] 本发明的另一个方面包括一种认证系统，用于通过对用户的注册认证数据的安全存储来唯一标识用户。该认证系统包括一个或多个数据存储设施，其中每个数据存储设施包括计算机可访问存储介质，所述介质存储至少一部分注册认证数据。该认证系统还包括认证引擎，该引擎与一个或多个数据存储设施通信。该认证引擎包括：数据分割模块，其对注册认证数据进行操作以产生多个部分；数据组装模块，其处理来自数据存储设施中的至少一个的那些部分，以组装注册认证数据；以及数据比较器模块，其接收来自用户的当前认证数据，并且将当前认证数据与组装后的注册认证数据相比较，以确定用户是否已经被唯一地标识。

[0025] 本发明的另一个方面包括一种密码系统。该密码系统包括一个或多个数据存储设施，其中每个数据存储设施包括计算机可访问存储介质，所述介质存储一个或多个密钥的至少一个部分。该密码系统还包括密码引擎，该引擎与数据存储设施通信。该密码引擎还包括：数据分割模块，其对密钥进行操作以产生多个部分；数据组装模块，其处理来自数据存储设施中的至少一个的那些部分，以组装密钥；以及密码处理模块，其接收组装后的密钥并利用其执行密码功能。

[0026] 本发明的另一个方面包括一种方法，该方法存储任何类型的数据，包括但不限于地理上远程的安全数据存储设施中的认证数据在内，从而保护数据免遭任何个体数据存储设施的合成。该方法包括在信任引擎处接收数据，在信任引擎处利用第一准随机数来组合数据以形成第一组合值，并且利用第二准随机数来组合数据以形成第二组合值。该方法包括利用第二组合值来产生第一准随机数的第一配对，利用第二准随机值来产生第一准随机值的第二配对，并且将第一配对存储在第一安全数据存储设施中。该方法还包括将第二配对存储在相对于第一安全数据存储设施远程的第二安全数据存储设施中。

[0027] 本发明的另一个方面包括一种方法，该方法存储包括但不限于认证数据在内的任何类型的数据，该方法包括接收数据，利用第一比特集合来组合数据以形成第二比特集合，并且利用第三比特集合来组合数据以形成第四比特集合。该方法还包括利用第三比特集合来产生第一比特集合的第一配对。该方法还包括利用第四比特集合来产生第一比特集合的第二配对，并且将第一和第二配对之一存储在第一计算机可访问存储介质中。该方法还包括将第一和第二配对中的另一个存储在第二计算机可访问存储介质中。

[0028] 本发明的另一个方面包括一种方法，该方法存储地理上远程的安全数据存储设施中的密码数据，从而保护密码数据免遭任何个体数据存储设施的合成。该方法包括在信任引擎处接收密码数据，在信任引擎处利用第一准随机数来组合密码数据以形成第一组合值，并且利用第二准随机数来组合密码数据以形成第二组合值。该方法包括利用第二组合值来产生第一准随机数的第一配对，利用第二准随机值来产生第一准随机值的第二配对，并且将第一配对存储在第一安全数据存储设施中。该方法还包括将第二配对存储在相对于第一安全数据存储设施远程的第二安全数据存储设施中。

[0029] 本发明的另一个方面包括一种存储密码数据的方法，该方法包括接收密码数据并且利用第一比特集合来组合密码数据以形成第二比特集合。该方法还包括利用第三比特集合来组合密码数据以形成第四比特集合，利用第三比特集合来产生第一比特集合的第一配对，并且利用第四比特集合来产生第一比特集合的第二配对。该方法还包括并且将第一和第二配对之一存储在第一计算机可访问存储介质中，并且将第一和第二配对中的另一个存

储在第二计算机可访问存储介质中。

[0030] 本发明的另一个方面包括一种在密码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利用敏感数据执行动作期间才以可使用形式存在。该方法还包括在软件模块中接收来自第一计算机可访问存储介质的基本上随机化的或经加密的敏感数据，并且在软件模块中接收来自一个或多个其他计算机可访问存储介质的可能是或可能不是敏感数据的基本上随机化的或经加密的数据。该方法还包括在软件模块中处理基本上随机化的预加密的敏感数据、和可能是或可能不是敏感数据基本上随机化的或经加密的数据，以便组装敏感数据并且在软件引擎中利用敏感数据来执行动作。所述动作包括但不限于认证用户和执行密码功能中的一种。

[0031] 本发明的另一个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定度唯一标识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，该冗余系统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标识。

附图说明

[0032] 以下联系附图更详细描述本发明，附图是用来描述而不是限制本发明的，其中：

[0033] 图 1 示出根据本发明的一个实施例的某些方面的密码系统的框图；

[0034] 图 2 示出根据本发明的一个实施例的某些方面的图 1 的信任 (trust) 引擎的框图；

[0035] 图 3 示出根据本发明的一个实施例的某些方面的图 2 的事务引擎的框图；

[0036] 图 4 示出根据本发明的一个实施例的某些方面的图 2 的仓库的框图；

[0037] 图 5 示出根据本发明的一个实施例的某些方面的图 2 的认证引擎的框图；

[0038] 图 6 示出根据本发明的一个实施例的某些方面的图 2 的密码引擎的框图

[0039] 图 7 示出根据本发明的另一个实施例的某些方面的仓库系统的框图；

[0040] 图 8 示出根据本发明的一个实施例的某些方面的数据分割过程的流程图；

[0041] 图 9 面板 A 示出根据本发明的一个实施例的某些方面的注册 (enrollment) 过程的数据流；

[0042] 图 9 面板 B 示出根据本发明的一个实施例的某些方面的协同工作过程的流程图；

[0043] 图 10 示出根据本发明的一个实施例的某些方面的认证过程的数据流；

[0044] 图 11 示出根据本发明的一个实施例的某些方面的签署过程的数据流；

[0045] 图 12 示出根据本发明的另一个实施例的某些方面的数据流和加密 / 解密过程；

[0046] 图 13 示出根据本发明的另一个实施例的某些方面的信任引擎系统的简化框图；

[0047] 图 14 示出根据本发明的另一个实施例的某些方面的信任引擎系统的简化框图；

[0048] 图 15 示出根据本发明的一个实施例的某些方面的图 14 的冗余模块的框图；

[0049] 图 16 示出根据本发明的一个方面的用于评估认证的过程；

[0050] 图 17 示出根据图 16 所示的本发明的一个方面的分配值给认证的过程；

[0051] 图 18 示出图 17 所示的本发明的一个方面中的用于执行信任仲裁的过程；

[0052] 图 19 示出根据本发明的一个实施例的某些方面的用户和销售者之间的示例性事

务，其中初始的基于 web 的接触导致双方签署销售合约；

[0053] 图 20 示出一个示例性用户系统，其具有密码服务提供者模块，该模块向用户系统提供安全性功能。

[0054] 图 21 示出用于解析、分割或分离数据的过程，其中加密主钥是与数据一起被加密和存储的。

[0055] 图 22 示出用于解析、分割或分离数据的过程，其中加密主钥是单独于数据被加密和存储的。

[0056] 图 23 示出用于解析、分割或分离数据的中间钥过程，其中加密主钥是与数据一起被加密和存储的。

[0057] 图 24 示出用于解析、分割或分离数据的中间钥过程，其中加密主钥是单独于数据被加密和存储的。

[0058] 图 25 示出将本发明的密码方法和系统用于小工作组。

具体实施方式

[0059] 本发明的一个方面是提供一种密码系统，其中一个或多个安全服务器或信任引擎存储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密码系统的功能，但是，信任引擎不发布实际钥匙和其他认证数据，因此钥匙和数据保持安全。这种对钥匙和认证数据的以服务器为中心的存储提供了独立于用户的安全性、便携性、可用性和直观性。

[0060] 因为用户可以相信或信任密码系统以执行用户和文档认证和其他密码功能，所以多种功能可被结合到系统中。例如，信任引擎提供者例如可通过以下方式来确保不受协定拒绝：认证协定参与者，代表参与者以数字方式签署协定，以及存储由每个参与者以数字方式签署的协定的记录。此外，密码系统可监控协定，并且例如基于价格、用户、销售者、地理位置、使用地点等等来确定应用不同程度的认证。

[0061] 为了帮助充分理解本发明，以下详细说明参考附图来描述本发明，附图中类似的元件始终用类似的标号来表示。

[0062] 图 1 示出根据本发明的一个实施例的某些方面的密码系统 100 的框图。如图 1 所示，密码系统 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、证书权力机构 115 和销售者系统 120。

[0063] 根据本发明的一个实施例，用户系统 105 包括传统通用计算机，该传统通用计算机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，如 Windows、Unix、Linux 等等。如图 1 所示，用户系统 105 可包括生物特征量度设备 107。生物特征量度设备 107 可有利地捕捉用户的生物特征量度，并将捕捉到的生物特征量度传送到信任引擎 110。根据本发明的一个实施例，生物特征量度设备可有利地包括具有与以下专利申请中公开的那些类似的属性和特征的设备：1997 年 9 月 5 日递交的题为“RELIEF OBJECTIMAGE GENERATOR(浮凸对象图像生成器)”的美国专利申请 No.08/926,277、2000 年 4 月 26 日递交的题为“IMAGING DEVICE FOR A RELIEF OBJECT AND SYSTEM AND METHOD OF USING THE IMA GE DEVICE(用于浮凸对象的图像设备以及使用该图像设备的系统和方法)”的美国专利 No.09/558,634、1999 年 11 月 5 日递交的题为“RELIEF OBJECT SENSOR ADAPTOR(浮凸对象传感器适配器)”的美国专

利申请 No.09/435,011 以及 2000 年 1 月 5 日递交的题为“PLANAR OPTICAL IMAGE SENSOR AND SYSTEM FOR GENERATING AN ELECTRONIC IMAGE OF A RELIEF OBJECT FOR FINGERPRINT READING(用于指纹读取的平面光学图像传感器和生成浮凸对象的电子图像的系统)”的美国专利申请 No.09/477,943,所有这些专利申请都属于当前的受让人所有,并且在这里通过引用将这些专利申请包含进来。

[0064] 此外,用户系统 105 可通过传统服务提供者连接到通信链路 125,所述服务提供者例如是拨号、数字用户线路 (DSL)、电缆调制解调器、光纤连接等等。根据另一个实施例,用户系统 105 通过网络连接连接到通信链路 125,所述网络连接例如是局域网或广域网。根据一个实施例,操作系统包括处理经由通信链路 125 传递的所有传入和传出消息流量的 TCP/IP 栈。

[0065] 虽然用户系统 105 是参考前述实施例公开的,但是本发明不限于此。相反,本领域的普通技术人员将会从此处的公开文本中认识到,用户系统 105 的大量备选实施例,其中几乎包括任何能够发送信息到另一个计算机系统或从另一个计算机系统接收信息的计算设备。例如,用户系统 105 可以包括但不限于可以与通信链路 125 交互的计算机工作站、交互式电视机、移动电话、笔记本电脑等、无线通信设备、智能卡、嵌入式计算设备等。在这种备选系统中,操作系统很可能会是不同的,并且适应于特定设备。但是,根据一个实施例,操作系统有利地持续提供与通信链路 125 建立通信所需的适当通信协议。

[0066] 图 1 示出信任引擎 110。根据一个实施例,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,所述敏感信息可以是任何类型或形式的数据,例如但不限于文本、音频、视频、用户认证数据和公共密钥和私用密钥。根据一个实施例,认证数据包括被设计为唯一标识密码系统 100 的用户的数据。例如,认证数据可包括用户标识号码、一个或多个生物特征量度以及由信任引擎 110 或用户生成的但最初由用户在注册时回答的一系列问题和答案。前述问题可包括 :人口统计数据,例如出生地点、地址、周年纪念等等 ;个人数据,例如母亲未婚时的名字、最爱的冰淇淋等等 ;或者被设计为唯一标识用户的其他数据。信任引擎 110 将与当前事务相关联的用户认证数据与较早时 (例如注册期间) 提供的认证数据相比较。信任引擎 110 可有利地要求用户在每次事务时产生认证数据,或者,信任引擎 110 可有利地允许用户周期性地产生认证数据,例如在一串事务开始时或登录到特定销售者网站上时。

[0067] 根据用户产生生物特征量度数据的实施例,用户向生物特征量度设备 107 提供物理特征,例如但不限于面部扫描、手部扫描、耳部扫描、虹膜扫描、视网膜扫描、血管样式、DNA、指纹、笔迹或语音。生物特征量度设备有利地产生该物理特征的电子样式或生物特征量度。该电子样式通过用户系统 105 被传送到信任引擎 110,以用于注册或认证目的。

[0068] 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) 和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 就向用户提供完整的密码功能。例如,被适当认证的用户可有利地采用信任引擎 110 来执行哈西处理、数字签署、加密和解密 (常被一起称为加密)、创建或分布数字证书等等。但是,在信任引擎 110 外部将不能获得密码功能中使用的私用密钥,从而确保了密钥的完好性。

[0069] 根据一个实施例,信任引擎 110 生成和存储密钥。根据另一个实施例,至少一个密钥与每个用户相关联。另外,当密钥包括公钥技术时,与用户相关联的每个私钥在信任引擎

110 内被生成，并且不被从信任引擎 110 中发表出去。从而，只要用户具有对信任引擎 110 的访问权限，用户就能用他或她的私钥或公钥来执行密码功能，这种远程访问有利地允许了用户完全保持移动并且实际上通过任何因特网连接来访问密码功能，所述因特网连接例如是蜂窝和卫星电话、公用电话亭、笔记本、旅馆房间等等。

[0070] 根据另一个实施例，信任引擎 110 利用为信任引擎 110 生成的钥匙对执行密码功能。根据此实施例，信任引擎 110 首先认证用户，并且在用户已正确地产生与注册认证数据匹配的认证数据之后，信任引擎 110 使用其自己的密钥对来代表被认证的用户执行密码功能。

[0071] 本领域的技术人员将会从此处的公开文本意识到密钥可以有利地包括对称钥匙、公钥和私钥中的某些或全部。此外，本领域的技术人员将会从此处的公开文本中意识到，前述钥匙可以利用多种可从商业技术获得的算法来实现，所述商业技术例如是 RSA、ELGAMAL 等等。

[0072] 图 1 还示出了证书权力机构 115。根据一个实施例，证书权力机构 115 可以有利地包括发布数字证书的受信任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协议（例如 PKCS10）将对于数字证书的请求发送到证书权力机构 115。作为响应，证书权力机构 115 将会以一个或多个不同的协议（例如 PKCS7）来发布数字证书。根据本发明的一个实施例，信任引擎 110 请求来自著名的证书权力机构 115 中的几个或全部的数字证书，以便信任引擎 110 具有对于与任何请求方的证书标准相对应的数字证书的访问权限。

[0073] 根据另一个实施例，信任引擎 110 内部执行证书发布。在此实施例中，信任引擎 110 可访问用于生成证书的证书系统和 / 或可以在证书被请求时内部生成证书，例如在钥匙生成之时或者在请求之时请求的证书标准中。信任引擎 110 将在下文中更详细公开。

[0074] 图 1 还示出了销售者系统 120。根据一个实施例，销售者系统 120 有利地包括 Web 服务器。典型 Web 服务器一般利用几种互联网标记语言或文档格式标准之一经由因特网提供内容，所述互联网标记语言或文档格式标准例如是超文本标记语言 (HTML) 或可扩展标记语言 (XML)。Web 服务器接收来自诸如 Netscape 和 Internet Explorer 这样的浏览器的请求，然后返回适当的电子文档。多种服务器或客户端侧技术可用于将 Web 服务器的效力增大到超出其递送标准电子文档的能力。例如，这些技术包括公共网关接口 (CGI) 脚本、安全套接字层 (SSL) 安全性和活动服务器页面 (ASP)。销售者系统 120 可有利地提供关于商业、个人、教育或其他事务的电子内容。

[0075] 虽然是参考前述实施例来公开销售者系统 120 的，但是本发明不限于此。相反，本领域的普通技术人员将会从此处的公开文本中意识到销售者系统 120 可有利地包括参考用户系统 105 描述的设备中的任何一个或其组合。

[0076] 图 1 还示出了连接用户系统 105、信任引擎 110、证书权力机构 115 和销售者系统 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。本公开文本中所使用的因特网是全球计算机网络。因特网的结构是本领域的普通技术人员所公知的，其包括网络中枢和从该中枢出来的网络分支。这些分支又具有从它们出来的网络分支，以此类推。路由器在网络层次之间移动信息分组，然后在网络间移动信息分组，直到分组到达其目的地的邻居。从该目的地中，目的地网络的主机将信息分组引导到适当的终端或节点。在一

个有利的实施例中，正如本领域中公知的，因特网路由选择集线器包括使用传输控制协议 / 因特网协议 (TCP/IP) 的域名系统 (DNS) 服务器。路由选择集线器经由高速通信链路连接到一个或多个路由选择集线器。

[0077] 因特网的一个当前流行的部分是万维网。万维网包含不同计算机，这些计算机存储能够显示图形和文本信息的文档。在万维网上提供信息的计算机通常被称为“网站”。网站是由具有相关联的电子页面的因特网地址来定义的。电子页面可由统一资源定位符 (URL) 来标识。一般而言，电子页面是组织文本、图形图像、音频、视频等等的呈现的文档。

[0078] 虽然通信链路 125 是按照其优选实施例被公开的，但是本领域的普通技术人员将会从此处的公开文本中认识到通信链路 125 可包括多种交互式通信链路。例如，通信链路 125 可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、定制的私用或公共计算机网络、交互式公用电话亭网络、自动出纳机网络、直接链路、卫星或蜂窝网络等等。

[0079] 图 2 示出根据本发明的一个实施例的某些方面的图 1 的信任引擎 110 的框图。如图 2 所示，信任引擎 110 包括事务引擎 205、仓库 210、认证引擎 215 和密码引擎 220。根据本发明的一个实施例，信任引擎 110 还包括大容量存储装置 225。正如图 2 中进一步示出的，事务引擎 205 与仓库 210、认证引擎 215 和密码引擎 220 以及大容量存储装置 225 通信。此外，仓库 210 与认证引擎 215、密码引擎 220 和大容量存储装置 225 通信。另外，认证引擎 215 与密码引擎 220 通信。根据本发明的一个实施例，前述通信中的某些或全部可有利地包括将 XML 文档传输到与接收设备相对应的 IP 地址。如前所述，XML 文档有利地允许设计者创建它们自己的定制的文档标签，从而使得能够定义、传输、验证和解释应用之间以及组织之间的数据。另外，前述通信中的某些或全部可包括传统 SSL 技术。

[0080] 根据一个实施例，事务引擎 205 包括数据路由选择设备，例如可从 Netscape、Microsoft、Apache 等获得的传统 Web 服务器。例如，Web 服务器可有利地接收来自通信链路 125 的传入数据。根据本发明的一个实施例，传入数据是寻址到用于信任引擎 110 的前端安全措施系统的。例如，前端安全措施系统可有利地包括防火墙、搜索已知攻击简档的入侵检测系统和 / 或病毒扫描器。在通过前端安全措施系统之后，数据被事务引擎 205 接收，并被路由到仓库 210、认证引擎 215、密码引擎 220 和大容量存储装置 225 之一。此外，事务引擎 205 监控来自认证引擎 215 和密码引擎 220 的传入数据，并且通过通信链路 125 将数据路由到特定系统。例如，事务引擎 205 可有利地将数据路由到用户系统 105、证书权力机构 115 或销售者系统 120。

[0081] 根据一个实施例，数据是用传统 HTTP 路由选择技术来路由的，例如用 URL 或统一资源指示符 (URI) 来路由。URI 与 URL 类似，但是，URI 通常指示文本或动作的源，例如可执行文件、脚本等等。因此，根据一个实施例，用户系统 105、证书权力机构 115、销售者系统 120 和仓库 210 的组件有利地在通信 URL 或 URI 内包括充足的数据，以便事务引擎 205 在整个密码系统中适当地路由数据。

[0082] 虽然数据路由选择是参考其优选实施例来公开的，但是本领域的技术人员将会意识到许多可能的数据路由选择解决方案或策略。例如，XML 或其他数据分组可有利地被解封装并按照其格式、内容等被识别，以使得事务引擎 205 可在整个信任引擎 110 中适当地路由数据。此外，本领域的技术人员将会意识到，例如当通信链路 125 包括本地网络时，数据路由选择可有利地适应于符合特定网络系统的数据传送协议。

[0083] 根据本发明的另一个实施例，事务引擎 205 包括传统 SSL 加密技术，以使得在特定通信期间，前述系统可向事务引擎 205 认证其自身，反之亦然。在整个公开文本中使用的术语“1/2SSL”是指服务器被 SSL 认证但客户端不一定被 SSL 认证的通信，术语“完全 SSL”是指客户端和服务器被 SSL 认证的通信。在当前的公开文本使用术语“SSL”时，通信可包括 1/2 或完全 SSL。

[0084] 在事务引擎 205 将数据路由到密码系统 100 的各种组件时，事务引擎 205 可有利地创建审计追踪 (audit trail)。根据一个实施例，审计追踪至少包括由事务引擎 205 在整个密码系统 100 中路由的数据的类型和格式的记录。这种审计数据可有利地被存储在大容量存储装置 225 中。

[0085] 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施，例如目录服务器、数据库服务器等等。如图 2 所示，仓库 210 存储密钥和注册认证数据。密钥可有利地对应于信任引擎 110 或对应于密码系统 100 的用户，例如用户或销售者。注册认证数据可有利地包括被设计为唯一标识用户的数据，例如用户 ID、密码、问题的答案、生物特征量度数据等等。此注册认证数据可有利地在用户注册时或稍后的其他时间被获取。例如，信任引擎 110 可包括注册认证数据的周期性或其他的更新或重新发布。

[0086] 根据一个实施例，从事务引擎 205 到认证引擎 215 和密码引擎 220 以及从认证引擎 215 和密码引擎 220 到事务引擎 205 的通信包括安全通信，例如传统 SSL 技术。此外，如前所述，去往和来自仓库 210 的通信的数据可利用 URL、URI、HTTP 或 XML 文档来传送，并且前述中的任何一个有利地在其中嵌入了数据请求和格式。

[0087] 如上所述，仓库 210 可有利地包括多个安全数据存储设施。在这种实施例中，安全数据存储设施可被配置为使得对一个数据存储设施中的安全性的危害不会危害到存储在其中的密钥或认证数据。例如，根据此实施例，对密钥和认证数据进行数学运算，以便使存储在每个数据存储设施中的数据从统计上而言基本上随机化。根据一个实施例，单个数据存储设施的数据的随机化使得该数据无法被解密。从而，对单个数据存储设施的危害只会产生随机化的无法解密的数据，而不会从整体上危害任何密钥或认证数据的安全性。

[0088] 图 2 还示出包括认证引擎 215 的信任引擎 110。根据一个实施例，认证引擎 215 包括被配置为将来自事务引擎 205 的数据与来自仓库 210 的数据进行比较的数据比较器。例如，在认证期间，用户将当前认证数据提供给信任引擎 110，以便事务引擎 205 接收到当前认证数据。如前所述，事务引擎 205 识别优选地采取 URL 或 URI 形式的数据请求，并且将认证数据路由到认证引擎 215。此外，在请求时，仓库 210 将与用户相对应的注册认证数据转发到认证引擎 215。从而，认证引擎 215 既有当前认证数据，又有注册认证数据，以便比较。

[0089] 根据一个实施例，去往认证引擎的通信包括安全通信，例如 SSL 技术。另外，可在信任引擎 110 组件内提供安全措施，例如利用公钥技术的超级加密 (super-encryption)。例如，根据一个实施例，用户利用认证引擎 215 的公钥加密当前认证数据。此外，仓库 210 还利用认证引擎 215 的公钥加密注册认证数据。通过这种方式，只有认证引擎的私钥能被用于解密传输。

[0090] 如图 2 所示，信任引擎 110 还包括密码引擎 220。根据一个实施例，密码引擎 220 包括密码处理模块，该模块被配置为有利地提供传统密码功能，例如公钥基础设施 (PKI) 功能。例如，密码引擎 220 可有利地为密码系统 100 的用户发布公钥和私钥。通过这种方式，

密钥在密码引擎 220 处被生成，并被转发到仓库 210，以使得至少私用密钥在信任引擎 110 外不可用。根据另一个实施例，密码引擎 220 至少对私用密密钥数据进行随机化和分割，从而只存储经随机化的分割后数据。与注册认证数据的分割类似，分割过程确保存储的钥匙在密码引擎 220 外部不可用。根据另一个实施例，密码引擎的功能可以与认证引擎 215 相结合，并由认证引擎 215 来执行。

[0091] 根据一个实施例，去往和来自密码引擎的通信包括安全通信，例如 SSL 技术。此外，有利地采用了 XML 文档以传送数据和 / 或做出密码功能请求。

[0092] 图 2 还示出了具有大容量存储装置 225 的信任引擎 110。如前所述，事务引擎 205 保存与审计追踪相对应的数据，并将这种数据存储在大容量存储装置 225 中。类似地，根据本发明的一个实施例，仓库 210 保存与审计追踪相对应的数据，并将这种数据存储在大容量存储装置 225 中。仓库审计追踪数据与事务引擎 205 的审计追踪数据的相同之处在于该审计追踪数据包括仓库 210 接收到的请求及其响应的记录。此外，大容量存储装置 225 可用于存储其中包含了用户公钥的数字证书。

[0093] 虽然信任引擎 110 是参考其优选和备选实施例来公开的，但是本发明不想要限于此。相反，本领域的技术人员将会在此处的公开文本中意识到信任引擎 110 的多种备选方案。例如，信任引擎 110 可有利地只执行认证，或者只执行密码功能中的某些或全部，例如数据加密和解密。根据这种实施例，认证引擎 215 和密码引擎 220 之一可有利地被去除，从而产生信任引擎 110 的更直观的设计。此外，密码引擎 220 还可与证书权力机构通信，以便证书权力机构被包含在信任引擎 110 内。根据另一个实施例，信任引擎 110 可有利地执行认证和一个或多个密码功能，例如数字签署。

[0094] 图 3 示出根据本发明的一个实施例的某些方面的图 2 的事务引擎 205 的框图。根据此实施例，事务引擎 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 可有利地与在传统高容量服务器中找到的那些类似，所述服务器例如是可从 Apache 获得的 Web 服务器。监听线程监控来自通信链路 125、认证引擎 215 和密码引擎 220 之一的传入通信以便获得传入数据流。处理线程识别传入数据流的特定数据结构，例如前述数据结构，从而将传入数据路由到通信链路 125、仓库 210、认证引擎 215、密码引擎 220 或大容量存储装置 225 之一。如图 3 所示，传入和传出数据可以有利地例如通过 SSL 技术而被保护。

[0095] 图 4 示出根据本发明的一个实施例的某些方面的图 2 的仓库 210 的框图。根据此实施例，仓库 210 包括一个或多个轻型目录访问协议 (LDAP) 服务器。LDAP 目录服务器可从诸如 Netscape、ISO 等等的多种制造商获得。图 4 还示出目录服务器优选地存储与密钥相对应的数据 405 和与注册认证数据相对应的数据 410。根据一个实施例，仓库 210 包括唯一用户 ID 的单个逻辑存储器结构索引认证数据和密密钥数据。单个逻辑存储器结构优选地包括用于确保存储在其中的数据的高度可信度或安全性的机制。例如，仓库 210 的物理位置可有利地包括多种传统安全性措施，例如有限雇员访问、现代监视系统等等。此外，或者作为替代，物理安全措施、计算机系统或服务器可有利地包括用于保护存储的数据的软件解决方案。例如，仓库 210 可有利地创建和存储与所采取的动作的审计追踪相对应的数据 415。此外，传入和传出通信可有利地被用耦合到传统 SSL 技术的公钥加密来加密。

[0096] 根据另一个实施例，仓库 210 可包括不同的并且物理上相分离的数据存储设施，正如以下参考图 7 进一步公开的。

[0097] 图 5 示出根据本发明的一个实施例的某些方面的图 2 的认证引擎 215 的框图。与图 3 的事务引擎 205 类似，认证引擎 215 包括至少具有传统 Web 服务器的经修改版本的监听和处理线程的操作系统 505，所述服务器例如是可从 Apache 获得的 Web 服务器。如图 5 所示，认证引擎 215 包括对至少一个私钥 510 的访问权限。私钥 510 可有利地被用于例如对来自事务引擎 205 或仓库 210 的数据进行解密，这些数据是用认证引擎 215 的相应的公钥来加密的。

[0098] 图 5 还示出包括比较器 515、数据分割模块 520 和数据组装模块 525 的认证引擎 215。根据本发明的优选实施例，比较器 515 包括能够对与前述生物特征量度认证数据相关的可能的复杂样式进行比较的技术。该技术可包括用于诸如指纹样式或嗓音样式这样的样式比较的硬件、软件或组合解决方案。此外，根据一个实施例，认证引擎 215 的比较器 515 可有利地比较文档的传统哈西码，以便给出比较结果。根据本发明的一个实施例，比较器 515 包括将探试 (heuristic)530 应用到比较。探试 530 可有利地针对处理认证尝试周围的环境，例如时间、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。

[0099] 此外，生物特征量度数据比较的性质将会导致从当前生物特征量度认证数据与注册数据的匹配中产生不同程度的信心。例如，与返回肯定或否定匹配的传统口令不同，指纹可以被确定为部分匹配，例如 90％匹配、75％匹配或 10％匹配，而不仅是正确或不正确。其他的诸如声印分析或面部识别这样的生物特征量度识别器可共享此概率认证的属性，而不是绝对认证。

[0100] 当利用这种概率认证进行工作时，或在认证被视为不够绝对可靠的其他情形下，需要应用探试 530 来确定对所提供的认证的信心级别是否足够高到认证正在进行的事务。

[0101] 有时候可能是这种情况，即所讨论的事务是相对低价值的事务，此时被认证到较低的信心级别是可接受的。这可包括有较低的美元值与其相关联的事务（例如 $10 的购买）或风险低的事务（例如准许进入只限成员的网站）。

[0102] 相反，对于认证其他事务，在允许事务继续下去之前，可能需要要求对认证的高度信心。这种事务可包括较高美元值的事务（例如签署数百万美元的供应合同）或者如果发生不适当的认证则具有高风险的事务（例如远程登录到政府计算机上）。

[0103] 正如下文所述，结合信心级别和事务值使用探试 530 可允许比较器提供动态的对上下文敏感的认证系统。

[0104] 根据本发明的另一个实施例，比较器 515 可有利地跟踪特定事务的认证尝试。例如，当事务失败时，信任引擎 110 可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 可有利地采用尝试限制器 535 来限制认证尝试的数据，从而阻止强力尝试模仿用户的认证数据。根据一个实施例，尝试限制器 535 包括监控事务的重复认证尝试并例如将给定事务的认证尝试限制到 3 的软件模块。从而，例如，尝试限制器 535 将会把用于模仿个人的认证数据的自动尝试限制到只是三次“猜测”。在三次失败之后，尝试限制器 535 可有利地拒绝更多的认证尝试。这种拒绝例如可有利地通过以下方式来实现：不论正在传输的当前认证数据为何，比较器 515 都返回否定结果。另一方面事务引擎 205 可阻止与其中先前三次尝试已失败的事务有关的任何更多的认证尝试。

[0105] 认证引擎 215 还包括数据分割模块 520 和数据组装模块 525。数据分割模块 520 有利地包括具有以下能力的软件、硬件或组合模块：对各种数据进行数学运算，以便基本上

将数据随机化并割成多个部分。根据一个实施例，无法从单个部分重新创建原始数据。数据组装模块 525 有利地包括被配置为进行以下操作的软件、硬件或组合模块 :对前述基本上经随机化的部分进行数学运算，以便其组合提供原始的解码后的数据。根据一个实施例，认证引擎 215 采用数据分割模块 520 来将注册认证数据随机化并将其分割成多个部分，并采用数据组装模块 525 来将这些部分重新组装回可使用的注册认证数据。

[0106] 图 6 示出根据本发明的一个实施例的某些方面的图 2 的密码引擎 220 的框图。与图 3 的 205 类似，密码引擎 220 包括至少具有传统 Web 服务器的经修改版本的监听和处理线程的操作系统 605，所述服务器例如是可从 Apache 获得的 Web 服务器。如图 6 所示，密码引擎 220 包括与图 5 中的那些功能类似的数据分割模块 610 和数据组装模块 620。但是，根据一个实施例，数据分割模块 610 和数据组装模块 620 处理密钥数据，而不是前述的注册认证数据。虽然，本领域的技术人员将会从此处的公开文本意识到，数据分割模块 610 和数据组装模块 620 可以与认证引擎 215 的那些相组合。

[0107] 密码引擎 220 还包括密码处理模块 625，该模块被配置为用于执行多种密码功能中的某些或全部。根据一个实施例，密码处理模块 625 可以包括软件模块或程序、硬件、或两者。根据另一个实施例，密码处理模块 625 可执行数据比较、数据解析、数据分割、数据分离、数据散列、数据加密或解密、数字签名验证或创建、数据证书生成、存储或请求、密钥生成等等。此外，本领域的技术人员将会从此处的公开文本中意识到，密码处理模块 625 可有利地包括公钥基础设施，例如良好稳私 (Pretty Good Privacy，PGP)，基于 RSA 的公钥系统或者多种备选钥匙管理系统。此外，密码处理模块 625 可执行公钥加密、对称钥匙加密或两者。除了前述以外，密码处理模块 625 可包括一个或多个计算机程序或模块、硬件、或两者，用于实现无缝透明的协同工作功能。

[0108] 本领域的技术人员将会从此处的公开文本中意识到密码功能可包括大量或多种通常与密钥管理系统相关的功能。

[0109] 图 7 示出根据本发明的一个实施例的某些方面的仓库系统 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施，例如数据存储设施 D1、D2、D3 和 D4。但是，本领域的普通技术人员易于理解，仓库系统可以只具有一个数据存储设施。根据本发明的一个实施例，数据存储设施 D1 至 D4 中的每一个可有利地包括参考图 4 的仓库 210 公开的元件中的某些或全部。与仓库 210 类似，数据存储设施 D1 至 D4 优选地通过传统 SSL 与事务引擎 205、认证引擎 215 和密码引擎 220 通信。通信链路例如传送 XML 文档。来自事务引擎 205 的通信可有利地包括对数据的请求，其中请求被有利地广播到每个数据存储设施 D1 至 D4 的 IP 地址。另一方面，事务引擎 205 可基于许多标准将请求广播到特定数据存储设施，所述标准例如是响应时间、服务器负载、维护时间安排等等。

[0110] 响应于来自事务引擎 205 的对数据的请求，仓库系统 700 有利地将存储的数据转发到认证引擎 215 和密码引擎 220。各自的数据组装模块接收被转发的数据并将数据组装成可使用的格式。另一方面，从认证引擎 215 和密码引擎 220 到数据存储设施 D1 至 D4 的通信可包括对要存储的敏感数据的传输。例如，根据一个实施例，认证引擎 215 和密码引擎 220 可有利地采用其各自的数据分割模块来将敏感数据划分成不可解密的多个部分，然后将敏感数据的一个或多个不可解密的部分发送到特定的数据存储设施。

[0111] 根据一个实施例，每个数据存储设施 D1 至 D4 包括单独且独立的存储系统，例如目

录服务器。根据本发明的一个实施例，仓库系统 700 包括多个地理上相分离的独立数据存储系统。通过将敏感数据分布到不同且独立的存储设施 D1 至 D4（其中的某些或全部可能有利地从地理上而言是分离的），仓库系统 700 提供了冗余性以及额外的安全措施。例如，根据一个实施例，要解密和重新组装敏感数据，只需要来自多个数据存储设施 D1 至 D4 中的两个的数据。从而，四个数据存储设施 D1 至 D4 中可以有两个由于维护、系统失败、断电等等而无法工作，而不会影响信任引擎 110 的功能。此外，根据一个实施例，因为存储在每个数据存储设施中的数据已随机化并且是不可解密的，因此对任何单个数据存储设施的危害都不一定会危害到敏感数据。此外，在数据存储设施从地理上相分离的实施例中，对于多个地理上远程的设施的危害变得越来越困难。实际上，即使是进行欺诈的雇员要想破坏所需的多个独立的地理上远程的数据存储设施，也会受到巨大挑战。

[0112] 虽然仓库系统 700 是参考其优选和备选实施例来公开的，但是本发明不想要限于此。相反，本领域的技术人员将会从此处的公开文本中意识到仓库系统 700 的多种备选方案。例如，仓库系统 700 可包括一个、两个或更多个数据存储设施。此外，可以对敏感数据进行数学运算，以便要重新组装和解密敏感数据需要来自两个或更多个数据存储设施的部分。

[0113] 如前所述，认证引擎 215 和密码引擎 220 各自分别包括数据分割模块 520 和 610，用于分割任何类型或形式的敏感数据，例如文本、音频、视频、认证数据和密钥数据。图 8 示出根据本发明的一个实施例的某些方面的由数据分割模块执行的数据分割过程 800 的流程图。如图 8 所示，数据分割过程 800 开始于步骤 805 处，此时敏感数据“S”被认证引擎 215 或密码引擎 220 的数据分割模块所接收。优选地，在步骤 810 中，数据分割模块随后生成基本上随机的数字、值或串或比特集合，“A”。例如，随机数 A 可以是用本领域的普通技术人员可获得的用于产生适用于密码应用的高质量随机数的多种不同的传统技术来生成的。此外，根据一个实施例，随机数 A 包括可以是任何合适的长度的比特长度，例如短于、长于或等于敏感数据 S 的比特长度。

[0114] 此外，在步骤 820 中，数据分割过程 800 生成另一个统计上随机的数字“C”。根据优选实施例，统计上随机的数字 A 和 C 的生成有利地是并行完成的。然后数据分割模块将数字 A 和 C 与敏感数据 S 相组合，以便生成新的数字“B”和“D”。例如，数 B 可以包括 A XOR S 的二进制组合，数字 D 可包括 C XOR S 的二进制组合。XOR 函数或“异或”函数是本领域的普通技术人员所公知的。前述组合优选地分别发生在步骤 825 和 830 中，并且根据一个实施例，前述组合也是并行发生的。然后 800 前进到步骤 835，在这里随机数 A 和 C 以及数字 B 和 D 被配对，以便没有哪一配对本身包含足以来重新组织和解密原始敏感数据 S 的数据。例如，可按如下方式对数字配对 :AC、AD、BC 和 BD。根据一个实施例，前述配对中的每一个被分布到图 7 的仓库 D1 至 D4 之一。根据另一个实施例，前述配对中的每一个被随机分布到仓库 D1 至 D4 之一。例如，在第一数据分割过程 800 期间，可通过对 D2 的 IP 地址的随机选择而将配对 AC 发送到仓库 D2。然后，在第二数据分割过程 800 期间，可通过对 D4 的 IP 地址的随机选择将配对 AC 发送到仓库 D4。此外，这些配对可以都被存储在一个仓库上，并且可以被存储在所述仓库的相分离的位置中。

[0115] 基于前述，数据分割过程 800 有利地将敏感数据的多个部分放置在四个数据存储设施 D1 至 D4 中的每一个中，以使得没有一个数据存储设施 D1 至 D4 包括足以重新创建原

始敏感数据 S 的加密数据。如前所述，这种将数据随机化成多个单独不可用的加密部分增大了安全性，并且即使数据存储设施 D1 至 D4 之一受到危害也保持了数据的可信度。

[0116] 虽然数据分割过程 800 是参考其优选实施例来公开的，但是本发明不想要限于此。相反，本领域的技术人员将会从此处的公开文本中意识到数据分割过程 800 的许多备选方案。例如，数据分割过程可有利地将数据分割成两个数字，例如随机数 A 和随机数 B，并且通过两个数据存储设施随机分布 A 和 B。此外，数据分割过程 800 可有利地通过生成更多随机数将数据分割在多个数据存储设施间。数据可被分割成任何所需的、所选择的、预定的或随机分配的大小的单元，包括但不限于一个比特、多个比特、字节、千字节、兆字节或更多，或这些大小的任何组合或序列。此外，改变产生自分割过程的数据单元的大小将会使数据更难以恢复到可使用形式，从而增大敏感数据的安全性。本领域的普通技术人员易于认识到，分割后的数据单元大小可以是多种数据单元大小或大小样式或大小组合。例如，数据单元大小可被选择或预定为是全都是相同大小、具有不同大小的固定集合、大小组合或随机生成大小。类似地，可根据每份数据单元固定或预定数据单元大小、数据单元大小的样式或组合，或随机生成的一个或多个大小，来将数据单元分布到一份或多份中。

[0117] 如前所述，为了重新创建敏感数据 S，数据部分需要被解随机化并重新组织。此过程可有利地发生在认证引擎 215 和密码引擎 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施 D1 至 D4 的数据部分，并将数据重新组装成可使用的形式。例如，根据数据分割模块 520 采取图 8 的数据分割过程 800 的一个实施例，数据组装模块 525 使用来自数据存储设施 D1 至 D4 中的至少两个的数据部分，以重新创建敏感数据 S。例如，配对 AC、AD、BC 和 BD 被分布，以使得任何两个都提供 A 和 B 或 C 和 D 之一。注意到 S = A XOR B 或 S = C XOR D 指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 可有利地重新组装敏感数据 S。从而数据组装模块 525 例如可在其至少接收到来自数据存储设施 D1 至 D4 的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 做出的组装请求。

[0118] 基于上述数据分割和组装过程，敏感数据 S 仅在信任引擎 110 的有限区域中以可使用形式存在。例如，当敏感数据 S 包括注册认证数据时，可使用的未随机化的注册认证数据仅能在认证引擎 215 中获得。类似地，当敏感数据 S 包括私用密密钥数据，可使用的未随机化的私用密密钥数据仅能在密码引擎 220 中获得。

[0119] 虽然数据分割和组装过程是参考其优选实施例来公开的，但是本发明并不想限于此。相反，本领域的技术人员将会从此处的公开文本中意识到分割和重新组装敏感数据 S 的多种备选方案。例如，公钥加密可用于进一步保护数据存储设施 D1 至 D4 处的数据。此外，本领域的普通技术人员将易于认识到这里所描述的数据分割模块也是本发明的单独的不同的实施例，它可被结合到任何现有计算机系统、软件套组、数据库或其组合中或本发明的任何实施例中，或与其相组合，或以其他方式作为其一部分，所述本发明的实施例例如是这里公开和描述的信任引擎、认证引擎和事务引擎。

[0120] 图 9A 示出了根据本发明的一个实施例的某些方面的注册过程 900 的数据流。如图 9A 所示，注册过程 900 开始于步骤 905 处，此时用户希望向密码系统 100 的信任引擎 110 注册。根据此实施例，用户系统 105 有利地包括客户端侧小应用程序 (applet)，例如基于 Java 的小应用程序，这种 applet 查询用户以要求其输入注册数据，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、生物特征量度等等。根据一个实施例，在查询过程期间，客户端侧小应用程序优选地与信任引擎 110 通信以确保所选中的用户 ID 是唯一的。当用户 ID 非唯一时，信任引擎 110 可有利地建议唯一的用户 ID。客户端侧小应用程序收集注册数据，并且例如通过 XML 文档将注册数据发送到信任引擎 110，尤其是到事务引擎 205。根据一个实施例，该传输被认证引擎 215 的公钥编码。

[0121] 根据一个实施例，在注册过程 900 的步骤 905 期间，用户执行单次注册。例如，用户将其自己注册为特定的人，例如 Joe User。当 JoeUsser 需要注册为 Joe User，Mega 公司的 CEO 时，那么根据此实施例，Joe User 再次注册，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。根据本发明的另一个实施例，注册过程 900 为单个用户 ID 提供多个用户身份。从而，在以上示例中，信任引擎 110 将会有利地将 Joe User 的两个身份相关联。正如本领域的技术人员将从此处的公开文本中所理解的那样，用户可能具有许多身份，例如一家之主 Joe User、慈善基金会的成员 Joe User 等等。即使用户可能具有多个身份，根据此实施例，信任引擎 110 优选地也只存储一组注册数据。此外，用户可有利地根据需要添加、编辑 / 更新或删除身份。

[0122] 虽然注册过程 900 是参考其优选实施例来公开的，但是本发明不想要限于此。相反，本领域的技术人员将会从此处的公开文本中意识到，用于收集注册数据尤其是注册认证数据的多种备选方案。例如，小应用程序可以是基于公共对象模型 (COM) 的小应用程序等等。

[0123] 另一方面，注册过程可包括分级注册。例如，在最低级别的注册，用户可经由通信链路 125 注册而不会产生关于其身份的文档记录。根据更高级别的注册，用户利用诸如数字公证人这样的受信任的第三方来注册。例如，用户可亲自去见受信任的第三方，产生诸如出生证书、驾驶执照、军队 ID 之类的证书，并且受信任的第三方例如可有利地在注册提交中包括其数字签名。受信任的第三方可包括实际公证人、诸如邮局或机动车辆部这样的政府机构、大公司中注册雇员的负责人力资源的人，等等。本领域的技术人员将会从此处的公开文本理解到在注册过程 900 期间可发生多种不同级别的注册。

[0124] 在接收到注册认证数据后，在步骤 915 处，事务引擎 205 利用传统完全 SSL 技术将注册认证数据转发到认证引擎 215。在步骤 920 中，认证引擎 215 利用认证引擎 215 的私钥对注册认证数据进行解密。此外，认证引擎 215 用数据分割模块来对注册认证数据进行数学运算，以便将数据分割成至少两个不可独立解密的随机化数字。如前所述，至少两个数字可包括统计上随机的数字或经二进制异或后的数字。在步骤 925 中，认证引擎 215 将经随机化的数字的每个部分转发到数据存储设施 D1 至 D4 之一。如前所述，认证引擎 215 还可有利地就哪些部分被传送到哪些仓库进行随机化。

[0125] 在注册过程 900 期间，用户常会希望被发布数字证书，以便他或她能接收来自密码系统 100 之外的他人的加密文档。如前所述，证书权力机构 115 一般根据几个传统标准中的一个或多个来发布数字证书。一般而言，数字证书包括所有人都公知的用户或系统的公钥。

[0126] 不论用户在注册时还是在其他时间请求数字证书，请求都经由信任引擎 110 被传送到认证引擎 215。根据一个实施例，请求包括 XML 文档，其具有例如用户的适当名称。根据步骤 935，认证引擎 215 将请求传送到密码引擎 220，以指示大容量存储装置 225 生成密

钥或密钥对。

[0127] 在被请求时，在步骤 935 处，密码引擎 220 生成至少一个密钥。根据一个实施例，密码处理模块 625 生成钥匙对，其中一个钥匙被用作私钥，一个被用作公钥。密码引擎 220 存储私钥，并且根据一个实施例，还存储公钥的复本。在步骤 945 中，密码引擎 220 将对于数字证书的请求发送到事务引擎 205。根据一个实施例，该请求有利地包括标准化的请求，例如 PKCS10，嵌入在例如 XML 文档中。对于数字证书的请求可有利地对应于一个或多个证书权力机构以及这些证书权力机构所要求的一个或多个标准格式。

[0128] 在步骤 950 中，事务引擎 205 将此请求转发到证书权力机构 115，该证书权力机构 115 在步骤 955 中返回数字证书。所返回的数字证书可有利地采取标准化格式，例如 PKCS7，或者采取证书权力机构 115 中的一个或多个专用格式。在步骤 960 中，数字证书被事务引擎 205 接收到，并且一个复本被转发到用户，一个复本被存储在信任引擎 110 中。信任引擎 110 存储证书的复本，以便信任引擎 110 将不会需要依赖于证书权力机构 115 的可用性。例如，当用户希望发送数字证书时，或第三方请求用户的数字证书时，对于数字证书的请求通常被发送到证书权力机构 115。但是，如果证书权力机构 115 正在进行维护或者遭受了故障或安全性危害，则数字证书可能不可用。

[0129] 在发布密钥之后的任何时刻，密码引擎 220 都可以有利地采用上述数据分割过程 800，以便密钥被分割成无法独立解密的经随机化的数字。与认证数据类似，在步骤 965 处，密码引擎 220 将经随机化的数字传送到数据存储设施 D1 至 D4。

[0130] 本领域的技术人员将会从此处的公开文本中意识到用户可在注册后的任何时刻请求数字证书。此外，系统之间的通信可以有利地包括完全 SSL 或公钥加密技术。此外，注册过程可发布来自多个证书权力机构的多个数字证书，所述证书权力机构包括信任引擎 110 内部或外部的一个或多个私有证书权力机构。

[0131] 正如步骤 935 至 960 中所公开的，本发明的一个实施例包括对最终被存储在信任引擎 110 上的证书的请求。因为根据一个实施例，密码处理模块 625 发布由信任引擎 110 所使用的钥匙，所以每个证书对应于一个私钥。因此，信任引擎 110 可有利地通过监控用户所拥有的证书或与用户相关联的证书来提供互相协作性。例如，当密码引擎 220 接收到对密码功能的请求时，密码处理模块 625 可调查由发出请求的用户所拥有的证书，以确定该用户是否拥有与请求的属性相匹配的私钥。当存在这种证书时，密码处理模块 625 可使用该证书或与其相关联的公钥或私钥，以执行被请求的功能。当不存在这种证书时，密码处理模块 625 可有利地并且透明地执行多个动作，以尝试补救正确钥匙的缺乏。例如，图 9B 示出协同工作过程 970 的流程图，根据本发明的一个实施例的某些方面，该过程公开了前述步骤，以确保密码处理模块 625 利用适当的钥匙执行密码功能。

[0132] 如图 9B 所示，协同工作过程 970 开始于步骤 972 处，在这里密码处理模块 625 确定所需的证书类型。根据本发明的一个实施例，证书类型可以有利地在对于密码功能的请求中指定，或者在由请求者提供的其他数据中指定。根据另一个实施例，证书类型可以通过请求的数据格式来确定。例如，密码处理模块 625 可有利地识别与特定类型相对应的请求。

[0133] 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 等等。此外，证书类型可包括一个或多个钥匙类型，例如对称钥匙、公钥、诸如 256 比特钥匙这样的强加密钥匙、不那么安全的钥匙等等。此外，证书类型可包括更新或替换一个或多个前述

算法标准或钥匙、一个或多个消息或数据格式、一个或多个数据封装或编码方案，例如 Base 32 或 Base 64。证书类型还可包括与一个或多个第三方密码应用或接口、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。本领域的技术人员将会从此处的公开文本中意识到在证书类型中可存在其他差异，并且可以根据这里所公开的实现去往和来自这些差异的转换。

[0134] 一旦密码处理模块 625 确定了证书类型，则协同工作过程 970 前进到步骤 974，并确定用户是否拥有与步骤 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 能够例如通过其先前的存储装置访问匹配证书时，密码处理模块 625 知道匹配私钥也被存储在信任引擎 110 内。例如，匹配私钥可被存储在仓库 210 或仓库系统 700 内。密码处理模块 625 可有利地请求匹配私钥被从例如仓库 210 组装，然后在步骤 976 中，使用匹配私钥来执行密码动作或功能。例如，如前所述，密码处理模块 625 可有利地执行散列 (hashing)、散列比较、数据加密或解密、数字签名验证或创建等等。

[0135] 当用户不拥有匹配证书时，协同工作过程 970 前进到步骤 978，在这里密码处理模块 625 确定用户是否拥有交叉证明的证书。根据一个实施例，证书权力机构之间的交叉证明发生在第一证书权力机构确定信任来自第二证书权力机构的证书时。换言之，第一证书权力机构确定来自第二证书权力机构的证书符合某些质量标准，因此，可以被“证明”为等同于第一证书权力机构自己的证书。当证书权力机构发布例如具有信任级别的证书时，交叉证明变得更加复杂。例如，第一证书权力机构通常可基于注册过程的可靠程度为特定证书提供三个信任级别，而第二证书权力机构可提供七个信任级别。交叉证明可有利地跟踪来自第二证书权力机构的哪些级别和哪些证书可代替来自第一证书权力机构的哪些级别和哪些证书。当前述交叉证明是在两个证明权力机构之间正式且公开地完成的时，彼此之间的证书和级别映射常被称为“链接 (chaining)”。

[0136] 根据本发明的另一个实施例，密码处理模块 625 可有利地在证书权力机构所同意的那些之外开发交叉证明。例如，密码处理模块 625 可访问第一证书权力机构的证书操作声明 (CPS)，或者其他公布的策略声明，并且例如利用特定信任级别所要求的认证令牌，将第一证书权力机构的证书与另一个证书权力机构的那些相匹配。

[0137] 当在步骤 978 中，密码处理模块 625 确定用户拥有交叉证明的证书时，协同工作过程 970 前进到步骤 976，并利用交叉证明的公钥、私钥或两者来执行密码动作或功能。或者，当密码处理模块 625 确定用户不拥有交叉证明的证书时，协同工作过程 970 前进到步骤 980，在这里密码处理模块 625 选择发布被请求的证书类型或与其交叉证明的证书的证书权力机构。在步骤 982 中，密码处理模块 625 确定如前所述的用户注册认证数据是否符合选中的证书权力机构的认证要求。例如，如果用户通过回答比如人口调查或其他问题而在网络上注册，则所提供的认证数据比起用户提供生物特征量度数据和出现在诸如公证人这样的第三方面前的用户来说，提供的信任级别较低。根据一个实施例，前述认证要求可以有利地的在选中的认证权力机构的 CPS 中提供。

[0138] 当用户已向信任引擎 110 提供了符合选中的证书权力机构的要求的注册认证数据时，协同工作过程 970 前进到步骤 984，在这里密码处理模块 625 获取来自选中的证书权力机构的证书。根据一个实施例，密码处理模块 625 根据前述注册过程 900 的步骤 945 至 960 来获取证书。例如，密码处理模块 625 可有利地采用来自对于密码引擎 220 已经可用的

一个或多个钥匙对的一个或多个公钥，来请求来自证书权力机构的证书。根据另一个实施例，密码处理模块 625 可有利地生成一个或多个新的钥匙对，并使用与其相对应的公钥来请求来自证书权力机构的证书。

[0139] 根据另一个实施例，信任引擎 110 可有利地包括一个或多个能够发布一种或多种证书类型的证书发布模块。根据此实施例，证书发布模块可提供前述证书。当密码处理模块 625 获取证书时，协同工作过程 970 前进到步骤 976，并且利与所获取的证书相对应的公钥、私钥或两者来执行密码动作或功能。

[0140] 当在步骤 982 中用户未向信任引擎 110 提供符合选中的证书权力机构的注册认证数据时，在步骤 986 中，密码处理模块 625 确定是否存在其他的具有不同认证要求的证书权力机构。例如，密码处理模块 625 可查找具有较低认证要求但仍发布选中的证书或其交叉证明的证书权力机构。

[0141] 当存在前述具有较低要求的证书权力机构时，协同工作过程 970 前进到步骤 980，并选择该证书权力机构。或者，当不存在这种证书权力机构时，在步骤 988 中，信任引擎 110 可请求来自用户的另外的认证令牌。例如，信任引擎 110 可请求例如包括生物特征量度数据的新的注册认证数据。此外，信任引擎 110 可请求用户出现在受信任的第三方之前并提供适当的认证证书，例如带着驾驶执照、社会安全卡、银行卡、出生证书、军队 ID 等出现在公证人之前。当信任引擎 110 接收到更新后的认证数据时，协同工作过程 970 前进到步骤 984，并且获取前述选中的证书。

[0142] 通过前述协同工作过程 970，密码处理模块 625 有利地提供了不同密码系统之间的无缝透明翻译和转换。本领域的技术人员将会从此处的公开文本中意识到前述可协同操作的系统的多个优点和实现方式。例如，前述协同工作过程 970 的步骤 986 可有利地的包括下文中更详细论述的信任仲裁的某些方面，其中证书权力机构在某些特殊情况下可接受较低级别的交叉证明。此外，协同工作过程 970 可包括确保标准证书撤回之间的协同工作性及其使用，例如使用证书撤回列表 (CRL)、在线证书状态协议 (OCSP) 等等。

[0143] 图 10 示出根据本发明的一个实施例的某些方面的认证过程 1000 的数据流。根据一个实施例，认证过程 1000 包括收集来自用户的当前认证数据，并将其与用户的注册认证数据相比较。例如，认证过程 1000 开始于步骤 1005 处，在这里用户希望例如与销售者执行事务。这种事务例如可包括选择购买选项、请求访问销售者系统 120 的受限区域或设备等等。在步骤 1010，销售者向用户提供事务 ID 和认证请求。事务 ID 可有利地包括 192 比特的量，其中有 32 比特的时间戳，接着是 128 比特的随机量，或者“临时量 (nonce)”，接着是 32 比特的销售者特定常数。这种事务 ID 唯一地标识事务，以便模仿的事务可被信任引擎 110 所拒绝。

[0144] 认证请求可有利地包括对于特定事务需要哪个级别的认证。例如，销售者可在发布时指定事务所要求的特定信心级别。如果如下文所论述的，认证无法获得此信心级别，则在用户不进行进一步的认证以提高信心级别或者在销售者和服务器之间的认证没有变化的情况下，事务将不会发生。这些问题在下文中更完整地论述。

[0145] 根据一个实施例，事务 ID 和认证请求可有利地在由销售者侧小应用程序或其他软件程序来生成。此外，事务 ID 和认证数据的传输可包括一个或多个利用传统 SSL 技术，例如 1/2SSL，或换言之销售者方认证的 SSL 所加密的 XML 文档。

[0146] 在用户系统 105 接收到事务 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，其中可能包括当前生物特征量度信息。在步骤 1015 处，用户系统 105 利用认证引擎 215 的公钥至少对当前认证数据“B”和事务 ID 进行加密，并将该数据传送到信任引擎 110。传输有利地包括至少利用传统 1/2SSL 技术进行加密的 XML 文档。在步骤 1020 中，事务引擎 205 接收到传输，优选地识别出 URL 或 URI 中的数据格式或请求，并且将传输转发到认证引擎 215。

[0147] 在步骤 1015 和 1020 期间，销售者系统 120 在步骤 1025 处利用优选的完全 SSL 技术将事务 ID 和认证请求转发到信任引擎 110。此通信还可包括销售者 ID，虽然销售者标识也可通过事务 ID 的非随机部分来传输。在步骤 1030 和 1035 处，事务引擎 205 接收到通信、在审计追踪中创建记录，并生成对于从数据存储设施 D1 至 D4 组装的用户注册认证数据的请求。在步骤 1040 处，仓库系统 700 将注册认证数据与用户相对应的部分传送到认证引擎 215。在步骤 1045 处，认证引擎 215 利用其私钥对传输进行解码，并将注册认证数据与用户提供的当前认证数据相比较。

[0148] 步骤 1045 的比较可有利地应用前面提到过的并且将在下文中更详细论述的探试式上下文敏感的认证。例如，如果所接收到的生物特征量度信息没有完美匹配，则产生较低的信心匹配。在特定实施例中，认证的信心级别与事务性质以及用户和销售者的愿望相平衡。这一点也在下文中更详细的描述。

[0149] 在步骤 1050 处，认证引擎 215 利用步骤 1045 的比较结果填充认证请求。根据本发明的一个实施例，认证请求被填充为认证过程 1000 的“是 / 否”或“真 / 假”结果。在步骤 1055 中，填充后的认证请求被返回销售者，以便销售者按照该请求进行操作，从而例如允许用户完成发起认证请求的事务。根据一个实施例，确认消息被传递到用户。

[0150] 基于前述，认证过程 1000 有利地保持敏感数据安全，并产生被配置为保持敏感数据的完好性的结果。例如，敏感数据只在认证引擎 215 内部被组装。例如，注册认证数据是不可解密的，直到它在认证引擎 215 中被数据组装模块组装为止，当前认证数据是不可解密的，直到它被传统 SSL 技术和认证引擎 215 的私钥解开为止。此外，发送到销售者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认证数据。

[0151] 虽然认证过程 1000 是参考其优选和备选实施例来公开的，但本发明不想要限于此。相反，本领域的技术人员将会从此处的公开文本中意识到认证过程 1000 的多种备选方案。例如，销售者可有利地由几乎任何发出请求的应用，甚至是那些和用户系统 105 在一起的应用所取代。例如，客户端应用，例如 Microsoft Word，在解除文档的锁定之前，可使用应用程序接口 (API) 或密码 API(CAPI) 来请求认证。或者，邮件服务器、网络、蜂窝电话、个人或移动计算设备、工作站等等都可做出可由认证过程 1000 填充的认证请求。实际上，在提供了前述的受信任的认证过程 1000 之后，发出请求的应用或设备可提供对多种电子或计算机设备或系统的访问权限或使用。

[0152] 此外，在认证失败的情况下，认证过程 1000 可采用多种备选程序。例如，认证失败可保持相同的事务 ID，并要求用户重新输入其当前认证数据。如前所述，使用相同的事务 ID 允许了认证引擎 215 的比较器监控和限制对于特定事务的认证尝试的数目，从而产生更安全的密码系统 100。

[0153] 此外，认证过程 1000 可有利地被用来开发优雅的单次登记解决方案，例如解除敏

感数据存储库(sensitive data vault)的锁定。例如,成功或肯定的认证可向被认证的用户提供自动访问几乎无限数目的系统和应用的任何数目口令的能力。例如,用户的认证可向用户提供与多个在线销售者、局域网、各种个人计算设备、因特网服务提供者、拍卖提供者、投资经纪人等等相关联的口令、登录、金融证书等的访问权限。通过采用敏感数据存储库,用户可选择真正大且随机的口令,因为它们不再需要通过关联来记忆这些口令。相反,认证过程 1000 提供对其的访问。例如,用户可选择二十多位长的随机字母数字串,而不是与可记忆的数据、名称等相关联的东西。

[0154] 根据一个实施例,与特定用户相关联的敏感数据存储库可有利地被存储在仓库 210 的数据存储设施中,或者被分割并存储在仓库系统 700 中。根据此实施例,在肯定用户认证之后,信任引擎 110 向发出请求的应用提供被请求的敏感数据,例如适当的口令。根据另一个实施例,信任引擎 110 可包括用于存储敏感数据存储库的单独系统。例如,信任引擎 110 可包括独立的软件引擎,该软件引擎实现数据存储库功能,并且象征性地位于前述信任引擎 110 的前端安全措施系统的“后面”。根据此实施例,软件引擎在软件在接收到来自信任引擎 110 的指示肯定用户认证的信号之后提供被请求的敏感数据。

[0155] 在另一个实施例中,数据存储库可由第三方系统实现。与软件引擎实施例类似,第三方系统可有利地在接收到来自信任引擎 110 的指示肯定用户认证的信号之后提供被请求的敏感数据。根据另一个实施例,数据存储库可在用户系统 105 上实现。用户方软件引擎可有利地在接收到来自信任引擎 110 的指示肯定用户认证的信号之后提供前述数据。

[0156] 虽然前述数据存储库是参考备选实施例来公开的,但是本领域的技术人员将会从此处的公开文本中意识到它的多种其他实现方式。例如,特定数据存储库可包括来自前述实施例的某些或全部的方面。此外,前述数据存储库中的任何一个可在不同时间采用一个或多个认证请求。例如,数据存储库中的任何一个可以按以下方式来要求认证:每隔一个或多个事务、周期性地、每隔一个或多个会话、在每次访问一个或多个网页或网站时、以一个或多个其他指定间隔等等。

[0157] 图 11 示出根据本发明的一个实施例的某些方面的签署过程 1100 的数据流。如图 11 所示,签署过程 1100 包括与上文中参考图 10 所描述的认证过程 1000 的步骤类似的步骤。根据本发明的一个实施例,签署过程 1100 首先认证用户,然后如下文中更详细论述的,执行几个数字签署功能中的一个或多个。根据另一个实施例,签署过程 1100 可有利地存储与其相关的数据,例如消息或文档的散列等等。此数据可有利地被用于审计或任何其他事件中,例如当参与方尝试拒绝事务时。

[0158] 如图 11 所述,在认证步骤期间,用户和销售者可有利地就诸如合约这样的消息达成协议。在签署期间,签署过程 1100 有利地确保用户所签署的合约与销售者提供的合约相同。因此,根据一个实施例,在认证期间,销售者和用户在发送到认证引擎 215 的数据中包括其各自的消息或合约的复本的散列。通过只采用消息或合约的散列,信任引擎 110 可有利地存储少得多的数据,从而提供了有高效和节约成本的密码系统。此外,所存储的散列可有利地被与所考虑的文档的散列相比较,以确定所考虑的文档是否匹配任何一方所签署的那一个。确定文档是否与和事务相关的那个文档相同的能力提供了可以用于反对某一方拒绝事务的主张的附加证据。

[0159] 在步骤 1103 中,认证引擎 215 组装注册认证数据并将其与用户所提供的当前认证

数据相比较。当认证引擎 215 的比较器指示注册认证数据与当前认证数据匹配时，认证引擎 215 的比较器还将由销售者提供的消息的散列与用户提供的消息的散列相比较。从而，认证引擎 215 有利地确保了用户所同意的消息与销售者所同意的相同。

[0160] 在步骤 1105 中，认证引擎 215 将数字签名请求发送到密码引擎 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。但是，本领域的技术人员将会从此处的公开文本中意识到密码引擎 220 实际上可接受任何类型的数据，包括但不限于视频、音频、生物特征量度、图像或文本，来形成所需的数字签名。返回步骤 1105，数字签名请求优选地包括通过传统 SSL 技术传输的 XML 文档。

[0161] 在步骤 1110 中，认证引擎 215 将请求发送到数据存储设施 D1 至 D4 中的每一个，以便数据存储设施 D1 至 D4 中的每一个发送其与签署方相对应的各自的一个或多个密钥的部分。根据另一个实施例，密码引擎 220 采用如前所述的协同工作过程 970 的步骤中的某些或全部，以便密码引擎 220 首先确定从仓库 210 或仓库系统 700 请求的用于签署方的一个或多个适当的钥匙，并采用动作来提供适当的匹配钥匙。根据另一个实施例，认证引擎 215 或密码引擎 220 可有利地请求与签署方相关联的并且存储在仓库 210 或仓库系统 700 中的一个或多个钥匙。

[0162] 根据一个实施例，签署方包括用户和销售者之一或两者。在这种情况下，认证引擎 215 有利地请求与用户和 / 或销售者相对应的密钥。根据另一个实施例，签署方包括信任引擎 110。在此实施例中，信任引擎 110 证明认证过程 1000 适当的认证了用户、销售者或两者。因此，认证引擎 215 请求信任引擎 110 的密钥，例如属于密码引擎 220 的钥匙，以便执行数字签名。根据另一个实施例，信任引擎 110 执行数字式的类似公证人功能。在此实施例中，签署方包括用户、销售者或两者，以及信任引擎 110。从而，信任引擎 110 提供用户和 / 或销售者的数字签名，然后用其自己的数字签名指示用户和 / 或销售者已被适当地认证。在此实施例中，认证引擎 215 可有利地请求对与用户、销售者或两者相对应的密钥的组装。根据另一个实施例，认证引擎 215 可有利地请求与信任引擎 110 相对应的密钥的组装。

[0163] 根据另一个实施例，信任引擎 110 执行类似委托书的功能。例如，信任引擎 110 可代表第三方以数字方式签署消息。在这种情况下，认证引擎 215 请求与第三方相关联的密钥。根据此实施例，签署过程 1100 可有利地包括在允许类似委托书的功能之前认证第三方。此外，认证过程 1000 可包括检查第三方约束，例如规定特定第三方的签名在何时何种情况下被使用的商业逻辑等等。

[0164] 基于前述，在步骤 1110 中，认证引擎请求来自数据存储设施 D1 至 D4 的与签署方相对应的密钥。在步骤 1115 中，数据存储设施 D1 至 D4 将其与签署方相对应的各自的密钥的部分发送到密码引擎 220。根据一个实施例，前述传输包括 SSL 技术。根据另一个实施例，前述传输可有利地被用密码引擎 220 的公钥来超级加密。

[0165] 在步骤 1120 中，密码引擎 220 组装前述的签署方密钥，并利用其加密消息，从而形成数字签名。在签署过程 1100 的步骤 1125 中，密码引擎 220 将数字签名发送到认证引擎 215。在步骤 1130 中，认证引擎 215 将填充后的认证请求以及散列消息的复本和一个（或多个）数字签名发送到事务引擎 205。在步骤 1135 中，事务引擎 205 向销售者发送收据，其中包括事务 ID、关于认证是否成功的指示以及数字签名。根据一个实施例，前述传输可有利地包括信任引擎 110 的数字签名。例如，信任引擎 110 可利用其私钥对收据的散列进行加

密，从而形成将被附加到去往销售者的传输的数字签名。

[0166] 根据一个实施例，事务引擎 205 还向用户发送确认消息。虽然签署过程 1100 是参考其优选和备选实施例来公开的，但是本发明不想要限于此。相反，本领域的技术人员将会从此处的公开文本中意识到签署过程 1100 的多种备选方案。例如，可用诸如电子邮件应用这样的用户应用来替换销售者。例如，用户可能希望利用他或她的数字签名以数字方式签署特定电子邮件。在这种实施例中，在整个签署过程 1100 中的传输可有利地只包括消息散列的一个复本。此外，本领域的技术人员将会从此处的公开文本中意识到多种客户端应用可请求数字签名。例如，客户端应用可包括字处理器、电子数据表、电子邮件、语音邮件、对受限服务区域的访问等等。

[0167] 此外，本领域的技术人员将会从此处的公开文本中认识到签署过程 1100 的步骤 1105 至 1120 可有利地在采用图 9B 的协同工作过程 970 的步骤中的某些或全部，从而提供例如可能需要处理不同签名类型的数字签名的不同密码系统之间的协同工作性。

[0168] 图 12 示出根据本发明的另一个实施例的某些方面的加密 / 解密过程 1200 的数据流。如图 12 所示，解密过程 1200 通过利用认证过程 1000 认证用户而开始。根据一个实施例，认证过程 1000 在认证请求中包括同步会话钥匙。例如，在传统 PKI 技术中，本领域的技术人员理解到利用公钥或私钥对数据进行加密或解密是数学密集型的，并且可能要求大量系统资源。但是，在对称钥匙密码系统中，或者在消息发送者和接收者共享用于加密和解密消息的单个公用钥匙的系统中，数学运算简单得多也快得多。从而，在传统 PKI 技术中，消息的发送者将会生成同步会话钥匙，并利用更简单更快速的对称钥匙系统来加密消息。然后，发送者将会利用接收者的公钥来加密会话钥匙。加密后的会话钥匙将会被附加到同步加密的消息，并且两个数据都被发送到接收者。接收者使用其私钥来对会话钥匙进行解密，然后使用会话钥匙来对消息进行解密。基于前述，更简单和更快速的对称钥匙系统被用于大多数加密 / 解密处理。从而，在解密过程 1200 中，解密有利地假设同步钥匙已经被用户公钥加密。从而，如前所述，加密后的会话钥匙被包括在认证请求中。

[0169] 返回解密过程 1200，当在步骤 1205 中用户已被认证之后，认证引擎 215 将加密后的会话钥匙转发到密码引擎 220。在步骤 1210 中，认证引擎 215 将请求转发到数据存储设施 D1 至 D4 中的每一个，以请求用户的密钥数据。在步骤 1215 中，每个数据存储设施 D1 至 D4 将其各自的密钥部分发送到密码引擎 220。根据一个实施例，前述传输被用密码引擎 220 的公钥来加密。

[0170] 在解密过程 1200 的步骤 1220 中，密码引擎 220 组装密钥并利用该密钥来对会话钥匙进行解密。在步骤 1225 中，密码引擎将会话钥匙转发到认证引擎 215。在步骤 1227 中，认证引擎 215 填充包括解密后的会话钥匙的认证请求，并将填充后的认证请求转发到事务引擎 205。在步骤 1230 中，事务引擎 205 将认证请求以及会话钥匙一起转发到发出请求的应用或销售者。然后，根据一个实施例，发出请求的应用或销售者使用会话钥匙来对加密后的消息进行解密。

[0171] 虽然解密过程 1200 是参考其优选或备选实施例来公开的，但是本领域的技术人员将会从此处的公开文本中意识到解密过程 1200 的多种备选方案。例如，解密过程 1200 可以在同步钥匙加密之前，并且依赖于完全公钥技术。在这种实施例中，发出请求的应用可将整个消息发送到密码引擎 220，或者可以采用某种类型的压缩或可逆散列以将消息发送

到密码引擎 220。本领域的技术人员将会从此处的公开文本中意识到前述通信可有利地包括以 SSL 技术打包的 XML 文档。

[0172] 加密 / 解密过程 1200 还提供对文档或其他数据的加密。从而，在步骤 1235 中，发出请求的应用或销售者可有利地向信任引擎 110 的事务引擎 205 发送对于用户的公钥的请求。发出请求的应用或销售者做出此请求是因为发出请求的应用或销售者使用用户公钥来例如对将用于加密文档或消息的会话钥匙进行加密。正如在注册过程 900 中提到过的，事务引擎 205 例如在大容量存储装置 225 中存储用户的数字证书的复本。从而，在加密过程 1200 的步骤 1240 中，事务引擎 205 请求来自大容量存储装置 225 的用户数字证书。在步骤 1245 中，大容量存储装置 225 将与用户相对应的数字证书发送到事务引擎 205。在步骤 1250 中，事务引擎 205 将数字证书发送到发出请求的应用或销售者。根据一个实施例，加密过程 1200 的加密部分不包括用户认证。这是因为发出请求的销售者只需要用户的公钥，而不请求任何敏感数据。

[0173] 本领域的技术人员将会从此处的公开文本中意识到，如果特定用户不具有数字证书，则信任引擎 110 可采用注册过程 900 中的某些或全部来为该特定用户生成数字证书。然后，信任引擎 110 可发起加密 / 解密过程 1200，从而提供适当的数字证书。此外，本领域的技术人员将会从此处的公开文本中意识到，加密 / 解密过程 1200 的步骤 1220 和 1235 至 1250 可有利地在采用图 9B 的协同工作过程的步骤中的某些或全部，从而提供可能需要处理加密的不同密码系统之间的协同工作性。

[0174] 图 13 示出根据本发明的另一个实施例的某些方面的信任引擎系统 1300 的简化框图。如图 13 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。为了帮助更全面地理解本发明，图 13 将每个信任引擎 1305、1310、1315 和 1320 示为具有事务引擎、仓库和认证引擎。但是，本领域的技术人员将会意识到每个事务引擎可有利地包括参考图 1-8 所公开的元件和通信信道中的某些、组合或全部。例如，一个实施例可有利地包括具有一个或多个事务引擎、仓库、密码服务器或其任何组合的信任引擎。

[0175] 根据本发明的一个实施例，信任引擎 1305、1310、1315 和 1320 中的每一个从地理上而言都是分离的，以使得，例如，信任引擎 1305 可位于第一位置，1310 可位于第二位置，1315 可位于第三位置，信任引擎 1320 可位于第四位置。前述地理分离性有利地减小了系统响应时间，同时增大了整个信任引擎系统 1300 的安全性。

[0176] 例如，当用户登录到密码系统 100 上时，用户可能最靠近第一位置并且可能希望被认证。正如参考图 10 所描述的，为了被认证，用户提供当前认证数据，例如生物特征量度等等，并且当前认证数据被与用户的注册认证数据相比较。因此，根据一个示例，用户有利地向地理上最靠近的 1305 提供当前认证数据。然后信任引擎 1305 的事务引擎 1321 将当前认证数据转发到也位于第一位置处的认证引擎 1322。根据另一个实施例，事务引擎 1321 将当前认证数据转发到信任引擎 1310、1315 或 1320 的认证引擎中的一个或多个。

[0177] 事务引擎 1321 还请求组装例如来自信任引擎 1305 至 1320 中的每一个的仓库的注册认证数据。根据此实施例，每个仓库向信任引擎 1305 的认证引擎 1322 提供它的那部分注册认证数据。然后认证引擎 1322 使用例如来自前两个仓库的加密数据部分来做出响应，并且将注册认证数据组装成可解密的形式。认证引擎 1322 将注册认证数据与当前认证数据相比较，并向信任引擎 1305 的事务引擎 1321 返回认证结果。

[0178] 基于上述，信任引擎系统 1300 使用多个地理上相分离的信任引擎 1305 至 1320 中最靠近的那个来执行认证过程。根据本发明的一个实施例，将信息路由到最靠近的事务引擎可有利地在用户系统 105、销售者系统 120 或证书权力机构 115 中的一个或多个上执行的客户端侧小应用程序上执行。根据备选实施例，可采用更复杂的判决过程，来从信任引擎 1305 至 1320 中进行选择。例如，判决可基于给定信任引擎的可用性、可操作性、连接速度、负载、性能、地理邻近性或其组合。

[0179] 通过这种方式，信任引擎系统 1300 降低了它的响应时间，同时保持了与地理上远程的数据存储设施相关联的安全性优点，所述数据存储设施例如是参考图 7 所论述的那些数据存储设施，其中每个数据存储设施存储敏感数据的随机化部分。例如，信任引擎 1315 的仓库 1325 处的安全性危害不一定会危害到信任引擎系统 1300 的敏感数据。这是因为仓库只包含不可解密的经随机化的数据，这些数据在没有更多的情况下是完全无用的。

[0180] 根据另一个实施例，信任引擎系统 1300 可有利地包括多个与认证引擎类似地安排的密码引擎。密码引擎可有利地执行诸如参考图 1-8 公开的那些功能。根据另一个实施例，信任引擎系统 1300 可有利地用多个密码引擎来替换多个认证引擎，从而执行诸如参考图 1-8 所公开的那些密码功能。根据本发明的另一个实施例，信任引擎系统 1300 可用具有如前所述的认证引擎、密码引擎或二者的功能中的某些或全部的引擎来替换多个认证引擎中的每一个。

[0181] 虽然信任引擎系统 1300 是参考其优选和备选实施例来公开的，但是本领域的技术人员将会意识到信任引擎系统 1300 可包括信任引擎 1305 至 1320 的某些部分。例如，信用引擎系统 1300 可包括一个或多个事务引擎、一个或多个仓库、一个或多个认证引擎或一个或多个密码引擎，或者其组合。

[0182] 图 14 示出根据本发明的另一个实施例的某些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，信任引擎 1405、1410、1415 和 1420 中的每一个包括参考图 1-8 公开的信任引擎 110 的元件中的某些或全部。根据此实施例，当用户系统 105、销售者系统 120 或证书权力机构 115 的客户端侧小应用程序与信任引擎系统 1400 通信时，这些通信被发送到信任引擎 1405 至 1420 中的每一个的 IP 地址。另外，信任引擎 1405、1410、1415 和 1420 中每一个的每个事务引擎的行为类似于参考图 3 公开的信任引擎 1305 的事务引擎 1321。例如，在认证过程期间，信任引擎 1405、1410、1415 和 1420 中每一个的每个事务引擎将当前认证数据发送到它们各自的认证引擎，并发送请求以组装存储在信任引擎 1405 至 1420 中每一个的每个仓库中的经随机化后的数据。图 14 没有示出这些通信的部分，因为这种图示将会变得过度复杂。继续所述认证过程，然后每个仓库将它的那部分经随机化的数据传输到信任引擎 1405 至 1420 中的每一个的每个认证引擎。每个信任引擎的每个认证引擎采用其比较器来确定当前认证数据是否匹配信任引擎 1405 至 1420 中的每一个的仓库提供的注册认证数据。根据此实施例，然后每个认证引擎进行的比较的结果被发送到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发送到信任引擎 1410、1415 和 1420 的冗余模块。从而，信任引擎 1405 的冗余模块类似地接收来自信任引擎 1410、1415 和 1420 的认证引擎的结果。

[0183] 图 15 示出图 14 的冗余模块的框图。冗余模块包括比较器，该比较器被配置为用

于接收来自三个认证引擎的认证结果并将该结果发送到第四个信任引擎的事务引擎。比较器比较来自三个认证引擎的认证结果，并且如果两个结果吻合，则比较器得出以下结论：认证结果应当与两个吻合的认证引擎的结果相匹配。然后此结果被发送回不与所述三个认证引擎相关联的信任引擎相对应的事务引擎。

[0184] 基于前述，冗余模块确定来自优选地与该冗余模块的信任引擎从地理上而言远程的认证引擎接收到的数据中的认证结果。通过提供这种冗余功能。信任引擎系统 1400 确保了对信任引擎 1405 至 1420 之一的认证引擎的危害不足以危害到该特定信任引擎的冗余模块的认证结果。本领域的技术人员将会意识到信任引擎系统 1400 的冗余模块功能还可应用于信任引擎 1405 至 1420 中的每一个的密码引擎。但是，图 14 中未示出这种密码引擎通信以避免复杂。此外，本领域的技术人员将会意识到，用于图 15 的比较器的多种备选的认证结果冲突分辨算法适用于本发明。

[0185] 根据本发明的另一个实施例，信任引擎系统 1400 可有利地在密码比较步骤期间采用冗余模块。例如，前述参考图 14 的关于冗余模块的公开文本中的某些或全部可有利地实现在特定事务期间由一方或多方提供的文档的散列比较期间。

[0186] 虽然已就某些优选和备选实施例来描述了本发明，但是从此处的公开文本中本领域的技术人员将会明显看出其他实施例。例如，信任引擎 110 可发布短期证书，其中在预定的时间段中私用密钥被发表给用户。例如，当前证书标准包括可被设置为在预定量的时间之后期满的有效性字段。从而，信任引擎 110 可向用户发表私钥，其中私钥可能例如在 24 小时中有效。根据这种实施例，信任引擎 110 可有利地发布与特定用户相关联的新的密钥对，然后发表该新的密钥对的私钥。然后，一旦私用密钥被发表，信任引擎 110 立即使任何对这种私钥的内部有效使用期满，因为这种使用不再能被信任引擎 110 保护。

[0187] 此外，本领域的技术人员将会意识到密码系统 100 或信任引擎 110 可包括识别任何类型的设备的能力，所述设备例如是但不限于笔记本电脑、蜂窝电话、网络、生物特征量度设备等等。根据一个实施例，这种识别可能来自对于特定服务的请求中提供的数据，所述请求例如是对于导致访问或使用的认证的请求、对于密码功能的请求等等。根据一个实施例，前述请求可包括唯一设备标识符，例如处理器 ID。或者，该请求可包括采取特定的可识别数据格式的数据。例如，移动和卫星电话通常不包括用于安全的 X509.v3 重型加密证书的处理能力，因此不会请求这些证书。根据此实施例，信任引擎 110 可识别给出的数据格式的类似，并且仅对适当的种类做出响应。

[0188] 在以上描述的系统的另一个方面中，可利用以下将要描述的各种技术来提供对上下文敏感的认证。上下文敏感认证，例如如图 16 所示的上下文敏感认证提供了以下可能性：不仅评估用户在尝试认证其自身时发送的实际数据，而且还评估围绕该数据的生成和递送的环境的可能性。这种技术还可支持用户和信任引擎 110 或销售者和信任引擎 110 之间的事务特定的信任仲裁，如下所述。

[0189] 如上所述，认证是证实用户是它所声称的那个人的过程。一般而言，认证要求向认证权力机构展示某些事实。本发明的信任引擎 110 代表了用户必须向其认证自身的权力机构。用户必须通过以下方式之一来向信任引擎 110 展示他就是他所声称他是的那个人：知道只有该用户应当知道的某个事情（基于知识的认证）、拥有只有该用户应当拥有的某个事物（基于令牌的认证）或者通过成为只有该用户应当成为的某个事物（基于生物特征量

度的认证）。

[0190] 基于知识的认证的示例包括但不限于口令、PIN 号码或锁定组合。基于令牌的认证的示例包括但不限于住宅钥匙、物理信用卡、驾驶执照或特定电话号码。基于生物特征量度的认证的示例包括但不限于指纹、笔迹分析、面部扫描、手部扫描、耳部扫描、虹膜扫描、血管样式、DNA、嗓音分析或视网膜扫描。

[0191] 每种类型的认证具有特定的优点和缺点，并且每一种提供不同级别的安全性。例如，比起偷听某人的口令并重复该口令来说，创建匹配他人的指纹的虚假指纹一般更难。每种类型的认证还要求认证权力机构已知不同类型的数据，以便利用该种形式的认证来验证某人。

[0192] 这里所使用的“认证”是广泛地指验证某个人的身份是他所声称他是的那个人的整个过程。“认证技术”是指基于特定信息片段、物理信牌或生物特征量度读取的特定类型的认证。“认证数据”是指被发送或以其他方式展示给认证权力机构以便确立身份的信息。“注册数据”是指最初被提交给认证权力机构以便确立与认证数据相比较的基线的数据。“认证实例”是指与通过认证技术进行的认证尝试相关联的数据。

[0193] 认证用户的过程中涉及的内部协议和通信是参考以上图 10 来描述的。此过程中发生上下文敏感认证的部分发生在图 10 的步骤 1045 所示的比较步骤内。此步骤发生在认证引擎 215 内，并且涉及组装从仓库 210 取回的认证数据 410 并将用户提供的认证数据与其相比较。此过程的一个特定实施例在图 16 中示出并在下文中描述。

[0194] 用户提供的当前认证数据和从仓库 210 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包括与认证的分离技术相关的数据。在步骤 1605 中，认证引擎 215 分离与每个认证实例相关联的认证数据。这一步是必要的，以便认证数据被与用户的适当的注册数据子集相比较（例如指纹认证数据应当被与指纹注册数据相比较，而不是与口令注册数据相比较）。

[0195] 一般而言，认证一个用户涉及一个或多个个体的认证实例，这取决于用户可获得的认证技术。这些方法由用户在注册过程期间提供的注册数据（如果用户的注册时未提供视网膜扫描，则他将不能用视网膜扫描来认证他自己）以及用户当前可获得的装置（例如，如果用户在其当前位置处不具有指纹读取器，则指纹认证就是不可行的）所限制。在某些情况下，单个认证实例可能就足以认证用户；但是，在某些情况下，可使用多个认证实例的组合以便为特定事务有信心地认证用户。

[0196] 每个认证实例由与特定认证技术（例如指纹、口令、智能卡等）相关的数据以及围绕用于该特定技术的数据捕捉和递送的环境构成。例如，尝试经由口令进行认证的特定实例不仅会生成与口令本身相关的数据，还会生成与口令尝试相关的环境数据，称为“元数据(metadata)”。此环境数据包括诸如以下信息：特定认证实例发生的时间、认证信息被递送自的网络地址以及本领域的技术人员已知的可确定关于认证数据的起源的任何其他信息（例如连接类型、处理器序列号等等）。

[0197] 在许多情况下，只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翻译或掩蔽起源计算机的地址的其他技术的网络之上，则只能确定代理或路由器的地址。类似地，在许多情况下，诸如处理器序列号这样的信息可能由于以下原因中的任何一种而不可用：所使用的硬件或操作系统的限制、系统的操作者对这种功能的禁用、或者用户的

系统和信任引擎 110 之间的连接的其他限制。

[0198] 如图 16 所示，一旦认证数据内代表的个体认证实例在步骤 1605 中被提取和分离，则认证引擎 215 评估每个实例在指示用户是他所声称的那个人方面的可靠度。单个认证实例的可靠度一般是基于几个因素来确定的。这些因素可以被分组为：在步骤 1610 中评估的涉及与技术相关的可靠度的因素，以及在步骤 1815 中评估的涉及所提供的特定认证数据的可靠度的因素。第一组包括但不限于所使用的认证技术的固有可靠度以及用于该方法的注册数据的可靠度。第二组包括但不限于注册数据和与认证实例一起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都可以独立于其他的因素而变化。

[0199] 认证技术的固有可靠度是基于冒名顶替者提供他人的正确数据的困难度以及该认证技术的整体差错率的。对于基于口令和知识的认证方法，此可靠度通常是相当低的，因为没有什么能够防止某人将其口令暴露给另一个人以及防止所述第二人使用该口令。即使更复杂的基于知识的系统也只会具有中等可靠度，这是因为知识很容易被从一个从传送到另一个人。基于令牌的认证，例如具有适当的智能卡或使用特定终端来执行认证的认证也类似地具有它自己所使用的低可靠度，这是因为无法保证正确的人拥有适当的令牌。

[0200] 但是，生物特征量度技术从本质上来说是更加可靠的，这是因为一般难以向他人提供以便利的方式（哪怕是故意地）使用你的指纹的能力。因为破坏生物特征量度认证技术更困难，所以生物特征量度方法固有可靠度一般高于单独的基于知识或基于令牌的认证技术的可靠度。但是，即使是生物特征量度技术也可能有生成虚假接受或虚拟拒绝的情形。这些事件可以由相同生物特征量度技术的不同实现方式的不同可靠度来反映。例如，一个公司提供的指纹匹配系统可能提供比另一个公司提供的指纹匹配系统更高的可靠度，这是因为其中一个使用较高质量的光学或更好的扫描分辨率，或减少错误接受或错误拒绝的发生的某个其他改进。

[0201] 注意此可靠度可以以不同方式来表达。希望可靠度被表达成探试 530 和认证引擎 215 的算法能够用于计算每个认证的信心级别的某种量度。表达这些可靠度的一种优选模式是表达成百分数或分数。例如，指纹可能被分配 97％的固有可靠度，而口令可被分配 50％的固有可靠度。本领域的技术人员将会意识到这些特定值只是示例性的，对于特定实现方式可以发生变化。

[0202] 必须为可靠度评定的第二个因素是注册的可靠度。这是上文中提到的“分级注册”过程的一部分。此可靠度因素影响初始注册过程期间提供的标识的可靠度。例如，如果个人最初以向公证人或其他公共官方机构物理地提供证据的方式进行注册，并且注册数据在此时被记录并公证，则这种数据比起在注册时经由网络提供并且仅由并非真正与个人联系在一起的数字签名或其他信息来担保的数据来要更可靠。

[0203] 其他的具有不同的可靠度级别的注册技术包括但不限于：在信任引擎 110 操作者的物理办事处注册；在用户工作地点注册；在邮局或护照局注册；通过信任引擎 110 操作者的附属或受信任方注册；不具名或匿名注册，其中被注册的身份尚未被用特定真实个体标识；以及现有技术中已知的其他这种手段。

[0204] 这些因素反映信任引擎 110 和注册过程期间提供的标识的源之间的信任。例如，如果注册是在初始的提供身份证据的过程期间联系雇员来执行的，则当用于公司内时，此

信息可能被视为极可靠的，但是政府机构或竞争者可能就不那么信任它。因此，这些其他组织中的每一个所操作的信任引擎可向此注册分配不同级别的可靠度。

[0205] 类似地，经由网络提交的但是是由相同信任引擎 110 的先前注册提供的其他受信任的数据来认证的附加数据可以被视为与原始注册数据一样可靠，即使后一种数据是经由开放网络提交的。在这种情况下，后续公证将会有效地增大与原始注册数据相关联的可靠度级别。通过这种方式，例如，则可以通过向某些注册官方机构展示与注册的数据相匹配的个人的身份来将不具名或匿名注册提升到完全注册。

[0206] 上述可靠度因素一般是可以在任何特定的认证实例之前确定的值。这是因为它们是基于注册和技术的，而不是基于实际认证的。在一个实施例中，基于这些因素生成可靠度的步骤涉及查找先前为此特定认证技术所确定的值以及用户的注册数据。在本发明的一个有利实施例的另一个方面中，这种可靠度可以与注册数据本身包括在一起。通过这种方式，这些因素与来自仓库 210 的注册数据一起被自动递送到认证引擎 215。

[0207] 虽然这些因素一般可以在任何个体的认证实例之前被确定，但是它们仍然对于为该用户使用该特定认证技术的每个认证实例有影响。此外，虽然值可能随着时间变化（例如如果用户以更可靠的方式重新注册），但是它们不依赖于认证数据本身。相反，与单个特定实例的数据相关联的可靠度因素在每个场合下可能是不同的。如下所述，必须为每个新的认证评估这些因素以便在步骤 1815 中生成可靠度得分。

[0208] 认证数据的可靠度反映了在特定认证实例中用户提供的数据与认证注册期间提供的数据之间的匹配。这是认证数据是否与用户声称他是的那个个体的注册数据相匹配的基本问题。通常，当数据不匹配时，用户被认为是未被成功认证，并且认证失败。评估这一点的方式可根据所使用的认证技术而变化。对这种数据的比较是由图 5 所示的认证引擎 215 的比较器 515 功能来执行的。

[0209] 例如，口令的匹配通常是以二元方式来评估的。换言之，口令或者是完全匹配，或者是失败匹配。如果口令不是完全正确的话，则通常不希望接受与正确口令接近的口令，来作为哪怕是部分匹配。因此，当评估口令认证时，比较器 515 所返回的认证的可靠度通常或者是 100％（正确）或者是 0％（错误），而不可能有中间值。

[0210] 与用于口令的规则类似的规则通常被应用到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拥有与正确智能卡相类似的智能卡和拥有任何其他不正确的令牌的错误程度是一样的。因此令牌往往也是二元认证：用户或者拥有正确令牌，或者没有。

[0211] 但是，某些类型的认证数据，例如问卷和生物特征量度，则通常不是二元认证。例如，指纹可能在不同程度上与参考指纹相匹配。从某种程度上而言，这可能是因为初始注册期间或者后续认证时捕捉的数据的质量的变化。（指纹可能被弄污，或者某个人的特定的手指上可能有仍在愈合中的疤痕或烧伤）。在其他情况下，数据可能不是那么完全地匹配，这是因为信息本身在某种程度上可变的并且是基于样式匹配的。（由于背景噪声或记录嗓音的环境的声学特性或者因为那个人感冒了，嗓音分析可能看起来接近但不是十分正确）。最后，在要比较大量数据的情况下，可能就有这种情况：许多数据匹配得很好，但是某些却不是。（十个问题的问卷可能产生八个对于个人问题的正确回答，但有两个不正确的回答）。由于这些原因中的任何一种，注册数据和特定认证实例的数据之间的匹配可能被比较

器 515 符合需要地分配部分匹配值。通过这种方式，例如，指纹可能被说成是 85％匹配，声纹是 65％匹配，问卷是 80％匹配。

[0212] 由比较器 515 产生的这种度量（匹配度）是代表认证正确与否的基本问题的因素。但是，如上所述，这只是可用于确定给定认证实例的可靠度的因素之一。还注意到即使可确定到某个部分程度的匹配，但是最终可能还是需要基于部分匹配提供二元结果。在备选操作模式中，也可以基于匹配度是否超过特定阈值匹配级别，来将部分匹配视为二元的，即或者是完全（100％）或者是失败（0％）匹配。这种过程可用于为（不然会产生部分匹配的）系统提供简单的通过 / 失败匹配级别。

[0213] 在评估给定认证实例的可靠度时考虑的另一个因素涉及提供此特定实例的认证数据的环境。如上所述，环境是指与特定认证实例相关联的元数据。这可包括但不限于诸如以下信息：认证者的网络地址（到可确定的程度为止）；认证时间；认证数据的传输模式（电话线路、蜂窝、网络等等）；以及认证者的系统的序列号。

[0214] 这些因素可用于产生用户通常请求的认证类型的简档。然后，此信息可用于以至少两种方式评价可靠度。一种方式是考虑用户是否正在以与此用户进行的认证的正常简档相一致的方式请求认证。如果用户在工作日期间（当其在工作时）通常从一个网络地址做出认证请求，而在晚间或周末（当其在家时）通常是从另一个网络地址做出认证请求，则在工作日期间从家庭地址发生的认证就不太可靠，这是因为它在正确认证简档的范围之外。类似地，如果用户通常使用指纹生物特征量度以及在夜时进行认证，则在日间仅用口令发起的认证就不那么可靠。

[0215] 环境元数据可用于评估认证实例的可靠度的另一种方式是确定环境对于认证者是他所声称的那个个体提供了多少确证。例如，如果认证来自具有已知与用户相关联的序列号的系统，则这是用户是他所声称的那个人的良好的环境指示物。相反，当已知用户位于伦敦时，如果认证来自已知处于洛杉矶的网络地址，则这就是对于此认证根据其环境不那么可靠的指示。

[0216] 还可能当用户与销售者系统或信任引擎 110 交互时，cookie 或其他电子数据被放置在用户所使用的系统之上。此数据被写入到用户系统的存储装置中，并且可包括可被用户系统上的 Web 浏览器或其他软件所读取的标识。如果此数据被允许在会话之间存在于用户系统上（“持续 cookie”），则在特定用户的认证期间，该数据可以与认证数据一起被发送，作为过去对此系统的使用的进一步的证据。从效果上而言，给定实例的元数据，尤其是持续 cookie，本身就可形成一类基于令牌的认证。

[0217] 一旦按照上文中分别在步骤 1610 和 1615 中描述的，适当的基于认证实例的技术和数据的可靠度因素被生成，这些可靠度因素就被用于产生步骤 1620 中提供的认证实例的整体可靠度。完成这一点的一种手段就是将每个可靠度表达为百分数，然后将它们乘在一起。

[0218] 例如，假设认证数据是从根据用户过去的认证简档已完全知道是用户的家用计算机的网络地址发送进来的，并且所使用的技术是指纹识别（97％），并且初始指纹数据已通过用户的雇主向信任引擎 110 登记（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％）。则此认证实例的整体可靠度将会被计算为这些可靠度之积：100％ * 97％ * 90％ * 99％ -86.4％的可靠度。

[0219] 这个计算出的可靠度代表单个认证实例的可靠度。单个认证实例的整体可靠度也可以用以不同方式对待不同可靠度因素的技术来计算，所述不同方式利用是通过使用其中不同权重被分配给每个可靠度因素的公式。此外，本领域的技术人员将会意识到所使用的实际值可代表除百分数之外的值，并且可使用非算术系统。一个实施例可包括被认证请求者用来设置每个因素的权重以及用于确立认证实例的整体可靠度的算法的模块。

[0220] 认证引擎 215 可使用上述技术及其变体来确定单个认证实例的可靠度，如步骤 1620 所示。但是，在许多认证情形中，同时提供多个认证实例可能是有用的。例如，在尝试使用本发明的系统来认证他自己的同时，用户可提供用户标识、指纹认证数据、智能卡和口令。在这种情况下，三个独立的认证实例被提供给信任引擎 110 以便评估。前进到步骤 1625，如果认证引擎 215 确定用户提供的数据包括多于一个认证实例，则每个实例又将会以步骤 1630 中所示的方式被选择，并且以上文中步骤 1610、1615 和 1620 中所描述的方式被评估。

[0221] 注意，所论述的许多可靠度因素在不同实例间可能是不同的。例如，这些技术的固有可靠度可能是不同的，认证数据和注册数据之间提供的匹配度也可能是不同的。此外，对于这些技术中的每一个，用户可能在不同时间和不同环境下提供了注册数据，从而也为这些实例中的每一个提供了不同的注册可靠度。最后，即使提交这些实例中的每一个的数据的环境是相同的，这些技术的使用也可能各自不同地与用户的简档相适配，因此可能被分配不同的环境可靠度。（例如，用户可能通常使用其口令和指纹，而不使用其智能卡）。

[0222] 这样，这些认证实例中的每一个的最终可靠度可能彼此不同。但是，通过一起使用多个实例，对于认证的整体信心级别往往会增大。

[0223] 一旦认证引擎已为认证数据中提供的所有认证实例执行了步骤 1610 至 1620，则在步骤 1635 中每个实例的可靠度被用于评估整体认证信心级别。将个体认证实例可靠度组合成认证信心级别的这一过程可以由与产生的个体可靠度相关的各种方法来模拟，并且也可针对解决这些认证技术中的某一些之间的特定交互。（例如，多个诸如口令这样的基于知识的系统产生的信心可能少于单个口令或者甚至是相当弱的生物特征量度，例如基本嗓音分析）。

[0224] 认证引擎 215 可用于组合多个同时发生的认证实例的可靠度来生成最终信心级别的手段是将每个实例的不可靠度相乘以得到总不可靠度。不可靠度一般是可靠度的补百分比。例如，84％可靠的技术是 16％不可靠的。产生 86％、75％和 72％的可靠度的上述三个认证实例（指纹、智能卡、口令）将会分别具有 (100-86)％、(100-75)％和 (100-72)％，或者说 14％、25％和 28％的相应的不可靠度。通过将这些不可靠度相乘，我们得到累积不可靠度 14％＊ 25％＊ 28％ -0.98％的不可靠度，这对应于 99.02％的可靠度。

[0225] 在另一种操作模式中，在认证引擎 215 内可应用另外的因素和探试 530，以考虑到各种认证技术之间的互相依赖性。例如，如果某人拥有对于特定家用计算机的未经授权的访问权限，则它们可能也拥有对于该地址处的电话线路的访问权限。因此，基于主叫电话号码以及基于认证系统的序列号的认证不会使认证的整体信心增加多少。但是，基于知识的认证很大程度上是独立于基于令牌的认证的（即，如果某人窃取了你的蜂窝电话或钥匙，则与他未曾窃取相比，他也不会更有可能知道你的 PIN 或口令）。

[0226] 此外，不同销售者或其他认证请求者可能希望以不同方式对认证的不同方面进行

加权。这可包括在计算个体实例的可靠度时使用单独的加权因素或算法，以及使用不同的手段来评估具有多个实例的认证事件。

[0227] 例如，某种类型的事务的销售者，例如公司电子邮件系统，可能希望默认地主要基于探试和其他环境数据来进行认证。因此，它们可能向与元数据相关的因素以及与围绕认证事件的环境相关联的其他简档相关信息应用高权重。此配置通过不向用户要求比他在工作期间登录到正确机器上时更多的东西，从而可以用于减除正常操作期间用户的负担。但是，另一个销售者可能为来自特定技术（例如指纹匹配）的认证应用最重的权重，这是因为以下策略判决：对于特定销售者的目的，这种技术最适合于认证。

[0228] 在一种操作模式中，这种变化的权重可由认证请求者在生成认证请求时定义，并且与认证请求一起被发送到信任引擎 110。在另一种操作模式中，这种选项也可在认证请求者的初始注册过程期间被设置为优选，并且被存储在认证引擎内。

[0229] 一旦认证引擎 215 为所提供的认证数据产生了认证信心级别，则在步骤 1640 中，此信心级别被用于完成认证请求，并且此信息被从认证引擎 215 转发到事务引擎 205，以便被包括在去往认证请求者的消息中。

[0230] 上述过程只是示例性的，本领域的技术人员将会意识到可以不以所示的顺序来执行步骤，或者只需要执行某些步骤，或者可能需要步骤的各种组合。此外，如果环境允许的话，则某些步骤，例如所提供的每个认证实例的可靠度的评估，可以彼此并行地执行。

[0231] 在本发明的另一个方面中，提供了一种适应以下状况的方法：上述过程产生的认证信心级别未能符合要求认证的销售者或其他方的所要求的信任级别。在所提供的信心级别和所需要的信任级别之间存在差距的情况下，信任引擎 110 的操作者处于以下位置：将提供备选数据或要求的机会提供给一方或两方，以便闭合此信任差距。此过程在此处将被称为“信任仲裁”。

[0232] 信任仲裁可发生在以上参考图 10 和 11 所描述的密码认证的框架内。如该处所示，销售者或其他方可能请求结合特定事务认证特定用户。在一种情况下，销售者就仅仅请求认证，该认证或者是肯定的，或者是否定的，并且在接收到来自用户的适当数据后，信任引擎 110 将会提供这种二元认证。在这种情况下，为了确保肯定认证所需的信心程度是基于信任引擎 110 内设置的优选项来确定的。

[0233] 但是，也可能销售者请求特定级别的信任以便完成特定事务。此要求的级别可与认证请求包括在一起（例如以 98％的信心认证此用户），或者可由信任引擎 110 基于与事务相关联的其他因素来确定（例如对于此事务适当地认证此用户）。一个这种因素可以是事务的经济价值。对于具有较大经济价值的事务，可能要求较高的信任度。类似地，对于具有高风险度的事务，则可能要求高信任度。相反，对于低风险或低价值的事务，销售者或其他认证请求者可能要求较低的信任级别。

[0234] 信任仲裁过程发生在图 10 的步骤 1050 中信任引擎 110 接收认证数据和图 10 的步骤 1055 中向销售者返回认证结果的步骤之间。在这些步骤之间，导致信任级别的评估并且可能导致信任仲裁的过程以如图 17 所示的方式发生。在执行简单二元认证的情况下，图 17 所示的过程简化为使事务引擎 205 直接将所提供的认证数据与标识的用户注册数据相比较，如上文中参考图 10 所述，并且将任何差异标记为否定认证。

[0235] 如图 17 所示，在步骤 1050 中接收到数据之后的第一步骤是在步骤 1710 中事务引

擎 205 确定此特定事务的肯定认证所要求的信任级别。此步骤可以由几种不同方法之一来执行。所要求的信任级别可以由认证请求者在做出认证请求时向信任引擎 110 指定。认证请求者还可预先设置优选项，该优选项被存储在可由事务引擎 205 访问的仓库 210 或其他存储装置内。然后每当此认证请求者做出认证请求时，此优选项可被读取并使用。优选项也可作为安全措施与特定用户相关联，以使得想要认证该用户则始终要求特定的信任级别，用户优选项被存储在可由事务引擎 205 访问的仓库 210 或其他存储介质中。所要求的级别还可由事务引擎 205 或认证引擎 215 基于认证请求中提供的信息来导出，所述信息例如是要认证的事务的价值和风险级别。

[0236] 在一种操作模式中，在生成认证请求时使用的策略管理模块或其他软件被用于指定事务认证的所要求的信任度。这可用于基于策略管理模块内指定的策略来提供分配所要求的信任级别时要遵守的一系列规则。一种有利的操作模式是用于这样一种模块的：该模块将与销售者的 web 服务器相合并，以便适当地确定利用销售者的 web 服务器发起的事务的所要求的信任级别。通过这种方式，来自用户的事务请求可根据销售者的策略被分配所要求的信任级别，并且这种信息可以与认证请求一起被转发到信任引擎 110。

[0237] 这种所要求的信任级别与以下确定度相关：销售者希望拥有的对于认证的个人实际上是他标识他自己是的那个人的确定度。例如，如果由于货物转手因此事务是销售者希望有相当大的确定度的事务，则销售者可能要求 85％的信任级别。对于销售者只不过是认证用户以允许他查看仅限成员的内容或行使聊天室特权的情形，不利风险可能足够小，以至于销售者只要求 60％的信任级别。但是，为了进入价值数万美元的产品合同，销售者可能要求 99％或更大的信任级别。

[0238] 这种所要求的信任级别代表用户为了完成事务必须认证自己到什么程度的量度。例如，如果所要求的信任级别是 85％，则用户必须向信任引擎 110 提供这样的认证，这种认证足以使信任引擎 110 以 85％的信心说该用户是他声称他是的那个人。产生肯定认证（到销售者满意的程度）或信任仲裁概率的是这种所要求的信任级别和认证信心级别之间的平衡。

[0239] 如图 17 所示，在事务引擎 205 接收到所要求的信任级别之后，在步骤 1720 中，它将所要求的信任级别与认证引擎 215 为当前认证计算的认证信心级别（参考图 16 论述）相比较。如果在步骤 1730 中，认证信心级别高于事务的所要求的信任级别，则过程前进到步骤 1740，在这里事务引擎 205 产生此事务的肯定认证。然后对于此效果的消息会被事务引擎 205 插入到认证结果中并被返回给销售者，如步骤 1055 所示（参见图 10）。

[0240] 但是，如果在步骤 1730 中，认证信心级别不满足所要求的信任级别，则对于当前认证存在信心差距，并且在步骤 1750 中进行信任仲裁。信任仲裁在下文中参考图 18 更详细描述。如上所述，此过程在信任引擎 110 的事务引擎 205 内发生。由于为了执行信任仲裁不需要认证或其他密码操作（除了事务引擎 205 和其他组件之间的 SSL 通信所要求的那些以外），因此过程可以在认证引擎 215 之外执行。但是，正如下文中将会论述的，对于认证数据或其他密码或认证事件的任何重新评估都将会要求事务引擎 205 向认证引擎 215 重新提交适当的数据。本领域的技术人员将会意识到信任仲裁过程也可被构造成部分或全部发生在认证引擎 215 本身之内。

[0241] 如上所述，信任仲裁是这样一个过程，在该过程中，信任引擎 110 调解销售者和用

户之间的协商，以尝试在适当时确保肯定认证。如步骤 1805 所示，事务引擎 205 首先确定当前情形是否适合于信任仲裁。这可以基于认证环境来确定，例如此认证是否已经通过几轮仲裁，以及基于销售者或用户的优选项来确定，正如下文中将进一步论述的。

[0242] 在不可能发生仲裁的情况下，过程前进到步骤 1810，在这里事务引擎 205 生成否定认证，然后将其插入到认证结果中，认证结果在步骤 1055 中被发送到销售者（见图 10）。可有利地用于防止认证不确定地待决的一个限制是设置从初始认证请求开始的超时时段。通过这种方式，任何在时间限制内未被肯定认证的事务被拒绝进一步仲裁并被否定认证。本领域的技术人员将会意识到这种时间限制可以根据事务的环境以及用户和销售者的愿望而变化。也可对在提供成功认证时可进行的尝试的次数施加限制。这种限制可由如图 5 所示的尝试限制器 535 来处理。

[0243] 如果在步骤 1805 中未禁止仲裁，则事务引擎 205 随后将会参与与进行事务的一方或两方的协商。如步骤 1820 所示，事务引擎 205 将会向用户发送消息，以请求某种形式的附加认证，以便提升所产生的认证信心级别。就最简单的形式而言，这可以仅仅是指示认证不充分。还可发送以下请求 :请求产生一个或多个附加认证实例以提高认证的整体信心级别。

[0244] 如果在步骤 1825 中用户提供某些附加认证实例，则事务引擎 205 将这些认证实例添加到事务的认证数据中，并且将其转发到认证引擎 215，如步骤 1015 所示（参见图 10），并且基于此事务的预先存在的认证实例以及新提供的认证实例来重新评估认证。

[0245] 可从信任引擎 110 请求另一种类型的认证，以在信任引擎 110 操作者（或受信任的合作者）与用户之间进行某种形式的人对人联系，例如通过电话呼叫进行联系。此电话呼叫或者其他非计算机认证可用于提供与个人的个人联系，以及用于进行某种形式的基于问卷的认证。这一点也提供了以下机会 :验证主叫电话号码并且在用户呼叫进来时对用户进行嗓音分析。即使不能提供附加的认证数据，与用户电话号码相关联的附加上下文也可提高认证上下文的可靠度。基于此电话呼叫的任何修改后的数据或环境被馈送到信任引擎 110 中，以用于考虑认证请求。

[0246] 此外，在步骤 1820 中，信任引擎 110 可向用户提供购买保险的机会，实际上也就是购买更确信的认证的机会。仅在认证的信心级别超过某个特定的开始阈值的情况下，信任引擎 110 的操作者有时才可能希望使这种选项可用。实际上，此用户方保险是在认证符合信任引擎 110 的通常要求的认证信任级别但不符合销售者对于此事务所要求的信任级别时，信任引擎 110 用来担保用户的方式。通过这种方式，用户仍可成功认证到销售者所要求的非常高的级别，即使他只具有产生对于信任引擎 110 来说足够的信心的认证实例。

[0247] 信任引擎 110 的这个功能允许了信任引擎 110 为被认证到信任引擎 110 满意的程度而未认证到销售者满意的程度的某个人进行担保。这与公证人在执行以下操作时执行的功能类似 :将其签名添加到文档，以向稍后阅读该文档者指示其签名出现在文档上的人实际上是签署该文档的人。公证人的签名为用户的签署行为作证。通过相同的方式，信任引擎提供以下指示 :进行事务的那个人就是他声称他是的那个人。

[0248] 但是，由于信任引擎 110 人工地提升用户提供的信心级别，因此信任引擎 110 操作者有更大的风险，因为用户实际上没有符合销售者要求的信任级别。保险的成本被设计为抵销错误肯定认证对于信任引擎 110 的风险（信任引擎 110 可能实际上正在为用户的认证

进行公证）。用户向信任引擎 110 操作者付出报酬，以承担认证到比实际提供的更高的信心级别的风险。

[0249] 由于这种保险系统允许某人从效果上从信任引擎 110 购买更高的信心等级，因此在某些事务中，销售者和用户都可能希望防止使用用户方保险。销售者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要求的信心度的情况，并且因此可能向信任引擎 110 指示用户方保险不被允许。类似地，为了保护其在线身份，用户可能希望防止在其账户上使用用户方保险，或者可能希望将其使用限制到没有保险的认证信心级别高于某个限度的情况。这可以用作安全措施，以防止某人偷听口令或窃取智能卡并使用它们来虚假地认证到低信心级别，然后购买保险来产生非常高的（虚假）信心级别。在确定是否允许用户方保险时可评估这些因素。

[0250] 如果在步骤 1840 中用户购买了保险，则在步骤 1845 中，认证信心级别基于购买的保险而被调整，并且在步骤 1730 中，认证信心级别和所要求的信任级别再次被比较（参见图 17）。过程从该处继续，并且可能通向步骤 1740 中的肯定认证（见图 17），或者返回到步骤 1750 中的信任仲裁过程以便进行进一步仲裁（如果允许的话），或者返回步骤 1810 中的否定认证（如果禁止进一步仲裁的话）。

[0251] 除了在步骤 1820 中向用户发送消息外，在步骤 1830 中，事务引擎 205 也可向销售者发送消息，以指示待决的认证目前处于所要求的信任级别之下。消息还可向销售者提供关于如何继续下去的各种选项。这些选项之一仅仅是通知销售者当前认证信心级别是什么，以及询问销售者是否希望保持其当前的未被满足的所要求的信任级别。这样做可能是有益的，因为在某些情况下，销售者可能具有用于认证事务的独立装置，或者可能使用了默认的要求集合，该默认要求集合一般导致初始指定的所要求的级别，该级别高于手边的特定事务所实际需要的级别。

[0252] 例如，标准做法是预期所有传入的与销售者的购买订单事务都要符合 98％的信任级别。但是，如果某个订单最近已通过销售者和长期顾客之间的电话而被讨论，并且紧接那之后事务就被认证，但是仅被认证到 93％的信心级别，则销售者可能希望简单地降低此事务的接受阈值，因为电话呼叫从效果上而言向销售者提供了附加认证。在某些情况下，销售者可能希望降低其所要求的信任级别，但是不是一直降低到当前认证信心级别。例如，上述示例中的销售者可能考虑订单之前的电话呼叫可能值得所需信任度的 4％的降低；但是这仍大于用户所产生的 93％的信心。

[0253] 如果在步骤 1835 中销售者确实调了其要求的信任级别，则在步骤 1730 中，认证所产生的认证信心级别以及所要求的信任级别被比较（见图 17）。如果现在信心级别超过所要求的信任级别，则在步骤 1740 中，在事务引擎 205 可肯定认证（见图 17）。如果否的话，则如果允许的话可以尝试进一步仲裁，如上所述。

[0254] 除了请求调整所要求的信任级别外，事务引擎 205 还可向请求认证的销售者提供销售者方保险。此保险的用途与以上对于用户方保险所描述的类似。但是，在这里，成本不是对应于信任引擎 110 认证所产生的高于实际认证信心级别时承担的风险，相反，保险成本对应于销售者在认证时接受较低信任级别时所承担的风险。

[0255] 销售者不是仅仅降低其实际要求的信任级别，而是可以选择购买保险，以保护自己以避免承担与认证用户时的较低信任级别相关联的附加风险。如上所述，以下将会是有

利的：销售者仅在现有认证已高于某个阈值时才考虑购买这种保险来覆盖信任差距。

[0256] 这种销售者方保险的可用性允许了销售者选择：直接降低其信任要求，这样他没有附加成本，自己承担虚假认证的风险（基于所要求的较低的信任级别）；或者为认证信心级别和其要求之间的信任差距购买保险，这样信任引擎 110 操作者承担所提供的较低的信心级别的风险。通过购买保险，销售者实际上保持了他的高信任级别要求；因为虚拟认证的风险被转移到了信任引擎 110 操作者。

[0257] 如果在步骤 1840 中销售者购买了保险，则在步骤 1730 中认证信心级别和所要求的信任级别被比较（见图 17），并且过程继续下去如上所述。

[0258] 注意，还可能用户和销售者都对来自信任引擎 110 的消息做出响应。本领域的技术人员将会有多种处理这种情形的方式。一种有利地处理可能的多个响应的模式是就简单地以先到先服务的方式来对待响应。例如，如果销售者以降低的要求信任级别做出响应，并且紧随这之后用户也购买了保险以提升其认证级别，则首先基于来自销售者的降低的信任要求来重新评估认证。如果现在认证是肯定的，则用户的保险购买被忽略。在另一种有利的操作模式中，可以只向用户收取符合销售者的新的降低后的信任要求所需的保险级别的费用（如果即使是在销售者信任要求降低的情况下仍存在信任差距的话）。

[0259] 如果在为认证设置的时间限制内，在步骤 1850 处的信任仲裁过程期间未接收到来自任一方的响应，则在步骤 1805 中仲裁被重新评估。这实际上是再次开始了仲裁过程。如果在步骤 1805 中，时间限制是最终的时间限制，或者其他环境阻止了进一步仲裁，则否定认证在步骤 1810 中被事务引擎 205 生成，并且在步骤 1055 中被返回到销售者（见图 10）。如果否的话，则新的消息可被发送到用户和销售者，并且过程可根据需要被重复。

[0260] 注意对于某些类型的事务，例如以数字方式签署不是事务的一部分的文档，则不一定有销售者或其他第三方；因此事务可能主要是在用户和信任引擎 110 之间。在这种情况下，信任引擎 110 将会拥有其自己的要求信任级别，要生成肯定认证必须满足此级别。但是，在这种情况下，通常不希望信任引擎 110 向用户提供保险以便其能提升其自己的签名的信任度。

[0261] 上文中所描述的并且在图 16-18 中示出的过程可利用上文中参考信任引擎 110 所描述的各种通信模式来实现。例如，消息可以是基于 web 的，并且是用信任引擎 110 和实时下载到运行在用户或销售者系统上的浏览器上的小应用程序之间的 SSL 连接来发送的。在备选操作模式中，用户和销售者可使用某些专用应用，这些专用应用辅助这种仲裁和保险事务。在另一种备选操作模式中，安全电子邮件操作可用于调解上述仲裁，从而允许对认证的推迟评估和批处理。本领域的技术人员将会意识到可使用适合于环境和销售者的认证要求的不同通信模式。

[0262] 以下参考图 19 进行的说明描述了集成了如上所述的本发明的各种方面的示例性事务。此示例示出了用户和销售者之间的由信任引擎 110 所调解的整个过程。虽然上文中详细描述的各种步骤和组件可用于实现以下事务，但是所示出的过程集中于信任引擎 110、用户和销售者之间的交互。

[0263] 在步骤 1900 中，当用户在在线查看网页的同时填写来自销售者的网站的订单表时，事务开始。用户希望把这个用他的数字签名签署的订单表提交给销售者。为了完成这一点，在步骤 1905 中，用户向信任引擎 110 提交订单表以及他对签名的请求。用户还可提

供认证数据，该认证数据如上所述可用于认证他的身份。

[0264] 在步骤 1910 中，如上所述，认证数据被信任引擎 110 与注册数据相比较，并且如果产生肯定认证，则以用户私钥签署的订单表的散列和订单表本身一起被转发到销售者。

[0265] 在步骤 1915 中，销售者接收到被签署的表，然后在步骤 1920 中，销售者将会生成与要进行的购买相关的发票或其他合同。在步骤 1925 中，此合同和对于签名的请求一起被发送回用户。在步骤 1930 中，销售者还向信任引擎 110 发送对于此合同事务的认证请求，其中包括将会被双方签署的合同的散列。为了允许合同被双方以数字形式签署，销售者还包括它自己的认证数据，以便如果必要的话，合同上的销售者签名稍后可被验证。

[0266] 如上所述，信任引擎 110 然后验证销售者提供的认证数据，以确认销售者的身份，并且如果在步骤 1935 中数据产生肯定认证，则继续进行步骤 1955，此时从用户接收到数据。如果销售者的认证数据未与销售者的注册数据匹配到所需程度，则请求进一步认证的消息被返回到销售者。如果必要的话，如上所述，在此处可执行信任仲裁，以便销售者能成功地向信任引擎 110 认证其自己。

[0267] 当在步骤 1940 中用户接收到合同时，他审阅合同，如果合同可接受的话则在步骤 1945 中生成认证数据以签署合同，然后在步骤 1950 中将合同的散列以及他的认证数据发送到信任引擎 110。在步骤 1955 中，信任引擎 110 验证认证数据，并且如果认证良好，则继续进行到处理合同，如上所述。正如以上参考图 17 和 18 所论述的，适当时可执行信任仲裁，以闭合存在于认证信心级别和对于事务所要求的认证级别之间的任何信任差距。

[0268] 在步骤 1960 中，信任引擎 110 利用用户私钥签署合同的散列，并将此签署后的散列发送回销售者，从而代表它自己签署整个消息，即包括以信任引擎 110 的私钥 510 加密的整个消息（包括用户的签名）的散列。在步骤 1965 中，此消息被销售者接收到。消息代表来自信任引擎 110 的已签署的合同（利用用户私钥加密的合同散列）以及收据（消息的散列包括利用信任引擎 110 的私钥加密的已签署的合同）。

[0269] 在步骤 1970 中，信任引擎 110 类似地利用销售者的私钥准备合同的散列，并且将已由信任引擎 110 签署的这个合同散列转发给用户。通过这种方式，在步骤 1975 中，用户也接收到由销售者签署的合同复本，以及由信任引擎 110 签署的关于已签署的合同递送的收据。

[0270] 除了前述以外，本发明的另一个方面提供了密码服务提供者模块 (SPM)，它可被客户端侧应用用作访问上述由信任引擎 110 所提供的功能的装置。一种有利的提供这种密码 SPM 的服务的方式是转接第三方应用编程接口 (API) 和可经由网络或其他远程连接访问的信任引擎 110 之间的通信。示例性的密码 SPM 在下文中参考图 20 描述。

[0271] 例如，在典型系统上，程序员可获得多个 API。每个 API 提供一个功能调用集合，该功能调用可由运行在系统上的应用 2000 进行。提供适用于密码功能、认证功能和其他安全性功能的编程接口的 API 的示例包括由 Microsoft 与其 Windows 操作系统一起提供的密码 API(CAPI)2010，以及由 IBM、Intel 和开放群组的其他成员赞助的公共数据安全体系结构 (CDSA)。在以下论述中，CAPI 将会被用作典型安全性 API。但是，所描述的密码 SPM 可以与 CDSA 或现有技术已知的其他安全性 API 一起使用。

[0272] 当对密码功能进行调用时，此 API 被用户系统 105 或销售者系统 120 所使用。这些功能中可以包括与执行各种密码操作相关联的请求，所述密码操作例如是利用特定钥匙

加密文档、签署文档、请求数字证书、验证已签署的文档上的签名以及这里所描述的或本领域的技术人员已知的其他这种密码功能。

[0273] 这种密码功能通常是在 CAPI 2010 所位于的系统上本地执行的。这是因为一般被调用的功能要求使用本地用户系统 105 的资源，例如指纹读取器，或者使用利用在本地机上执行的库来编程的软件功能。对于这些本地资源的访问通常是由上文提到的一个或多个服务提供者模块 (SPM)2015、2020 来提供的，这些模块提供被用来实现密码功能的资源。这种 SPM 可包括用于执行加密或解密操作的软件库 2015，或者能够访问专用硬件 2025 的驱动器和应用 2020，所述专用硬件 2025 例如是生物特征量度扫描设备。以与 CAPI 2010 提供可由系统 105 的应用 2000 所使用的功能类似的方式，SPM 2015、2020 向 CAPI 提供对较低级别的功能以及与系统上可用的服务相关联的资源的访问。

[0274] 根据本发明，可以提供密码 SPM 2030，该密码 SPM 2030 能够访问信任引擎 110 提供的密码功能，并通过 CAPI 2010 使这些功能可供应用 2000 所用。与 CAPI 2020 只能够通过 SPM 2015、2020 访问本地可用的资源的实施例不同，此处所描述的密码 SPM 2030 将会能够向位于远程的、可经由网络访问的信任引擎 110 提交对于密码操作的请求，以便执行所需操作。

[0275] 例如，如果应用 2000 需要密码操作，例如签署文档，则应用 2000 做出对适当的 CAPI 2010 功能的功能调用。CAPI 2010 依次执行此功能，利用通过 SPM 2015、2020 和密码 SPM 2030 而变得可供它所用的资源。在数字签名功能的情况下，密码 SPM 2030 将会生成适当的请求，该请求将会经由通信链路 125 被发送到信任引擎 110。

[0276] 在密码 SPM 2030 和信任引擎 110 之间发生的操作与任何其他系统和信任引擎 110 之间可能发生的操作相同。但是，这些功能实际上是通过 CAPI 2010 而可供用户系统 105 所用的，以使得它们看起来是在用户系统 105 本身上本地可用的。但是，与普通 SPM 2015、2020 不同，所述功能在远程信任引擎 110 上被实现，并且响应于适当的请求，结果经由通信链路 125 被传递到密码 SPM 2030。

[0277] 此密码 SPM 2030 使得多个操作可供用户系统 105 或销售者系统 120 可用，而这些操作在其他情况下是不可用的。这些功能包括但不限于 ：文档的加密和解密 ；数字证书的发布 ；文档的数字签署 ；数字签名的验证以及对于本领域的技术人员显而易见的其他这种操作。

[0278] 在另一个单独的实施例中，本发明包括用于对任何数据集合执行本发明的数据保护方法的完整系统。此实施例的计算机系统包括数据分割模块，该模块包括图 8 所示的、此处所描述的功能。在本发明的一个实施例中，数据分割模块包括解析器程序或软件套组，该解析器程序或软件套组包括数据分割、加密和解密、重新构成或重新组装功能。此实施例还可包括一个数据存储设施或多个数据存储设施。数据分割模块或解析器包括跨平台软件模块套组，该套组集成在电子基础设施内，或作为任何要求其数据元素的根本安全性的应用的附件。此解析过程对于任何类型的数据集合以及任何和所有文件类型进行操作，或者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。

[0279] 在一个实施例中，本发明的解析过程可以以模块式分层方式来设计，并且任何加密过程都适用于本发明的过程中。本发明的解析过程的模块式层次可以包括但不限于 1) 以密码方式分割，被散布并被安全地存储在多个位置中 ；2) 加密，以密码方式分割，被散布

并被安全存储在多个位置中；2) 加密，以密码方式分割，对每一份加密，然后被散布并且被存储在多个位置中；以及 4) 加密，以密码方式分割，利用与第一步骤中使用的不同类型的加密来对每一份加密，然后被散布并被安全地存储在多个位置中。

[0280] 在一个实施例中，所述过程包括根据生成的随机数或钥匙的内容来分割数据，并且对加密分割数据时使用的钥匙进行同样的密码分割，所述加密根据请求者对隐私和安全性的需要，将要保护的数据分割成两部分或更多部分的解析数据，或者两份或更多份的解析数据，对所有部分加密，然后将这些部分分散并存储回数据库中，或者将它们重新定位到任何指定的固定或可移动设备，在一个实施例中，优选地将要保护的数据分割成四部分或更多部分的解析数据。或者，在另一个实施例中，加密可能发生在分割模块或解析器分割数据集合之前。按此实施例中描述的方式被处理的原始数据被加密和打乱，并受到保护。如果需要的话，加密后的元素的实际上可以被散布到任何位置，包括但不限于单个服务器或数据存储设备，或者散布在独立数据存储设施或设备之中。加密钥匙管理在一个实施例中可以被包括在软件套组中，或者在另一个实施例中可以被集成到现有基础设施或任何其他所需要位置中。

[0281] 以密码方式进行的分割（密码分割）将数据划分成 N 份。划分可以在任何大小的数据单元上进行，所述数据单元包括单个比特、多个比特、字节、千字节、兆字节或更大的单元，以及不论是预定的还是随机生成的数据单元大小的任何样式或组合。基于随机或预定的值集合，单元也可具有不同大小。这意味着数据可以被看成是这些单元的序列。通过这种方式，数据单元本身的大小可以使数据更安全，例如是通过使用一个或多个预定的或随机生成的数据单元大小样式、序列或组合。然后单元被分布（或者是随机的，或者按照预定值集合）到 N 份中。分布还可涉及打乱这些份中的单元的顺序。本领域的普通技术人员易于看出，可以根据多种可能的选择来执行将数据单元分布到多份中，所述选择包括但不限于预定大小，或预定的或随机生成的数据单元大小的一个或多个组合、样式或序列。

[0282] 这个以密码方式进行的分割过程、或者密码分割的一个示例将会是考虑大小为 23 字节的数据，其中数据单元大小被选择为 1 字节，份数被选择为 4。每个字节将会被分布到 4 份中的 1 份。假设是随机分布，则将会获得钥匙以创建 23 个随机数（r1、r2、r3 至 r23）的序列，每个随机数的值在 1 到 4 之间，与四份相对应。数据单元（在此示例中是数据的 23 个字节）中的每一个与对应于四份之一的 23 个随机数之一相关联。将数据的字节分布到四份中将会通过以下方式来发生：、将数据的第一字节放置到第 r1 份中，将第二字节放置到第 r2 份中，将第三字节放置到第 r3 份中，一直到将数字的第 23 个字节放置到第 r23 份中。本领域的普通技术人员易于看出，多种其他可能的步骤或步骤组合或序列，包括数据单元的大小，可被用于本发明的密码分割过程中，并且上述示例是密码分割数据的一个过程的非限制性描述。为了创建原始数据，将会执行逆向操作。

[0283] 在本发明的密码分割过程的另一个实施例中，密码分割过程的一个选项是在份中提供足够的冗余性，以便要想将数据重新组装或恢复到其原始或可用形式则只需要这些份的子集。作为非限制性示例，密码分割可以被完成为“4 中取 3(3of 4)”型密码分割，以使得要将数据重新组装或恢复到其原始或可用形式只需要四份中的三份。这也被称为“N 中取 M 密码分割”，其中 N 是总份数，M 至少比 N 小 1。本领域的普通技术人员易于看出，在本发明的密码过程中，对于产生这种冗余性，有许多可能方式。

[0284] 在本发明的密码分割过程的一个实施例中，每个数据单元被存储在两份中，即主份和备用份中。利用上述“4 中取 3”密码分割过程，可以缺少任何一份，而这对于重新组装或恢复原始数据而不缺少数据单元来说已经足够了，这是因为只需要总共四份中的三份。如此处所述，生成与这些份之一相对应的随机数。基于钥匙使随机数与数据单元相关联并将其存储在相应份中。在此实施例中，使用了一个钥匙来生成主份随机数和备用份随机数。正如这里针对本发明的密码分割过程所描述的，生成与数据单元的数目相等的从 0 到 3 的随机数（也称为主份数字）集合。然后生成与数据单元的数目相等的从 1 到 3 的随机数（也称为备用份数字）的另一个集合。或者，可生成少于数据单元数目的随机数集合，并且重复随机数集合，但是这可能会降低敏感数据的安全性。主份数字被用于确定将数据单元存储到哪一份中。备用份数字与主份数字相结合，以产生 0 到 3 之间的第三份数字，并且此数字被用于确定将数据单元存储到哪一份中。在此示例中，确定第三份数字的方程是：

[0285] （主份数字 + 备用份数字）MOD 4 ＝第三份数字

[0286] 在上述实施例中，主份数字在 0 到 3 之间，备用份数字在 1 到 3 之间，确保了第三份数字与主份数字不同。这使得数据单元被存储到两个不同份中。本领域的普通技术人员易于看出，除了此处公开的实施例外，还有许多其他方法来执行冗余密码分割和非冗余密码分割。例如，可以用不同算法来打乱每一份中的数据单元。这种数据单元打乱例如可以在原始数据被分割成数据单元时执行，或者在数据单元被放置到份中之后执行，或者在份满之后执行。

[0287] 此处描述的各种密码分割过程和数据打乱过程，以及本发明的密码分割和数据打乱方法的所有其他实施例，都可以在任何大小的数据单元上执行，包括但不限于小到单个比特、多个比特、字节、千字节、兆字节或更大。

[0288] 将会执行此处所描述的密码分割过程的源代码的一个实施例的示例是：

[0289] DATA[1:24]- 具有要分割的数据的字节阵列

[0290] SHARES[0:3 ;1:24]-2 维阵列，其中每行代表一份

[0291] RANDOM[1:24]-0..3 范围内的阵列随机数

[0292] S1 ＝ 1 ;

[0293] S2 ＝ 1 ;

[0294] S3 ＝ 1 ;

[0295] S4 ＝ 1 ;

[0296]

```
For J=1 to 24 do
Begin
IF RANDOM[J[==0 then
        Begin
        SHARES[1,S1] = DATA [J];
        S1 = S1 + 1;
        End
ELSE IF RANDOM[J[ ==1 then
        Begin
        SHARES[2,S2] = DATA [J];
        S2 = S2 + 1;
        End
ELSE IF RANDOM[J[ ==2 then
        Begin
        SHARES[3,S3] = DATA [J];
        S3 = S3 + 1;
        End
Else begin
        SHARES[4,S4] = DATA [J];
```

[0297]

```
                S4 = S4 + 1;
                End
            END;
```

[0298] 将会执行此处描述的密码分割 RAID 过程的源代码的一个实施例的示例是：

[0299] 生成两个数字集合，PrimaryShare 为 0 到 3，BackupShare 为 1 到 3。然后利用与上述密码分割相同的过程，将每个数据单元放到 share[primaryshare[1]] 和 share[(primaryshare[1]+backupshare[1])mod 4] 中。此方法可以被缩放到任何大小 N，其中要恢复数据只需要 N-1 份。

[0300] 加密后的数据元素的检索、重新组合、重新组装或重新构成可利用任意多种认证技术，包括但不限于生物特征量度，例如指纹识别、面部扫描、手部扫描、虹膜扫描、视网膜扫描、耳部扫描、血管样式识别或 DNA 分析。本发明的数据分割或解析器模块可根据需要被集成到多种基础设施产品或应用中。

[0301] 本领域中已知的传统加密技术依赖于用于加密数据的一个或多个钥匙并且如果没有钥匙的话则数据不可用。但是，数据保持完整无损，并且易遭受攻击。在一个实施例中，本发明的解析器软件套组通过以下方法来针对解决此问题：将加密后的文件分割或解析成两部分或更多部分或者两份或更多份来针对解决此问题（在另一个实施例中优选为四份或更多份），向每份数据添加另一层加密，然后将这些份存储在不同物理和/或逻辑位置中。当通过使用诸如数据存储设备这样的可移动设备、或通过将数据份置于另一方的控制之下来从系统中物理地移除一个或多个数据份时，则有效地去除了危害受保护数据的任何可能性。

[0302] 本发明的解析器软件套组的一个实施例的示例以及关于如何使用该套组的示例在图 21 中示出，并在下文中描述。但是，本领域的普通技术人员易于看出，除了以下的非限制性示例外，还可用多种其他方式来利用本发明的解析器软件套组。作为一种部署选项，在一个实施例中，可以用外部会话钥匙管理或会话钥匙的安全内部存储来实现解析器。在实现时，将会生成解析器主钥 (Parser Master Key)，该解析器主钥将会被用于保护应用和加密目的。还应当注意，在产生的受保护数据中结合解析器主钥允许了工作组、企业或扩展听众内的个人共享受保护数据的灵活性。

[0303] 如图 21 所示，本发明的此实施例示出了解析器软件套组在数据上执行的以将会话主钥与解析后的数据存储在一起的过程的步骤：

[0304] 1. 生成会话主钥并利用 RS1 流密码来加密数据。

[0305] 2. 根据会话主钥的样式将产生的加密后的数据分割成四份解析数据或四部分解析数据。

[0306] 3. 在本发明的此实施例中，会话主钥将与受保护的数据份一起被存储在数据仓库中。根据解析器主钥的样式分离会话主钥，并将密钥数据附加到经加密的解析数据。

[0307] 4. 产生的四份数据将会包含原始数据的加密部分以及会话主钥的部分。为四个数据份中的每一份生成流密钥。

[0308] 5. 对每份加密，然后将加密钥匙存储在与加密数据部分或数据份不同的位置中：第 1 份获得钥匙 4，第 2 份获得钥匙 1，第 3 份获得钥匙 2，第 4 份获得钥匙 3。

[0309] 要恢复原始数据格式，则逆转以上步骤。

[0310] 本领域的普通技术人员易于看出，此处所描述的方法的某些步骤可以根据需要以不同顺序来执行，或者被重复多次。本领域的普通技术人员还易于看出，可以按不同方式来处理数据的部分。例如，可以仅对解析后的数据的一份执行多个解析步骤。可以以所需要的方式唯一地保护解析后的数据的每个部分，只要数据可以被重新组装、重新构成、重新形成、解密或恢复到其原始形式或其他可用形式即可。

[0311] 如图 22 所示及此处所述，本发明的另一个实施例包括解析器软件套组在数据上执行的以将会话主钥存储在一个或多个单独的钥匙管理表中的过程的步骤：

[0312] 1. 生成会话主钥并利用 RS1 流密码来加密数据。

[0313] 2. 根据会话主钥的样式将产生的加密后的数据分割成四份解析数据或四部分解析数据。

[0314] 3. 在本发明的此方法实施例中，会话主钥将被存储在数据仓库中的单独的钥匙管理表中。为此事务生成唯一的事务 ID。将事务 ID 和会话主钥存储在单独的钥匙管理表中。

根据解析器主钥的样式分离事务 ID，并将数据附加到经加密的解析数据或分离后的数据。

[0315] 4. 产生的四份数据将会包含原始数据的加密部分以及事务 ID 的部分。

[0316] 5. 为四个数据份中的每一份生成流密钥。

[0317] 6. 对每份加密，然后将加密钥匙存储在与加密数据部分或数据份不同的位置中：第 1 份获得钥匙 4，第 2 份获得钥匙 1，第 3 份获得钥匙 2，第 4 份获得钥匙 3。

[0318] 要恢复原始数据格式，则逆转以上步骤。

[0319] 本领域的普通技术人员易于看出，此处所描述的方法的某些步骤可以根据需要以不同顺序来执行，或者被重复多次。本领域的普通技术人员还易于看出，可以按不同方式来处理数据的部分。例如，可以仅对解析后的数据的一份执行多个单独步骤或解析步骤。可以以所需要的方式唯一地保护解析后的数据的每个部分，只要数据可以被重新组装、重新构成、重新形成、解密或恢复到其原始形式或其他可用形式即可。

[0320] 如图 23 所示，本发明的这个实施例示出解析器软件套组在数据上执行的以将会话主钥存储与解析后的数据存储在一起的过程的步骤：

[0321] 1. 访问与已认证的用户相关联的解析器主钥

[0322] 2. 生成唯一会话主钥

[0323] 3. 从解析器主钥和会话主钥的异或函数得出中间钥

[0324] 4. 任选地，利用以中间钥为钥匙的现有或新加密算法来加密数据。

[0325] 5. 根据中间钥的样式，将产生的任选地加密后的数据分离成四份解析数据或四部分解析数据。

[0326] 6. 在本方法的此实施例中，会话主钥将与受保护的数据份一起被存储在数据仓库中。根据解析器主钥的样式分离会话主钥，并将钥匙数据附加到经任选地加密的解析数据份。

[0327] 7. 产生的多份数据将会包含原始数据的经任选加密的部分以及会话主钥的部分。

[0328] 8. 任选地，为四个数据份中的每一份生成加密钥匙。

[0329] 9. 任选地，利用现有的或新的加密算法对每一份加密，然后将加密钥匙存储在与加密数据部分或数据份不同的位置中：例如，第 1 份获得钥匙 4，第 2 份获得钥匙 1，第 3 份获得钥匙 2，第 4 份获得钥匙 3。

[0330] 要恢复原始数据格式，则逆转以上步骤。

[0331] 本领域的普通技术人员易于看出，此处所描述的方法的某些步骤可以根据需要以不同顺序来执行，或者被重复多次。本领域的普通技术人员还易于看出，可以按不同方式来处理数据的部分。例如，可以仅对解析后的数据的一份执行多个单独步骤或解析步骤。可以以所需要的方式唯一地保护解析后的数据的每个部分，只要数据可以被重新组装、重新构成、重新形成、解密或恢复到其原始形式或其他可用形式即可。

[0332] 如图 24 所示及此处所述，本发明的另一个实施例包括解析器软件套组在数据上执行的以将会话主钥存储在一个或多个单独的钥匙管理表中的过程的步骤：

[0333] 1. 访问与已认证的用户相关联的解析器主钥

[0334] 2. 生成唯一会话主钥

[0335] 3. 从解析器主钥和会话主钥的异或函数得出中间钥

[0336] 4. 任选地，利用以中间钥为钥匙的现有或新加密算法来加密数据。

[0337] 5. 根据中间钥的样式，将产生的任选地加密后的数据分离成四份解析数据或四部分解析数据。

[0338] 6. 在本发明的此方法实施例中，会话主钥将会被存储在数据仓库中的单独的钥匙管理表中。为此事务生成唯一的事务 ID。将事务 ID 和会话主钥存储在单独的钥匙管理表中，或者将会话主钥和事务 ID 传递回外部管理的调用程序。根据解析器主钥的样式分离事务 ID，并将数据附加到经任选地加密的解析数据或分离后的数据。

[0339] 7. 产生的多份数据将会包含原始数据的经任选加密的部分以及事务 ID 的部分。

[0340] 8. 任选地，为四个数据份中的每一份生成加密钥匙。

[0341] 9. 任选地，对每一份加密，然后将加密钥匙存储在与加密后的数据部分或数据份不同的位置中。例如，第 1 份获得钥匙 4，第 2 份获得钥匙 1，第 3 份获得钥匙 2，第 4 份获得钥匙 3。

[0342] 要恢复原始数据格式，则逆转以上步骤。

[0343] 本领域的普通技术人员易于看出，此处所描述的方法的某些步骤可以根据需要以不同顺序来执行，或者被重复多次。本领域的普通技术人员还易于看出，可以按不同方式来处理数据的部分。例如，可以仅对解析后的数据的一份执行多个单独步骤或解析步骤。可以以所需要的方式唯一地保护解析后的数据的每个部分，只要数据可以被重新组装、重新构成、重新形成、解密或恢复到其原始形式或其他可用形式即可。

[0344] 正如本领域的技术人员易于看出的，有多种加密方法适合用于本发明的方法中。一次性密码本 (One Time Pad) 算法常被看作是最安全的加密方法之一，并且适合用于本发明的方法中。使用一次性密码本算法要求只要保护数据就生成钥匙。在某些情况下，使用此方法可能不太符合需要，所述情况例如是由于要保护的数据的大小而导致生成和管理非常长的钥匙的情况。在一次性密码本 (OTP) 算法中，使用简单的异或函数 XOR。对于长度相同的二进制流 x 和 y，x XOR y 意味着 x 和 y 的按位异或。

[0345] 在位级别上生成：

[0346] 0XOR 0 = 0

[0347] 0XOR 1 = 1

[0348] 1XOR 0 = 1

[0349] 1XOR 1 = 0

[0350] 此处针对要分割的 n 字节秘密 s（或数据集合）来描述此过程的示例。该过程将会生成 n 字节随机值 a，然后设置：

[0351] b = a XOR s。

[0352] 注意可经由以下方程得出“s”：

[0353] s = a XOR b。

[0354] 值 a 和 b 被称为份或部分，并且被放置在分离的仓库中。一旦将秘密 s 分割成了两份或多份，就以安全的方式将其放置。

[0355] 本发明的解析器软件套组可利用此函数，执行结合多个不同的秘密钥匙值的多个 XOR 函数，所述多个不同密钥值是 K1、K2、K3、Kn、K5。在操作开始时，经由第一加密操作传递要保护的数据，安全数据 = 数据 XOR 秘密钥匙 5：

[0356] S = D XOR K5

[0357] 为了例如以四份 S1、S2、S3、Sn 安全地存储产生的加密后数据，根据 K5 的值，数据被解析成“n”段或“n”份。此操作产生原始加密数据的“n”个伪随机份。然后可以利用剩余的秘密钥匙值对每一份执行后续 XOR 函数，例如安全数据段 1 ＝加密数据份 1XOR 秘密钥匙 1 ：

[0358] SD1 ＝ S1 XOR K1

[0359] SD2 ＝ S2 XOR K2

[0360] SD3 ＝ S3 XOR K3

[0361] SDn ＝ Sn XOR Kn。

[0362] 在一个实施例中，可能不希望任何一个仓库包含足够的信息以解密其中保存的信息，因此解密该份所需要的钥匙被存储在不同的数据仓库中 ：

[0363] 仓库 1 :SD1，Kn

[0364] 仓库 2 :SD2，K1

[0365] 仓库 3 :SD3，K2

[0366] 仓库 n :SDn，K3。

[0367] 另外，附加到每一份的可以是取回原始会话加密钥匙 K5 所需要的信息。因此，在此处所描述的钥匙管理示例中，原始会话主钥被事务 ID 所引用，该事务 ID 根据取决于安装的解析器主钥的内容被分割成“n”份 (TID1、TID2、TID3、TIDn) ：

[0368] 仓库 1 :SD1，Kn，TID1

[0369] 仓库 2 :SD2，K1，TID2

[0370] 仓库 3 :SD3，K2，TID3

[0371] 仓库 n :SDn，K3，TIDn。

[0372] 在此处所描述的所结合的会话钥匙示例中，会话主钥根据取决于安装的解析器主钥被分割成“n”份 (SK1、SK2、SK3、SKn) ：

[0373] 仓库 1 :SD1，Kn，SK1

[0374] 仓库 2 :SD2，K1，SK2

[0375] 仓库 3 :SD3，K2，SK3

[0376] 仓库 n :SDn，K3，SKn。

[0377] 根据此示例，除非取回了所有四份，否则无法重新组装数据。即使捕捉了所有的四份，在不访问会话主钥和解析器主钥的情况下，也不可能重新组装或恢复原始信息。

[0378] 此示例描述了本发明的方法的一个实施例，并且在另一个实施例中，还描述了用于将份放置在仓库中以便来自所有仓库的份可以被组合以形成秘密认证材料的算法。所需要的计算是非常简单和迅速的。但是，利用一次性密码本 (OTP) 算法，可以存在导致它不那么符合需要的情况，例如要保护大数据集合的情况，这是因为钥匙大小与要存储的数据的大小相同。因此，将会需要存储和发送原始数据的约两倍的量，这在某些情况下可能是不那么符合需要的。

[0379] 流密码 RS1

[0380] 流密码 RS1 分割技术与此处描述的 OTP 分割技术非常相似。取代 n 字节随机值，一个 n’＝ min(n,16) 字节的随机值被生成和用于作为 RS1 流密码算法的钥匙。RS1 流密码算法的优点是从小得多的种子数生成伪随机钥匙。执行 RS1 流密码加密的速度也被估计

为大概是本领域中公知的三倍 DES(Triple DES) 加密的速度的 10 倍，而不会危害安全性。RS1 流密码算法是本领域中公知的，并且可以用于生成 XOR 函数中使用的钥匙。RS1 流密码算法可以与其他商业上可获得的流密码算法协同工作，并用适用于本发明的方法中，所述其他商业上可获得的流密码算法例如是 RSA Security，Inc 的 RC4™ 流密码算法。

[0381] 利用上述钥匙记号，K1 至 K5 现在是 n’ 字节随机值，并且我们设置：

[0382] SD1 = S1 XOR E(K1)

[0383] SD2 = S2 XOR E(K2)

[0384] SD3 = S3 XOR E(K3)

[0385] SDn = Sn XOR E(Kn)

[0386] 其中 E(K1) 至 E(Kn) 是来自由 K1 至 Kn 为钥匙的 RS1 流密码算法的输出的前 n’ 字节。现在，如此处所描述的，这些份被放置在数据仓库中。

[0387] 在此流密码 RS1 算法中，所需要的计算几乎与 OPT 算法一样简单和迅速。这个使用 RS1 流密码的示例的优点是对于每一份，系统需要存储和发送的大小只比要保护的原始数据的大小平均约大 16 字节。当原始数据的大小大于 16 字节时，此 RS1 算法比 OPT 算法更高效，因为它更短。本领域的普通技术人员易于看出，多种加密方法或算法适合用于本发明中，包括但不限于 RS1、OTP、RC4™、三倍 DES 和 AES。

[0388] 比起传统加密方法来，本发明的数据安全性方法和计算机系统提供了重大优点。一个优点是由于将数据份移动到可能处于不同逻辑、物理或地理位置中的一个或多个数据仓库或存储设备上的不同位置而获得的安全性。例如，当数据份被物理地分割并且在不同人员的控制之下时，危害到数据的可能性大大降低。

[0389] 本发明的方法和系统提供的另一个优点是组合本发明的用于保护数据的方法的步骤，以提供维护敏感数据安全性的综合过程。数据被用安全钥匙加密，并且根据安全钥匙被分割成一份或多份，在一个实施例中是被分割成四份。安全钥匙与引用指针安全地存储在一起，该引用指针被根据安全钥匙分割成四份。然后数据份被独立加密，并且钥匙与不同的加密份安全地存储在一起。当被组合时，根据此处公开的方法的用于保护数据的整个过程变成用于数据安全性的综合包。

[0390] 根据本发明的方法来保护的数据易于取回和恢复、重新构成、重新组装、解密或以其他方式返回其原始或其他适合使用的形式。为了恢复原始数据，可利用以下项目：

[0391] 1. 数据集合的所有份或部分。

[0392] 2. 关于再现用于保护数据的方法的过程流程的知识，以及再现用于保护数据的方法的过程流程的能力。

[0393] 3. 对会话主钥的访问权限

[0394] 4. 对解析器主钥的访问权限

[0395] 因此，可能希望计划一种安全的安装，其中以上元素中的至少一个可以与系统的其他成分从物理上相分离（例如在不同系统管理员的控制之下）。

[0396] 可通过使用解析器主钥，来加强调用数据保护方法应用的针对欺诈应用的保护。在本发明的这个实施例中，在采取任何动作之前，可能需要 Secure Parser™ 和应用之间的相互认证握手。

[0397] 系统的安全性规定不存在用于重新创建原始数据的“后门 (backdoor)”方法。对

于可能出现数据恢复问题的安装，可增强 Secure Parser™ 以提供四个份和会话主钥仓库的镜像。在数据恢复计划设定中，诸如 RAID（冗余廉价磁盘阵列，用于在几个磁盘上散布信息）这样的硬件选项以及诸如复制这样的软件选项也能有所帮助。

[0398] 钥匙管理

[0399] 在本发明的一个实施例中，对于一个加密操作，数据保护方法使用三个钥匙集合。基于安装，每个钥匙集合具有各自的钥匙存储、检索、安全性和恢复选项。可使用的钥匙包括但不限于：

[0400] 1. 解析器主钥

[0401] 此钥匙是与数据解析器的安装相关联的个体钥匙。它被安装在部署所述解析器的服务器上。存在多种适用于保护此钥匙的选项，例如包括但不限于：智能卡、单独硬件钥匙存储、标准钥匙存储、定制钥匙存储或存储在受保护的数据表内。

[0402] 2. 会话主钥

[0403] 每次保护数据时可生成会话主钥。会话主钥被用于在解析操作之前对数据进行加密。它还可以被结合成为解析加密数据的手段（如果会话主钥未被结合到解析数据中的话）。可以用多种方式来保护会话主钥，例如包括但不限于：标准钥匙存储、定制钥匙存储、单独数据库表或被保护在加密后的份内。

[0404] 3. 份加密钥匙

[0405] 对于创建的数据集合的每一份或每个部分，可生成各自的份加密钥匙，以进一步对这些份进行加密。份加密钥匙可以被存储在与被加密的份不同的份中。

[0406] 本领域的普通技术人员易于看出，本发明的数据保护方法和计算机系统可广泛应用于任何设置或环境中的任何类型的数据。除了在因特网上或顾客和销售者之间进行的商业应用外，本发明的数据保护方法和计算机系统还高度适用于非商业或私用设置或环境。可以用此处描述的方法和系统来保护需要被保护以免被未经授权的用户所危害的任何数据集合。例如，通过采用本发明的用于保护数据的方法和系统，可将对公司或组织内的特定数据库的访问仅限于被选中的用户。另一个示例是文档的生成、修改或访问，其中需要限制选中的个体、计算机或工作站的群组之外的访问或防止未经授权的或意外的访问或公开。本发明的数据保护方法和系统的方式的这些和其他示例适用于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。

[0407] 在本发明的另一个实施例中，对于一个加密操作，数据保护方法使用三个钥匙集合。基于安装，每个钥匙集合具有各自的钥匙存储、检索、安全性和恢复选项。可使用的钥匙包括但不限于：

[0408] 1. 解析器主钥

[0409] 此钥匙是与数据解析器的安装相关联的个体钥匙。它被安装在部署所述解析器的服务器上。存在多种适用于保护此钥匙的选项，例如包括但不限于：智能卡、单独硬件钥匙存储、标准钥匙存储、定制钥匙存储或存储在受保护的数据表内。

[0410] 2. 会话主钥

[0411] 每次保护数据时可生成会话主钥。会话主钥被用于结合解析器主钥得出中间钥。可以用多种方式来保护会话主钥，例如包括但不限于：标准钥匙存储、定制钥匙存储、单独数据库表或被保护在加密后的份内。

[0412] 3. 中间钥

[0413] 每次保护数据时可生成中间钥。中间钥被用于在解析操作之前对数据进行加密。它还可以被结合成为解析加密数据的手段。

[0414] 4. 份加密钥匙

[0415] 对于创建的数据集合的每一份或每个部分，可生成各自的份加密钥匙，以进一步对这些份进行加密。份加密钥匙可以被存储在与被加密的份不同的份中。

[0416] 本领域的普通技术人员易于看出，本发明的数据保护方法和计算机系统可广泛应用于任何设置或环境中的任何类型的数据。除了在因特网上或顾客和销售者之间进行的商业应用外，本发明的数据保护方法和计算机系统还高度适用于非商业或私用设置或环境。可以用此处描述的方法和系统来保护需要被保护以免被未经授权的用户所危害的任何数据集合。例如，通过采用本发明的用于保护数据的方法和系统，可将对公司或组织内的特定数据库的访问仅限于被选中的用户。另一个示例是文档的生成、修改或访问，其中需要限制选中的个体、计算机或工作站的群组之外的访问或防止未经授权的或意外的访问或公开。本发明的数据保护方法和系统的方式的这些和其他示例适用于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。

[0417] 工作组、项目、个体 PC/ 笔记本电脑或变叉平台数据安全性

[0418] 本发明的数据保护方法和计算机系统也可用于保护工作组、项目、个体 PC/ 笔记本和任何其他平台产生数据，这些工作组、项目、个体 PC/ 笔记本和任何其他平台例如被用于商行、工作站、政府机构或产生、处理或存储敏感数据的任何设置。本发明提供了用于保护已知被诸如美国政府这样的组织所寻求的数据的方法和计算机系统，以便在整个政府组织上实现或在国家或联邦级别上在政府间实现。

[0419] 本发明的数据保护方法和计算机系统不仅提供了解析平面文本的能力，还提供了解析任何类型的数据字段、集合和 / 或表的能力。此外，在此过程下，能够保护任何形式的数据，包括但不限于文本、视频、图像、生物特征量度和语音数据。本发明的保护数据的方法的可缩放性、速度和数据吞量只受用户可使用的硬件所限。

[0420] 在本发明的一个实施例中，如下所述，数据保护方法被用于工作组环境中。在一个实施例中，如图 23 所示和如下所述，本发明的工作组规模 (Workgroup Scale) 数据保护方法使用 TrustEngine 的私钥管理功能来存储用户 / 群组关系以及用户群组共享安全数据所必需的相关联的私钥（解析器群组主钥）。根据解析器主钥的部署方式，本发明的方法能够为企业、工作组或个体用户保护数据。

[0421] 在一个实施例中，可提供附加钥匙管理和用户 / 群组管理程序，以使得利用单点管理和钥匙管理来实现大规模工作组。钥匙生成、管理和撤回由单个维护程序所处理，随着用户数目增大，这些都变得尤其重要。在另一个实施例中，也可跨一个或几个不同系统管理员建立钥匙管理，根据需要，这样做可能不允许任何一个个人或群组控制数据。这允许了根据组织定义的角色、职责、成员身份、权利等等来管理受保护的数据，并且对受保护数据的访问可仅限于被允许或要求仅能访问它们正在工作的那部分的那些人，而其他人，例如管理员或主管人员可以拥有对所有受保护数据的访问权限。此实施例允许了在公司或组织内的不同群组间共享受保护数据，同时只允许某些选中的个体，例如具有授权和预定的角色和职责的个体从整体上察看数据。此外，本发明的此实施例和本发明的系统还允许了在如

下组织之间共享数据：这些组织例如是要求某种共享但不是任何一方都被允许访问所有数据的单独的公司、或单独的公司部分或分部、或任何政府或组织的任何单独的组织部门、群组、机构或办事处，或任何类型的任何单独的组织部门、群组、机构或办事处。对本发明的这种方法和系统的需求和利用的特别明显的示例例如是用于允许在政府区域、机构和办事处之间以及大公司的不同分部、部分或办事处之间或任何其他组织之间进行共享，但保持安全性。

[0422] 本发明的方法对于较小的规模的实用性的示例如下。解析器主钥被用作对于组织的解析器连载 (serialization) 或标记 (branding)。由于使用解析器主钥的规模从整个企业减小到了较小的工作组，因此此处所描述的数据保护方法被用于在用户群组内共享文件。

[0423] 在图 25 所示以及下文所述的示例中，定义了六个用户以及他们在组织内的头衔或角色。侧边条代表用户根据其角色可能属于的五个群组。箭头代表用户在一个或多个群组中的成员身份。

[0424] 当配置 SecureParser 以用于此示例中时，系统管理员通过维护程序访问来自操作系统的用户和群组信息。此维护程序基于用户在群组中的成员身份向用户生成和分配解析器群组主钥。

[0425] 在此示例中，在高级职员群组中有三个成员。对于此群组，动作将会是：

[0426] 1. 访问高级职员群组的解析器群组主钥（如果钥匙不可用则生成钥匙）；

[0427] 2. 生成将 CEO 与高级成员群组关联起来的数字证书；

[0428] 3. 生成将 CFO 与高级成员群组关联起来的数字证书；

[0429] 4. 生成将 Vice President,Marketing（副总裁，市场）与高级成员群组关联起来的数字证书；

[0430] 对于每个群组和每个群组内的每个成员，可完成相同的动作集合。当维护程序完成时，解析器群组主钥变成群组的每个成员的共享证书。当从群组中去除用户时，可通过维护程序自动完成已分配的数字证书的撤回，而不会影响群组的其他成员。

[0431] 一旦定义了共享的证书，解析器过程就保持不变。当要保护文件、文档或数据元素时，用户被提示保护数据时要使用的目标群组。所产生的受保护数据只能由目标群组的其他成员访问。本发明的方法和系统的这个功能可与任何其他计算机系统或软件平台一起使用，并且例如可以被集成到现有应用程序中，或被独立用于文件安全性。

[0432] 本领域的普通技术人员易于看出，加密算法中的任何一种或其组合都适用于本发明的方法和系统。例如，在一个实施例中，可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织可用于重复的加密步骤中，以便不同的加密算法被应用到多层加密方案的不同层。这样，加密方案本身变成本发明的方法的成分，以用于保护敏感数据免遭未经授权的使用或访问。

[0433] 此外，本领域的技术人员在考虑到此处的公开文本的情况下将会明显看出其他组合、允许、替换和修改。因此，本发明不希望受优选实施例的反应所限，而是通过参考所附权利要求书来限定。

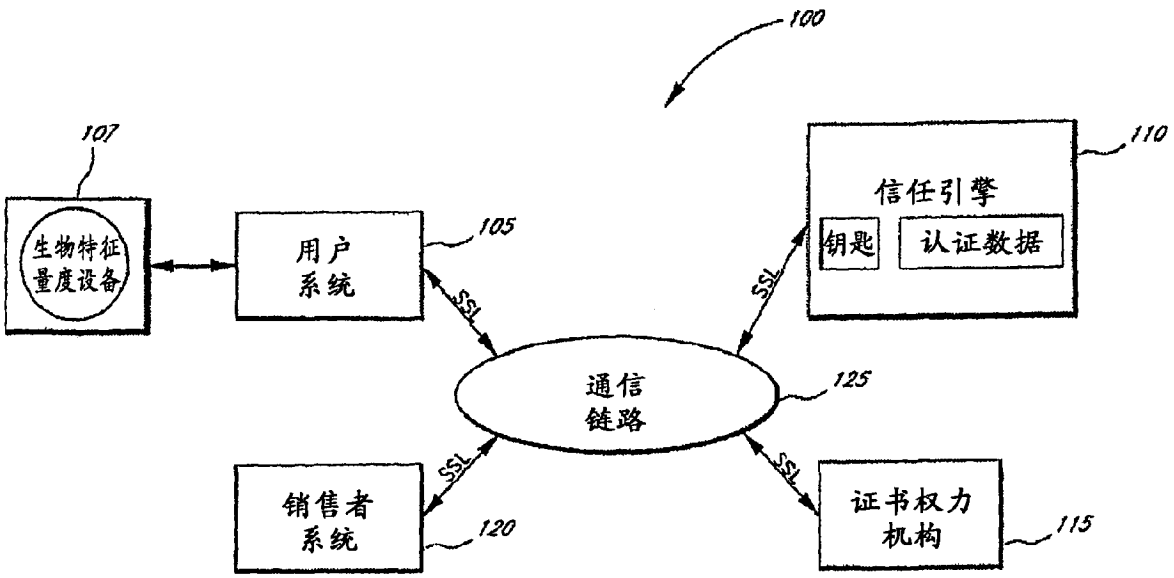

100
107
生物特征量度设备
用户系统
105
SSL
通信链路
125
信任引擎
钥匙
认证数据
110
SSL
SSL
销售者系统
120
证书权力机构
115

图 1

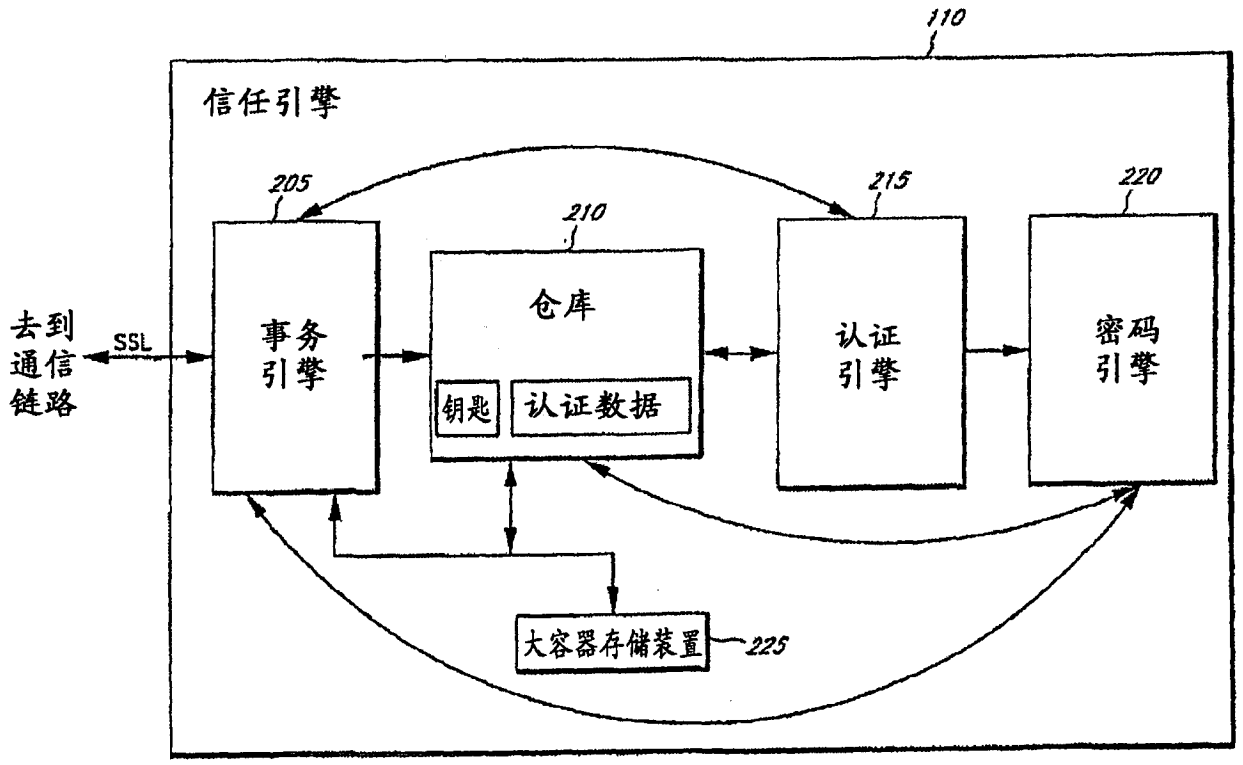

110
信任引擎
205
210
215
220
去到通信链路
SSL
事务引擎
仓库
钥匙
认证数据
认证引擎
密码引擎
大容器存储装置
225

图 2

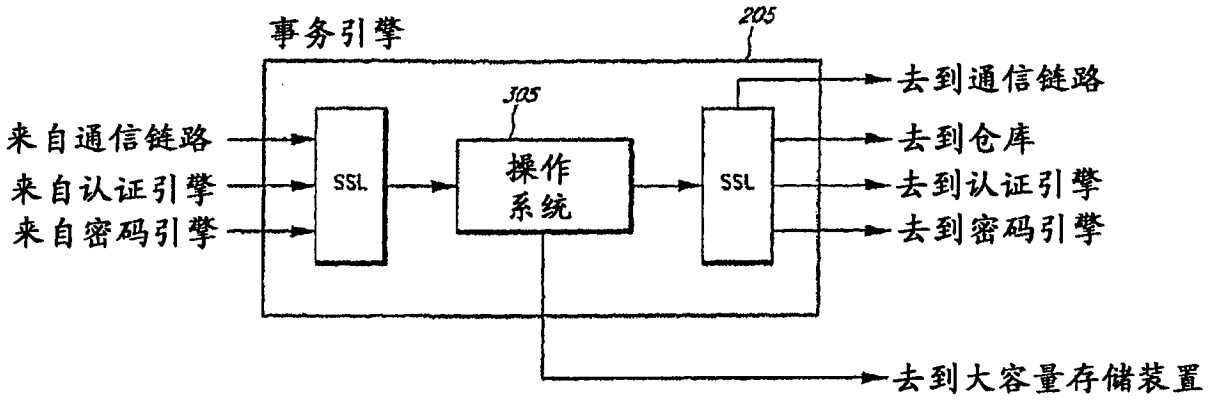

事务引擎
205
305
来自通信链路
来自认证引擎
来自密码引擎
SSL
操作系统
SSL
去到通信链路
去到仓库
去到认证引擎
去到密码引擎
去到大容量存储装置

图 3

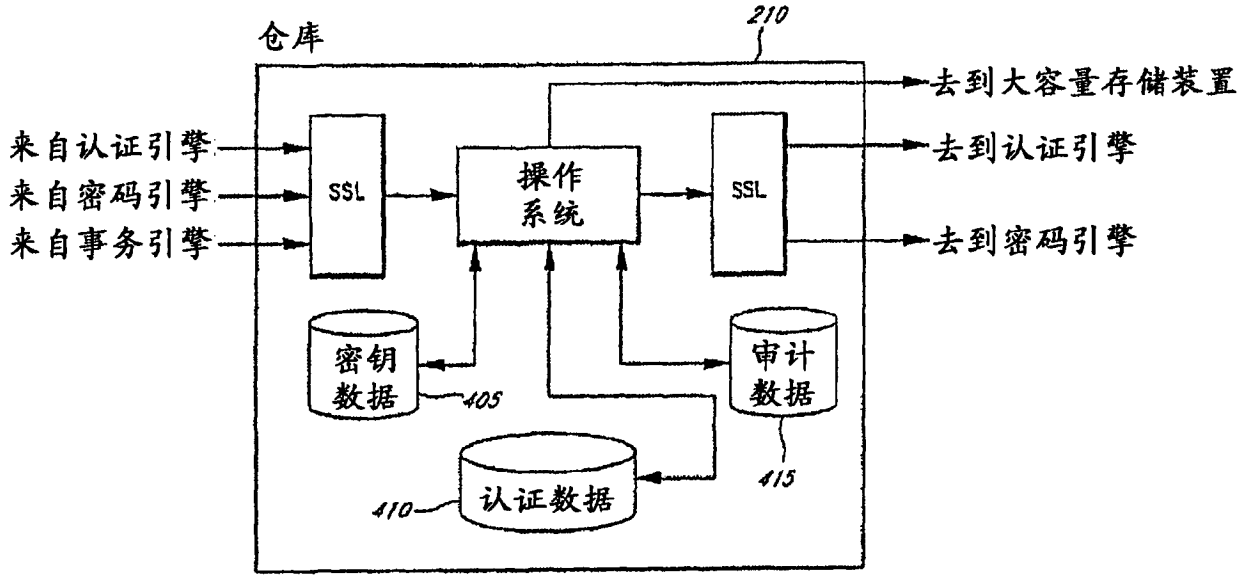

仓库
210
来自认证引擎
来自密码引擎
来自事务引擎
SSL
操作系统
SSL
去到大容量存储装置
去到认证引擎
去到密码引擎
密钥数据
405
审计数据
415
认证数据
410

图 4

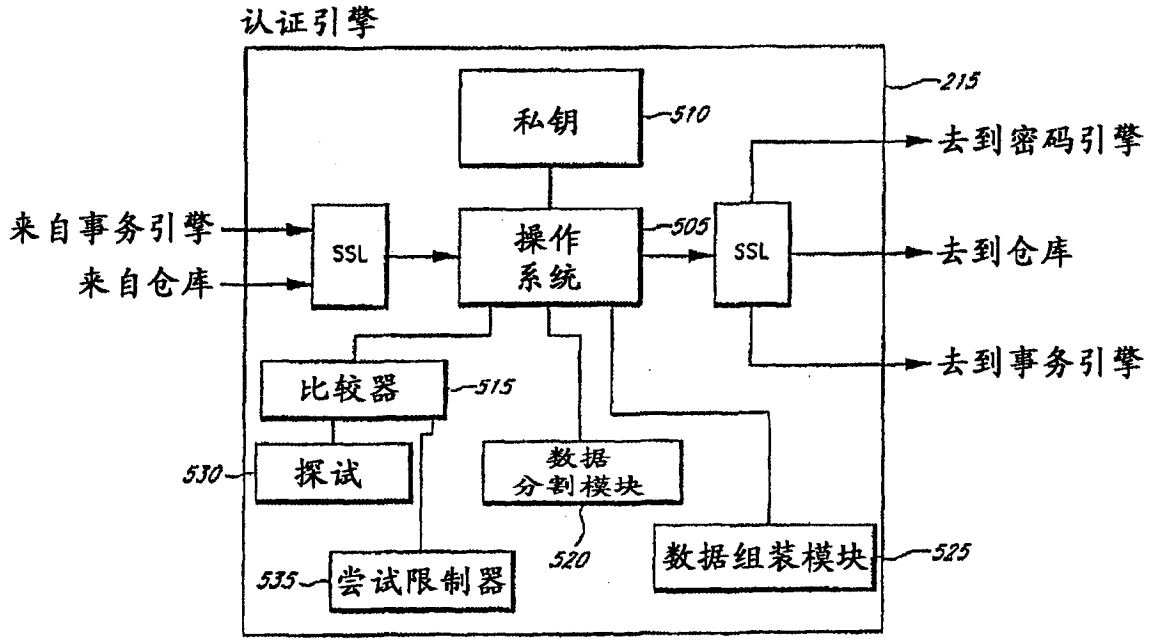

认证引擎
215
私钥
510
去到密码引擎
来自事务引擎
来自仓库
SSL
操作系统
505
SSL
去到仓库
去到事务引擎
比较器
515
530
探试
数据分割模块
520
数据组装模块
525
535
尝试限制器

图 5

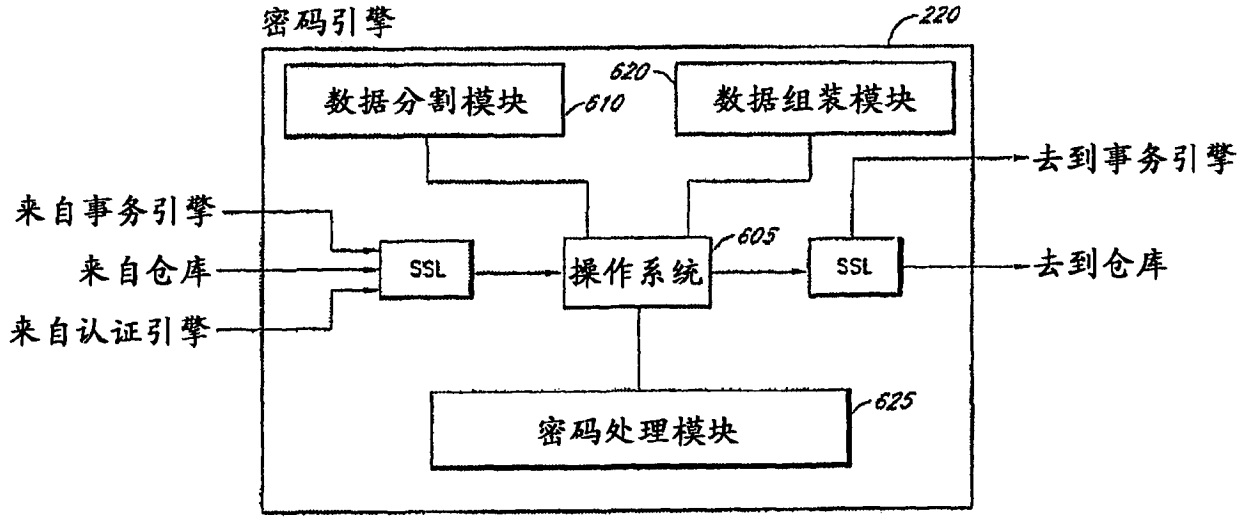

密码引擎
220
620
数据分割模块
610
数据组装模块
去到事务引擎
来自事务引擎
来自仓库
来自认证引擎
SSL
605
操作系统
SSL
去到仓库
625
密码处理模块

图 6

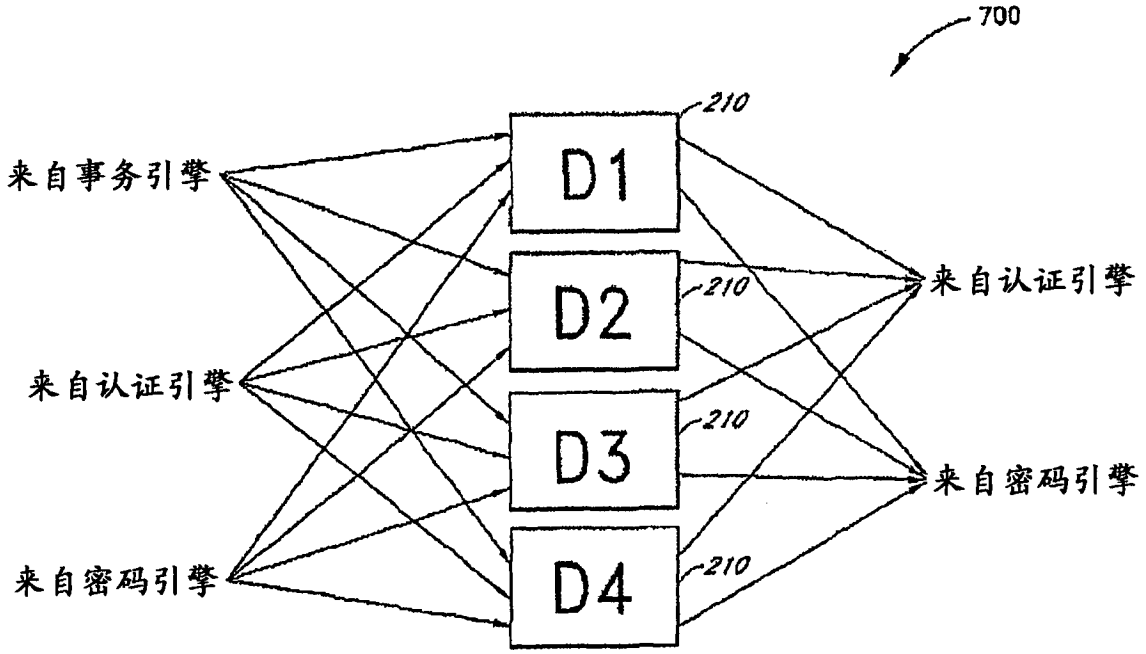


图 7

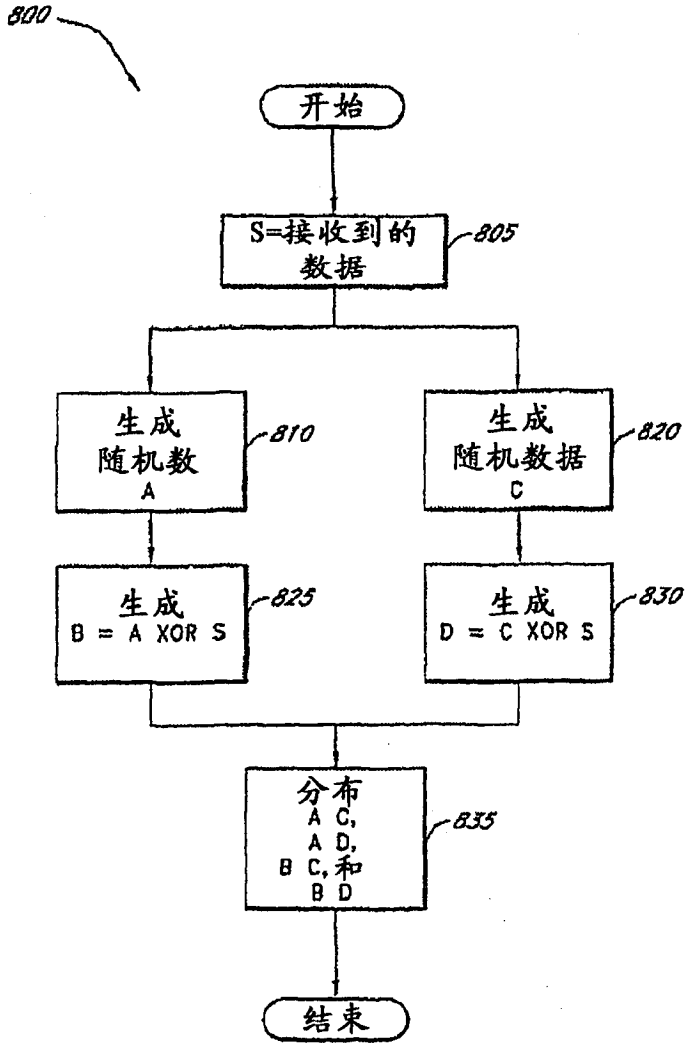

800
开始
S=接收到的数据
805
生成随机数A
810
生成随机数据C
820
生成B = A XOR S
825
生成D = C XOR S
830
分布A C, A D, B C, 和B D
835
结束

图 8

	注册数据流			
	发送	接收	SSL	动作
905	用户	事务引擎 (TE)	1/2	发送(PUB_AE(UID,B))形式的利用认证引擎(AE)公钥加密的注册认证数据(B)和用户ID(UID)
915	TE	AE	完全	转发传输
920				AE解密并分割转发的数据
925	AE	第X个仓库(DX)	完全	存储各自的那部分数据
	当请求数字证书时			
930	AE	密码引擎(CE)	完全	请求钥匙生成
935				CE生成并分割钥匙
945	CE	TE	完全	发送对数字证书的请求
950	TE	证书权力机构(CA)	1/2	发送请求
955	CA	TE	1/2	发送数字证书
960	TE	用户	1/2	发送数字证书
	TE	MS	完全	存储数字证书
965	CE	DX	完全	存储各自的那部分钥匙

图 9A

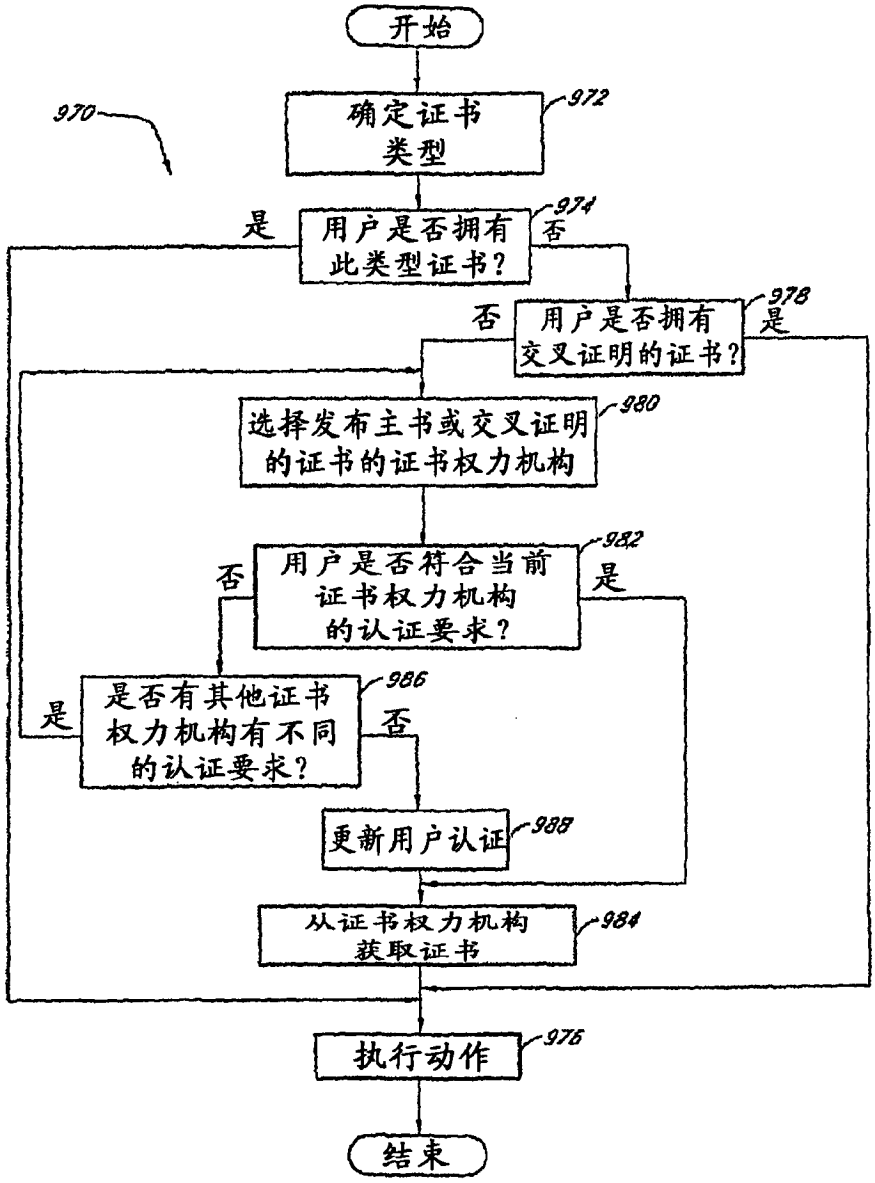
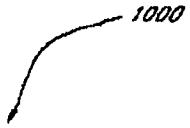

开始
970
确定证书类型
972
是
用户是否拥有此类型证书？
974
否
否
用户是否拥有交叉证明的证书？
978
是
选择发布主书或交叉证明的证书的证书权力机构
980
否
用户是否符合当前证书权力机构的认证要求？
982
是
是
是否有其他证书权力机构有不同的认证要求？
986
否
更新用户认证
988
从证书权力机构获取证书
984
执行动作
976
结束

图 9B



认证数据流

	发送	接收	SSL	动作
1005	用户	销售者	1/2	事务发生，例如选择购买
1010	销售者	用户	1/2	发送事务ID(TID)和认证请求(AR)
				从用户收集认证数据(B′)
1015	用户	TE	1/2	发送(PUB_AE(TID, B′))形式的利用认证引擎(AE)公钥包装的TID和B′
1020	TE	AE	完全	转发传输
				注册认证数据(B)被请求和收集
1025	销售者	事务引擎(TE)	完全	发送TID、AR
1030	TE	大容量存储装置(MS)	完全	在数据库中创建记录
1035	TE	第X个仓库(DX)	完全	UID, TID
1040	DX	AE	完全	发送TID和注册(BX)处存储的那部分认证数据(PUB_AE(TID, BX))
1045				
1050				AE组装B并其将与B′相比较
	AE	TE	完全	TID, 已填充的AR
	TE	销售者	完全	TID, 是/否
1055	TE	用户	1/2	TID, 确认消息

图 10

1100

签署数据流			
发送	接收	SSL	动作
用户	销售者	1/2	事务发生，例如就交易达成协议
销售者	用户	1/2	发送事务标识号码(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′)和用户接收到的消息散列(h(M′))
用户	TE	1/2	发送(PUB_AX(TID, B′, h(M′))形式的以认证引擎(AE)公钥包装的TID、B′、AR和h(M′)
TE	AE	完全	转发传输
			收集注册认证数据
销售者	事务引擎(TE)	完全	发送UID、TID、AR和消息散列(h(M))
TE	大容量存储装置(MS)	完全	在数据库中创建记录
TE	第X个仓库(DX)	完全	UID, TID
DX	AE	完全	发送(PUB_AE(TID, BX))形式的TID和存储在注册(BX)处的那部分认证数据
			原始销售者消息被发送到AE
TE	AE	完全	发送h(M)
			AE组装B，将其与B′相比较，并将h(M)与h(M′)相比较
AE	密码引擎(CE)	完全	请求数字签名和要签署的消息，例如，散列后的消息
AE	DX	完全	TID、签署UID
DX	CE	完全	发送密钥中与签署方相对应的部分
			CE组装钥匙并签署
CE	AE	完全	发送签署方的数字签名(S)
AE	TE	完全	TID，已填充的AR，h(M)和S
TE	销售者	完全	TID，收据=(TID，是/否，S)，以及例如信任引擎的数字签名，利用信任引擎私钥加密的数据散列(Priv_TE(h(RECEIPT)))
TE	用户	1/2	TID，确认消息

1103 1105 1110 1115 1120 1125 1130 1135 1140

图 11

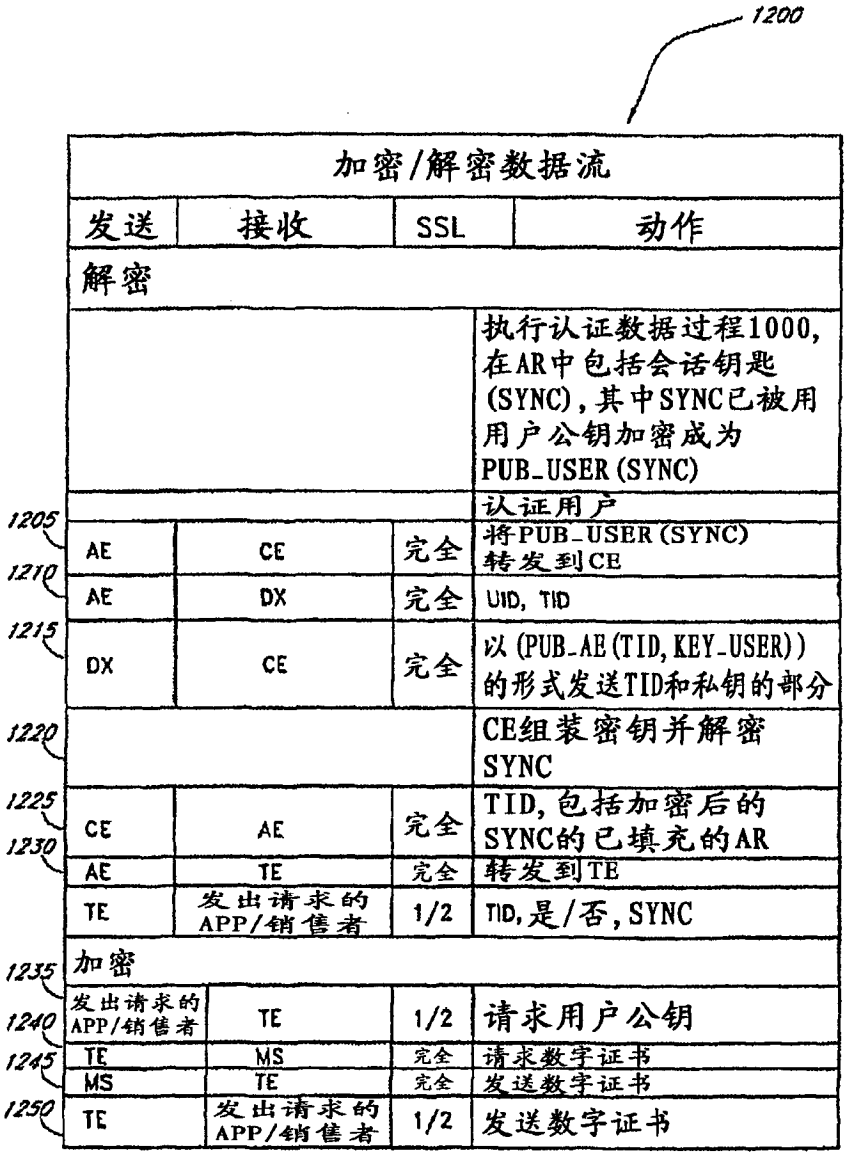

1200
加密/解密数据流
发送
接收
SSL
动作
解密
执行认证数据过程1000，在AR中包括会话钥匙(SYNC)，其中SYNC已被用用户公钥加密成为PUB_USER(SYNC)
认证用户
1205
AE
CE
完全
将PUB_USER(SYNC)转发到CE
1210
AE
DX
完全
UID, TID
1215
DX
CE
完全
以(PUB_AE(TID, KEY_USER))的形式发送TID和私钥的部分
1220
CE组装密钥并解密SYNC
1225
CE
AE
完全
TID，包括加密后的SYNC的已填充的AR
1230
AE
TE
完全
转发到TE
TE
发出请求的APP/销售者
1/2
TID，是/否，SYNC
1235
加密
1240
发出请求的APP/销售者
TE
1/2
请求用户公钥
1245
TE
MS
完全
请求数字证书
MS
TE
完全
发送数字证书
1250
TE
发出请求的APP/销售者
1/2
发送数字证书

图 12

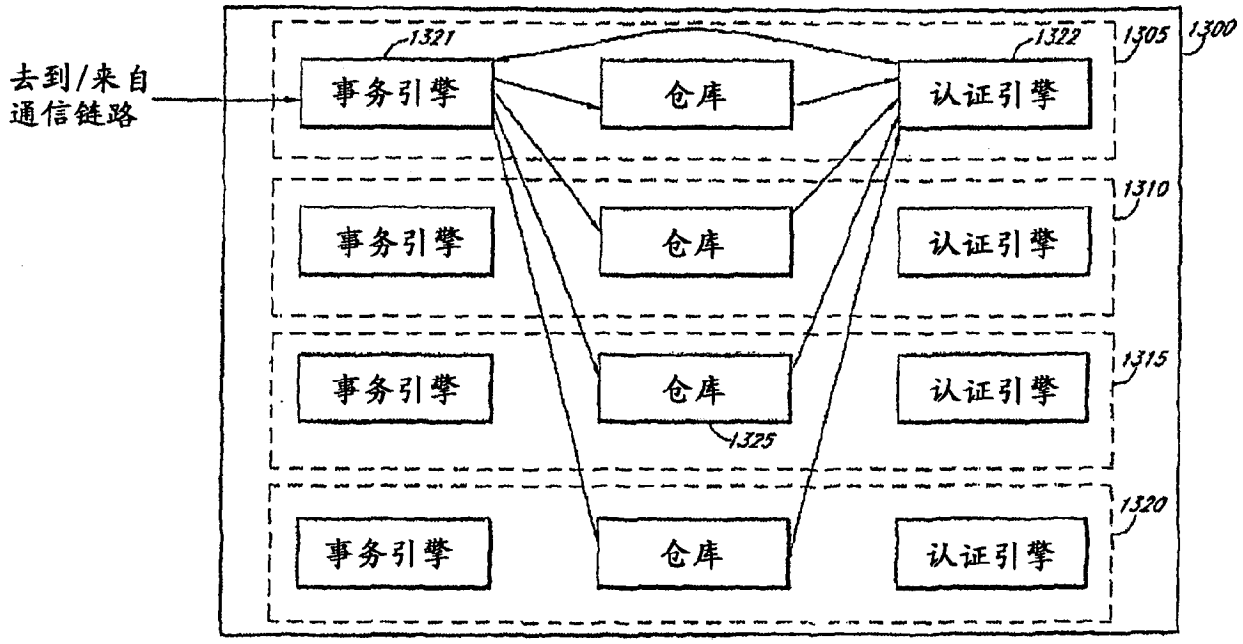

去到/来自
通信链路
1321
1322
1305
1300
事务引擎
仓库
认证引擎
1310
事务引擎
仓库
认证引擎
1315
事务引擎
仓库
认证引擎
1325
1320
事务引擎
仓库
认证引擎

图 13

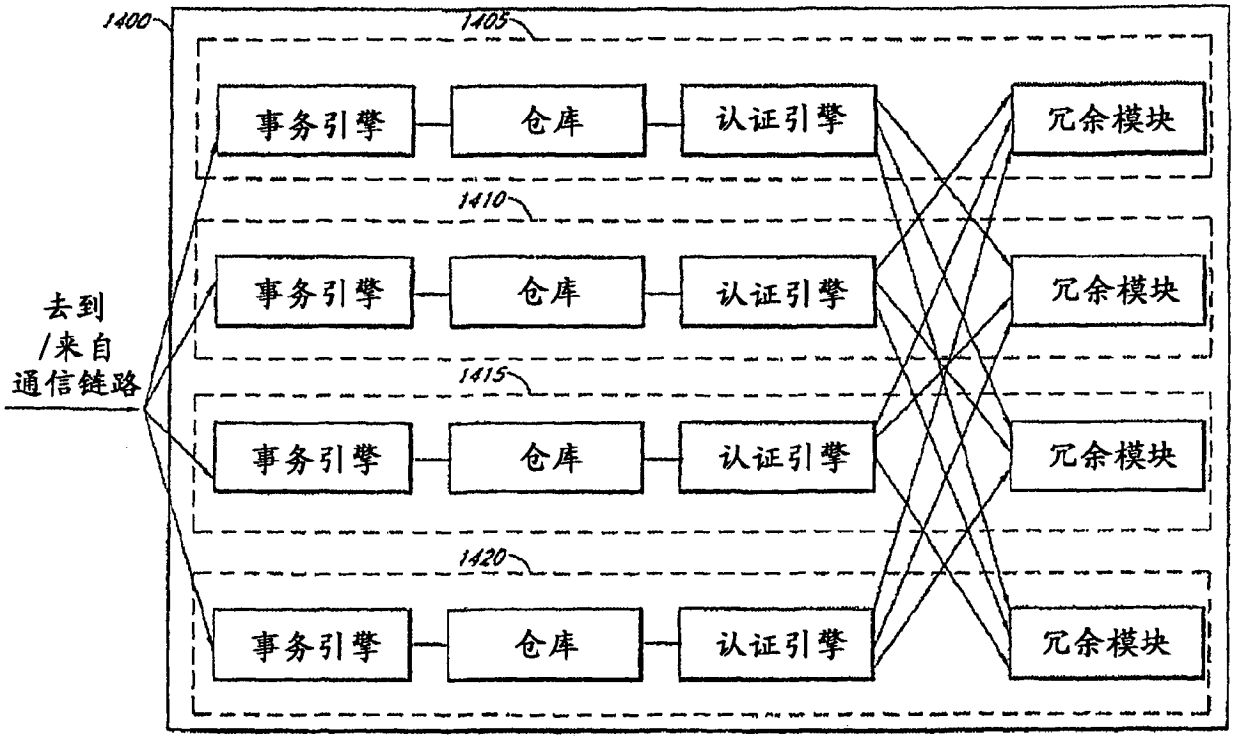

1400
1405
事务引擎
仓库
认证引擎
冗余模块
1410
事务引擎
仓库
认证引擎
冗余模块
去到
/来自
通信链路
1415
事务引擎
仓库
认证引擎
冗余模块
1420
事务引擎
仓库
认证引擎
冗余模块

图 14

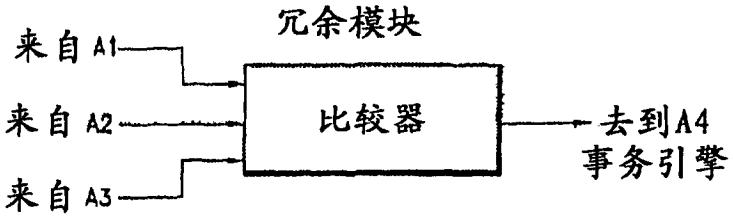

冗余模块
来自 A1
来自 A2
来自 A3
比较器
去到A4
事务引擎

图 15

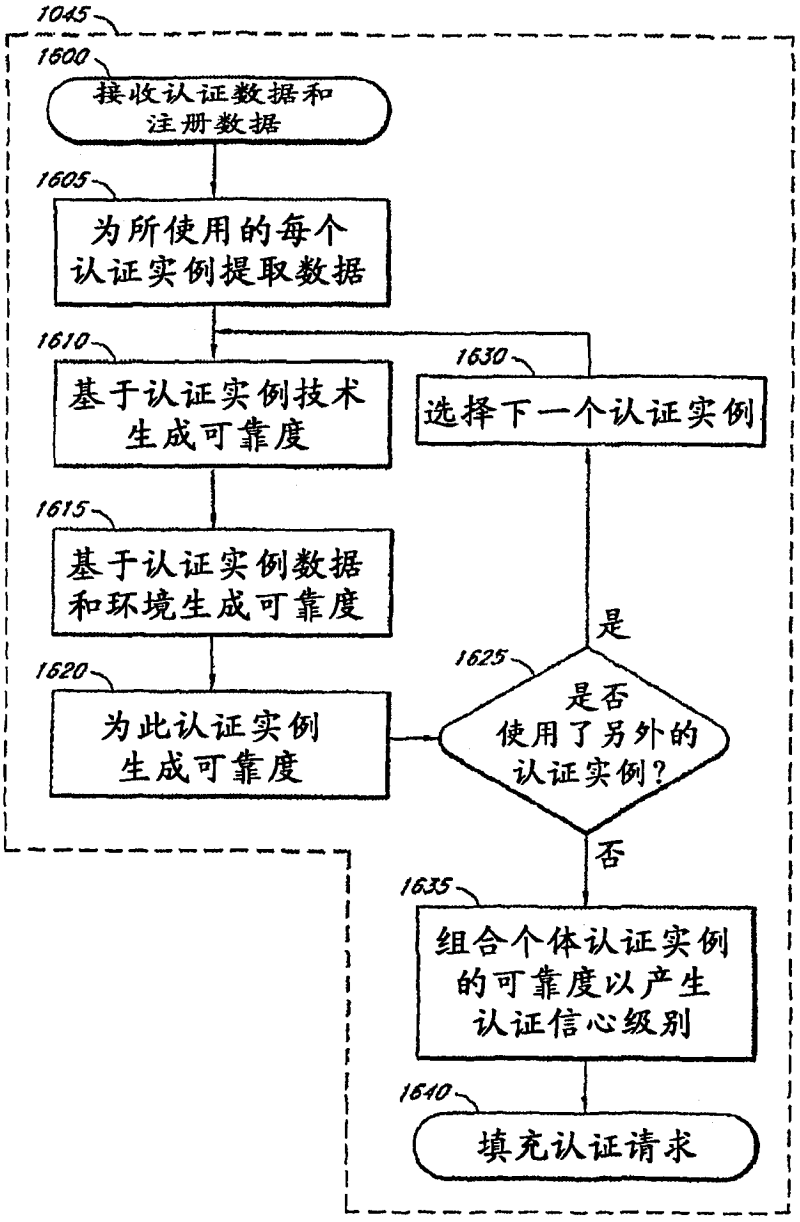

1045
1600
接收认证数据和注册数据
1605
为所使用的每个认证实例提取数据
1610
基于认证实例技术生成可靠度
1615
基于认证实例数据和环境生成可靠度
1620
为此认证实例生成可靠度
1625
是否使用了另外的认证实例？
是
否
1630
选择下一个认证实例
1635
组合个体认证实例的可靠度以产生认证信心级别
1640
填充认证请求

图 16

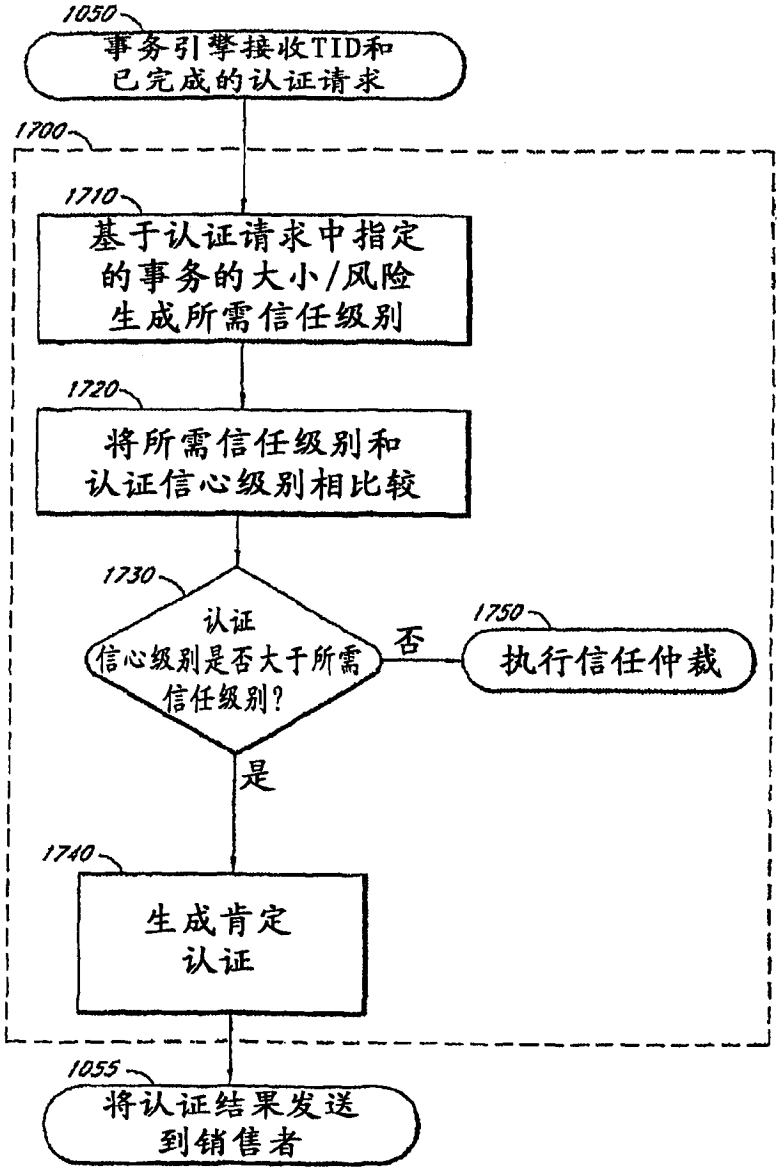

1050
事务引擎接收TID和已完成的认证请求
1700
1710
基于认证请求中指定的事务的大小/风险生成所需信任级别
1720
将所需信任级别和认证信心级别相比较
1730
认证信心级别是否大于所需信任级别?
否
1750
执行信任仲裁
是
1740
生成肯定认证
1055
将认证结果发送到销售者

图 17

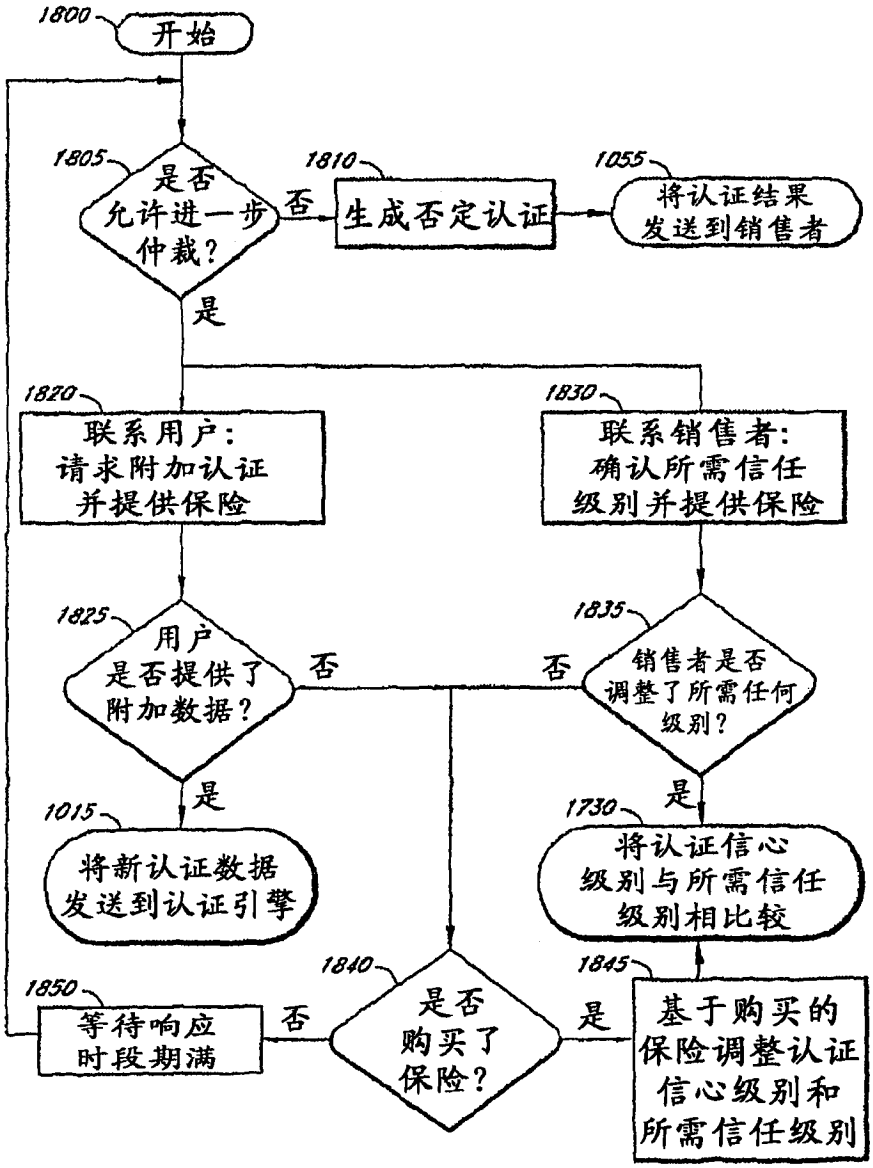

1800
开始
1805
是否允许进一步仲裁?
否
1810
生成否定认证
1055
将认证结果发送到销售者
是
1820
联系用户:请求附加认证并提供保险
1830
联系销售者:确认所需信任级别并提供保险
1825
用户是否提供了附加数据?
否
否
1835
销售者是否调整了所需任何级别?
是
是
1015
将新认证数据发送到认证引擎
1730
将认证信心级别与所需信任级别相比较
1840
是否购买了保险?
1845
基于购买的保险调整认证信心级别和所需信任级别
1850
等待响应时段期满
否
是

图 18

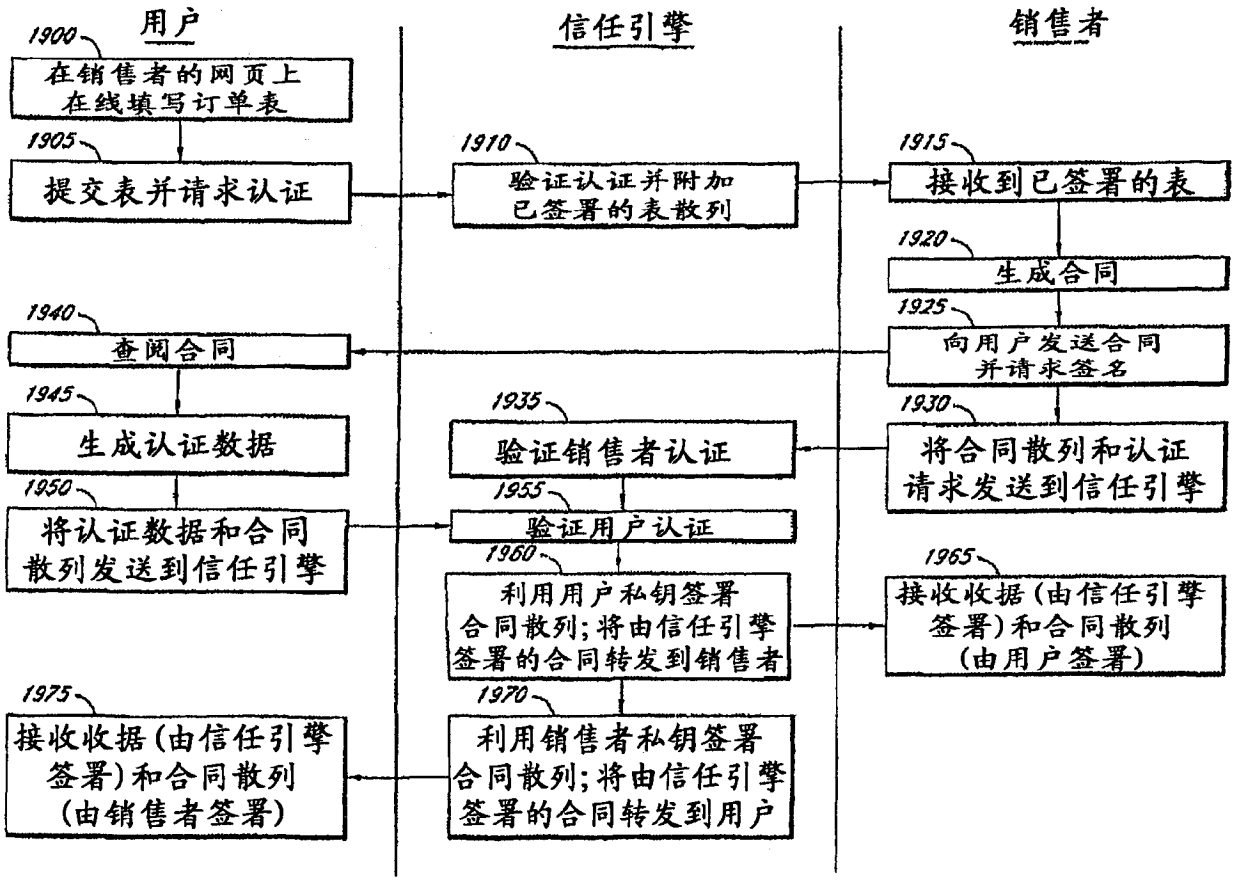

用户
信任引擎
销售者
1900
在销售者的网页上在线填写订单表
1905
提交表并请求认证
1910
验证认证并附加已签署的表散列
1915
接收到已签署的表
1920
生成合同
1925
向用户发送合同并请求签名
1930
将合同散列和认证请求发送到信任引擎
1935
验证销售者认证
1940
查阅合同
1945
生成认证数据
1950
将认证数据和合同散列发送到信任引擎
1955
验证用户认证
1960
利用用户私钥签署合同散列;将由信任引擎签署的合同转发到销售者
1965
接收收据(由信任引擎签署)和合同散列(由用户签署)
1970
利用销售者私钥签署合同散列;将由信任引擎签署的合同转发到用户
1975
接收收据(由信任引擎签署)和合同散列(由销售者签署)

图 19

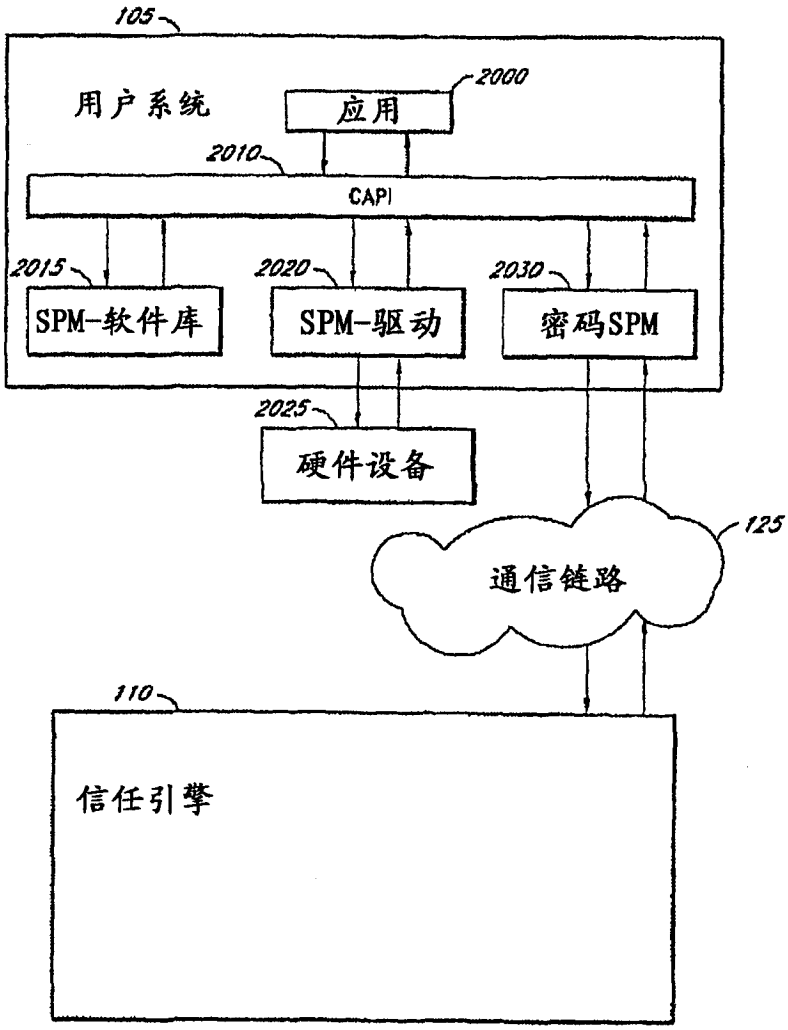

105
用户系统
2000
应用
2010
CAPI
2015
SPM-软件库
2020
SPM-驱动
2030
密码SPM
2025
硬件设备
125
通信链路
110
信任引擎

图 20

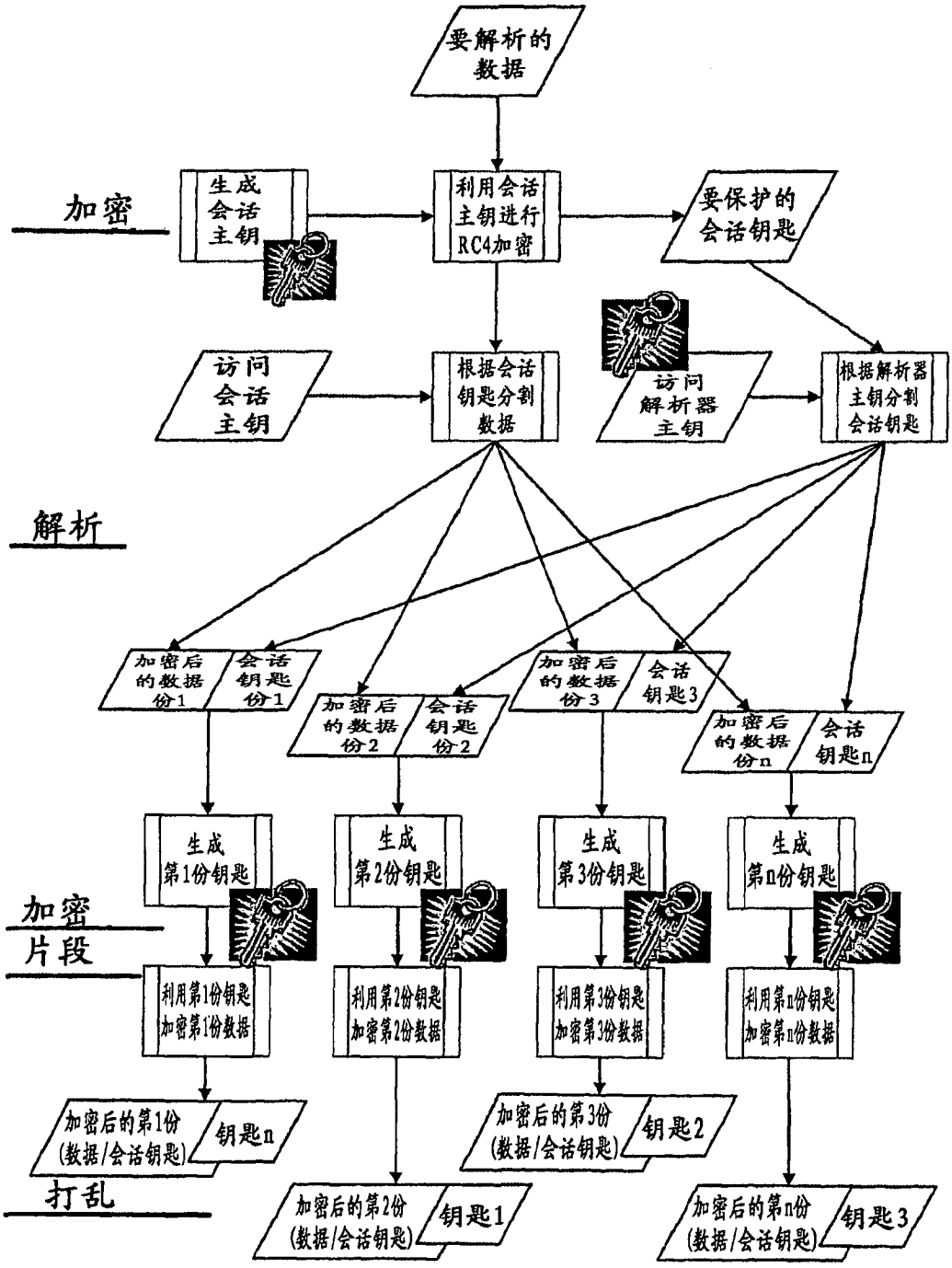

要解析的数据
加密
生成会话主钥
利用会话主钥进行RC4加密
要保护的会话钥匙
访问会话主钥
根据会话钥匙分割数据
访问解析器主钥
根据解析器主钥分割会话钥匙
解析
加密后的数据份1
会话钥匙份1
加密后的数据份2
会话钥匙份2
加密后的数据份3
会话钥匙3
加密后的数据份n
会话钥匙n
生成第1份钥匙
生成第2份钥匙
生成第3份钥匙
生成第n份钥匙
加密片段
利用第1份钥匙加密第1份数据
利用第2份钥匙加密第2份数据
利用第3份钥匙加密第3份数据
利用第n份钥匙加密第n份数据
加密后的第1份(数据/会话钥匙)
钥匙n
加密后的第3份(数据/会话钥匙)
钥匙2
打乱
加密后的第2份(数据/会话钥匙)
钥匙1
加密后的第n份(数据/会话钥匙)
钥匙3

图 21

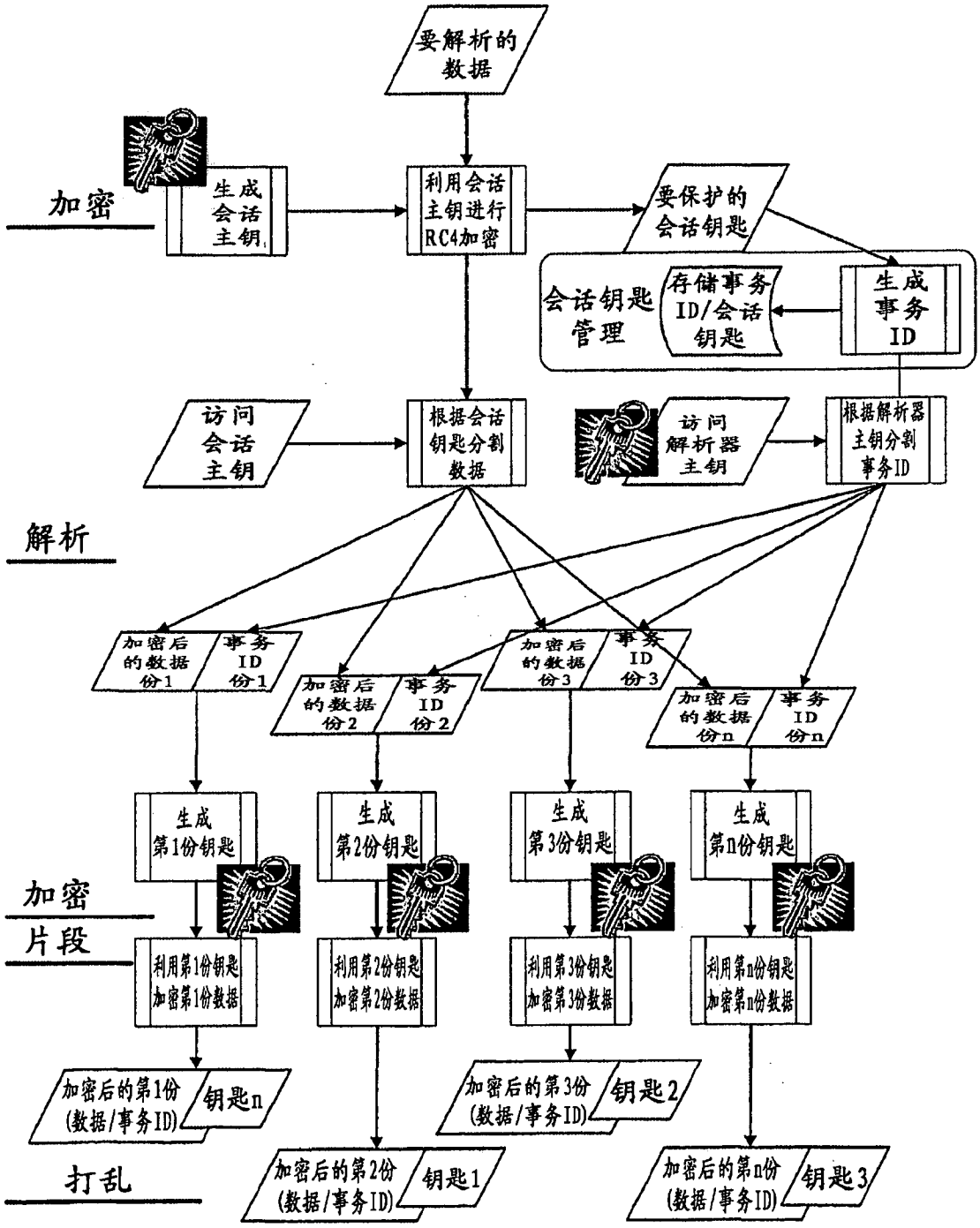

要解析的数据
加密
生成会话主钥
利用会话主钥进行RC4加密
要保护的会话钥匙
会话钥匙管理
存储事务ID/会话钥匙
生成事务ID
访问会话主钥
根据会话钥匙分割数据
访问解析器主钥
根据解析器主钥分割事务ID
解析
加密后的数据份1
事务ID份1
加密后的数据份2
事务ID份2
加密后的数据份3
事务ID份3
加密后的数据份n
事务ID份n
生成第1份钥匙
生成第2份钥匙
生成第3份钥匙
生成第n份钥匙
加密片段
利用第1份钥匙加密第1份数据
利用第2份钥匙加密第2份数据
利用第3份钥匙加密第3份数据
利用第n份钥匙加密第n份数据
加密后的第1份(数据/事务ID)
钥匙n
加密后的第3份(数据/事务ID)
钥匙2
打乱
加密后的第2份(数据/事务ID)
钥匙1
加密后的第n份(数据/事务ID)
钥匙3

图 22

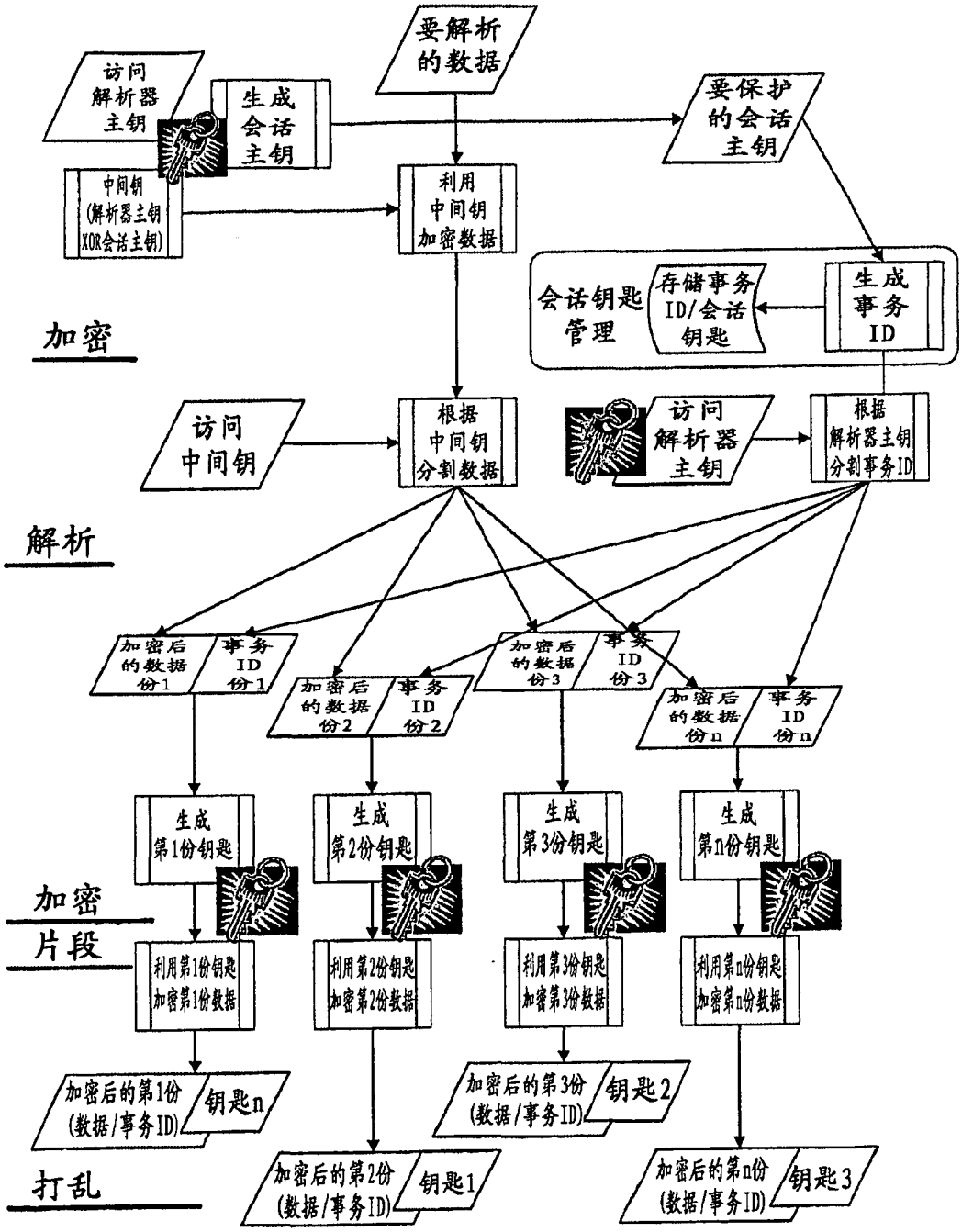

要解析的数据
访问解析器主钥
生成会话主钥
要保护的会话主钥
中间钥(解析器主钥XOR会话主钥)
利用中间钥加密数据
会话钥匙管理
存储事务ID/会话钥匙
生成事务ID
加密
访问中间钥
根据中间钥分割数据
访问解析器主钥
根据解析器主钥分割事务ID
解析
加密后的数据份1
事务ID份1
加密后的数据份2
事务ID份2
加密后的数据份3
事务ID份3
加密后的数据份n
事务ID份n
生成第1份钥匙
生成第2份钥匙
生成第3份钥匙
生成第n份钥匙
加密片段
利用第1份钥匙加密第1份数据
利用第2份钥匙加密第2份数据
利用第3份钥匙加密第3份数据
利用第n份钥匙加密第n份数据
加密后的第1份(数据/事务ID)
钥匙n
加密后的第3份(数据/事务ID)
钥匙2
加密后的第2份(数据/事务ID)
钥匙1
加密后的第n份(数据/事务ID)
钥匙3
打乱

图 23

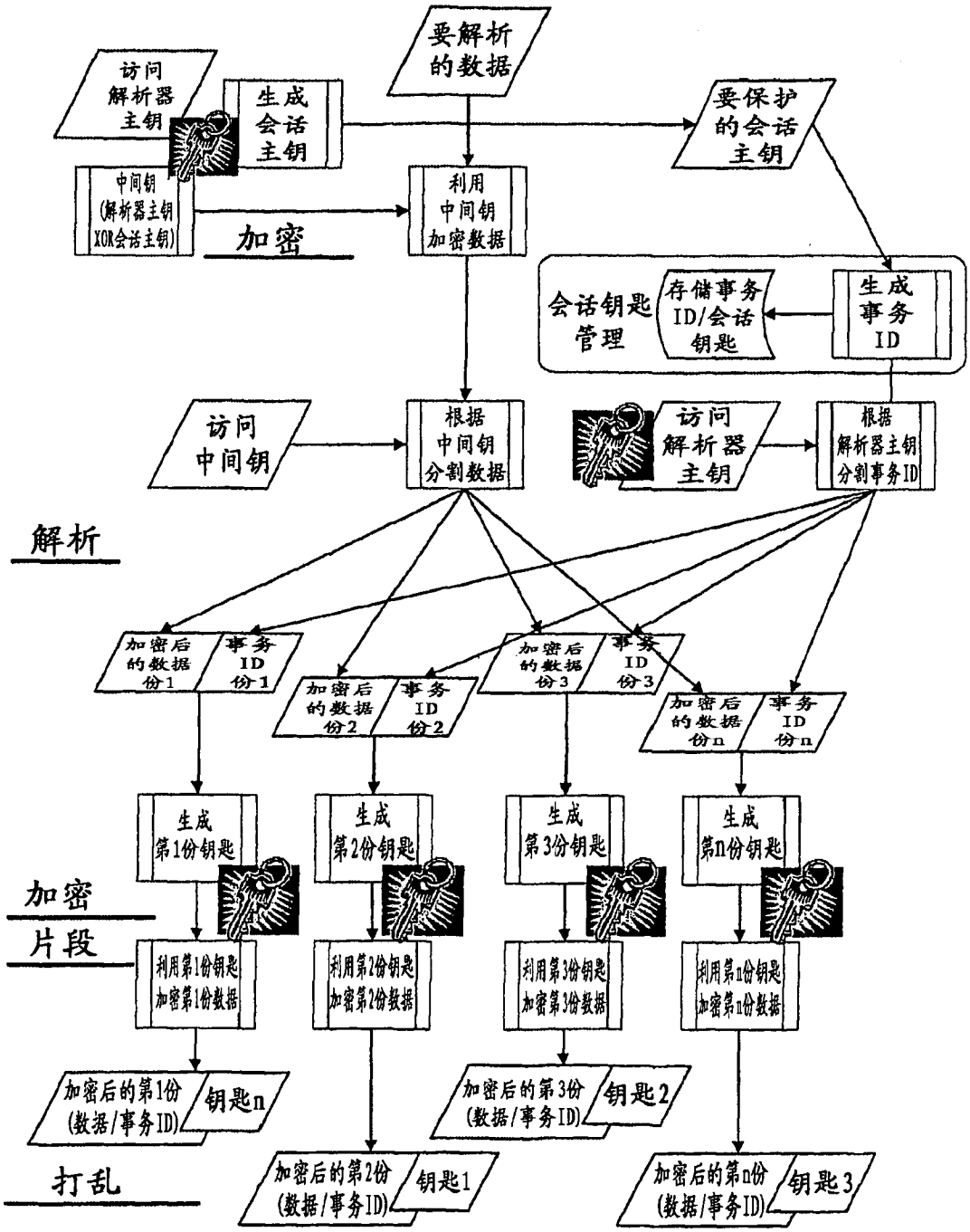

访问
解析器
主钥
生成
会话
主钥
要解析
的数据
要保护
的会话
主钥
中间钥
(解析器主钥
XOR会话主钥)
加密
利用
中间钥
加密数据
会话钥匙
管理
存储事务
ID/会话
钥匙
生成
事务
ID
访问
中间钥
根据
中间钥
分割数据
访问
解析器
主钥
根据
解析器主钥
分割事务ID
解析
加密后
的数据
份1
事务
ID
份1
加密后
的数据
份2
事务
ID
份2
加密后
的数据
份3
事务
ID
份3
加密后
的数据
份n
事务
ID
份n
生成
第1份钥匙
生成
第2份钥匙
生成
第3份钥匙
生成
第n份钥匙
加密
片段
利用第1份钥匙
加密第1份数据
利用第2份钥匙
加密第2份数据
利用第3份钥匙
加密第3份数据
利用第n份钥匙
加密第n份数据
加密后的第1份
(数据/事务ID)
钥匙n
加密后的第3份
(数据/事务ID)
钥匙2
打乱
加密后的第2份
(数据/事务ID)
钥匙1
加密后的第n份
(数据/事务ID)
钥匙3

图 24

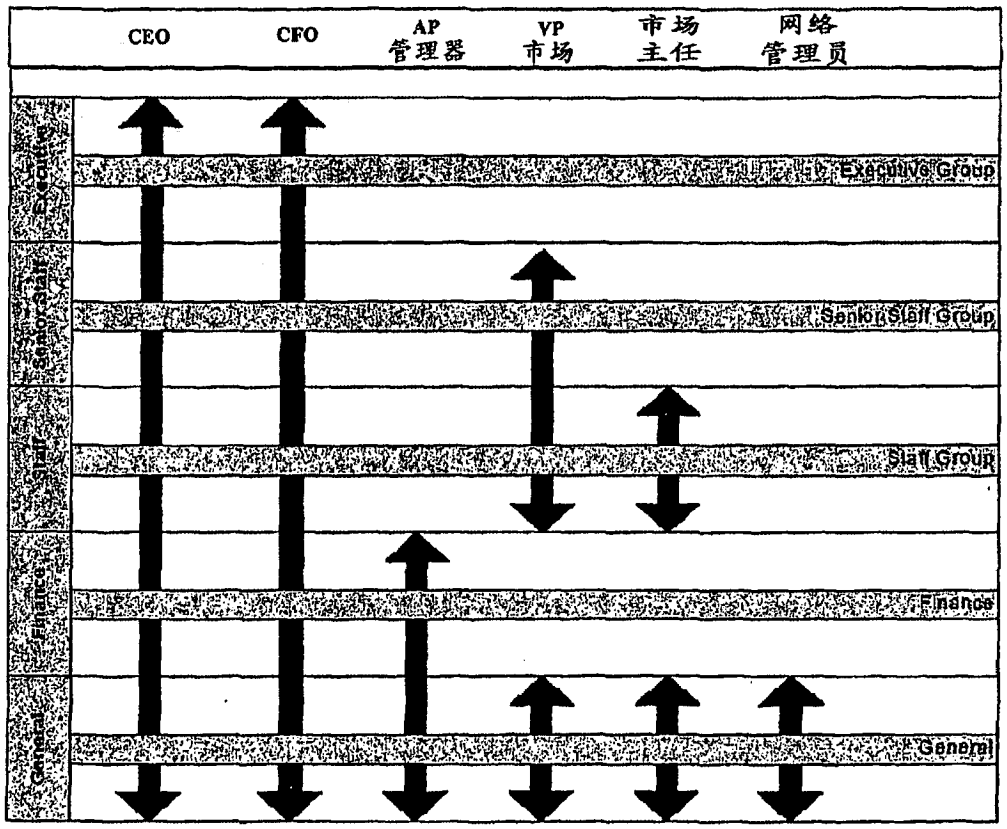

CEO
CFO
AP
管理器
VP
市场
市场
主任
网络
管理员
Executive
Executive Group
Senior Staff
Senior Staff Group
Staff
Staff Group
Finance
Finance
General
General

图 25