

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
H04L 9/00 (2006.01)



[12] 发明专利申请公布说明书

[21] 申请号 200780003321.5

[43] 公开日 2009 年 12 月 9 日

[11] 公开号 CN 101601222A

[22] 申请日 2007.1.25

[21] 申请号 200780003321.5

[30] 优先权

[32] 2006.1.25 [33] US [31] 11/340,376

[86] 国际申请 PCT/US2007/001899 2007.1.25

[87] 国际公布 WO2007/087352 英 2007.8.2

[85] 进入国家阶段日期 2008.7.23

[71] 申请人 甲骨文国际公司

地址 美国加利福尼亚州

[72] 发明人 托马斯·埃曼努埃尔·瓦赫瑟

史蒂文·卢卡斯·哈里斯

乔恩·布赖恩·费希尔

唐·布斯科·迪瑞

[74] 专利代理机构 北京东方亿思知识产权代理有限公司

代理人 宋鹤

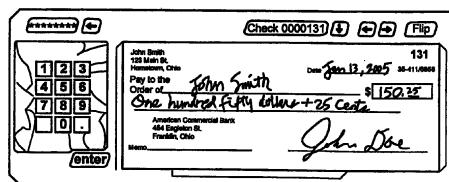
权利要求书 3 页 说明书 23 页 附图 34 页

[54] 发明名称

在线数据加密和解密

[57] 摘要

本发明提供了用于在输入到用户计算机中时对在由计算机实现的网络上传输的数据(优选为用户认证标识符数据,例如口令)提供加密和解密的系统和方法。这些系统和方法使得末端用户能够从图形图像的第一部分上随机安排的元素之一中选择标记。图形图像的第二部分包括任何个别认证标识符序列的可能元素的安排,并且被置于第一部分附近。这些系统和方法提示用户通过根据需要移动所选标记和第一部分以将所选标记与出现在较外部分上的认证标识符的所选元素基本上对齐来输入标识符的每个元素。根据一个实施例,图像部分是同心轮盘。根据另一实施例,图像部分被安排成相邻的行。



1. 一种由计算机实现的认证界面系统，包括：
连接到计算机网络的服务器；
连接到所述计算机网络的用户计算机；
所述用户计算机和所述服务器被编程为经由所述计算机网络向彼此发送信息和从彼此接收信息；以及
图形界面，该图形界面包括组合图像和输入界面，该组合图像包括与用户名相关的特定于用户的图像；
其中所述服务器被编程为响应于从所述用户计算机接收到的用户名向所述用户计算机发送所述图形界面，并且所述用户计算机被编程为接收和显示由所述服务器发送的所述图形界面。
2. 如权利要求 1 所述的认证界面系统，其中所述输入界面被叠加在所述特定于用户的图像上。
3. 如权利要求 1 所述的认证界面系统，其中所述图形界面还包括特定于用户的文本元素。
4. 如权利要求 1 所述的认证界面系统，其中所述输入界面采取小键盘或键盘的形式。
5. 如权利要求 3 所述的认证界面系统，其中所述特定于用户的文本元素由预先选择的、用户定义的文本短语组成。
6. 如权利要求 1 所述的认证界面系统，其中所述特定于用户的图像包括动画元素或照片。
7. 如权利要求 1 所述的认证界面系统，其中所述图像是由所述服务器加密的校验和。
8. 如权利要求 1 所述的认证界面系统，其中所述输入界面被所述服务器进行移位加密。
9. 如权利要求 1 所述的认证界面系统，其中所述用户计算机被编程为接收和显示用户名界面，并且向所述服务器发送由用户输入到所述用户名界面中的用户名，所述服务器被编程为将所述用户名与预先定义的用户计

算机列表关联起来，并且所述服务器被编程为在所述用户计算机的身份与所述预先定义的列表上列出的计算机之一相匹配的情况下向所述用户计算机发送个性化的图形界面。

10. 如权利要求 9 所述的认证界面系统，其中所述服务器被编程为在所述用户计算机的身份不与所述预先定义的用户计算机列表上列出的用户计算机之一的身份相匹配的情况下向所述用户计算机发送第二界面，并且所述第二界面从所述用户接受标识符以认证所述用户的身份。

11. 如权利要求 10 所述的认证界面系统，其中所述第二界面被利用与所述用户名相关的特定于用户的图像进行了个性化，以验证所述服务器的身份。

12. 如权利要求 1 所述的认证界面系统，其中所述图形界面还包括从数据库取得的文档的图像，该文档的图像是响应于由所述用户输入到所述图形界面中的特定标识符而被显示的。

13. 如权利要求 12 所述的认证界面系统，其中：

所述图像的至少一部分在标识符输入之前可见；并且
所述文档的模糊部分响应于所述特定标识符而变得可见。

14. 如权利要求 13 所述的认证界面系统，其中：

所述输入界面采取键盘的形式；并且
所述图形界面还包括由预先选择的、用户定义的文本短语组成的特定于用户的文本元素。

15. 一种用户和保存用户的信息的实体之间的由计算机实现的认证方法，而所述实体在提供对所述信息的用户访问之前必须认证所述用户的身份，而所述实体向所述用户提供图形界面，该图形界面包括组合图像和输入界面，该组合图像包括与所述用户相关的特定于用户的图像，从而所述用户在尝试访问所述信息之前可以认证所述实体。

16. 如权利要求 15 所述的方法，其中所述用户仅在认证了提供所述组合图像的所述实体的身份之后才输入标识符以访问所述信息。

17. 如权利要求 15 所述的方法，其中用户在接收所述图形界面之前向所述实体提供用户名。

18. 如权利要求 15 所述的方法，其中所述图形界面还包括特定于用户的文本元素。

19. 如权利要求 15 所述的方法，其中所述图形界面被所述实体加密。

20. 如权利要求 17 所述的方法，其中所述用户提供标识用于访问所述信息的特定计算机的标识符，并且所述实体仅在将标识所述特定计算机的所述标识符与预先定义的与所述用户相关联的计算机的列表上列出的计算机相匹配之后才提供所述图形界面。

21. 一种用户和保存用户的信息的实体之间的由计算机实现的认证方法，而所述实体在提供对所述信息的用户访问之前必须认证所述用户的身份，并且所述实体向所述用户提供包括输入界面的图形界面以认证所述用户的身份，其中改进包括：

所述实体向所述用户提供与用户相关的预先选择的、用户定义的图像的组合以及输入界面，从而所述用户仅在所述图像对应于预先指定的图像的情况下才向所述输入界面中输入用户标识符。

22. 如权利要求 21 所述的由计算机实现的认证方法，其中仅在所述用户尝试从预先选择的、用户定义的计算机访问所述信息的情况下，所述实体才提供所述组合的预先选择的、用户定义的图像和输入界面。

23. 如权利要求 22 所述的由计算机实现的认证方法，其中如果所述用户尝试从除预先选择的、用户定义的计算机之外的计算机访问所述信息，则所述实体向所述用户提供质询界面。

在线数据加密和解密

本专利文献的公开内容的一部分包含受著作权保护的素材。著作权所有人不反对任何对专利文献或专利公开以其在专利文件记录中的形式进行复制，但除此之外保留一切著作权权利。

技术领域

本发明一般地涉及用于要在因特网或其他网络上进行的交易提供包括认证在内的加密和解密的系统和方法，并且涉及适合用于这种系统和方法中的用户界面。

背景技术

由企业和个人在因特网上进行的在线交易量的增长已经十分惊人。敏感的私人身份信息一般用于认证用户以便进行在线交易。伴随着身份信息在因特网交易中的越来越多的使用的，是该信息被截取和盗用的越来越大的危险。当某人未经同意使用另一个人的口令、用户名、社会安全号码、信用卡号码或者其他标识性个人信息来进行欺骗时，就会发生身份盗用。根据 2003 年 9 月的联邦贸易委员会 (FTC) 调查，在最近 5 年中有 2730 万美国人成为身份盗用的受害者，其中包括在 2002 年一年中就有 990 万人。根据 FTC 的调查，在 2002 年中商业和金融机构的身份盗用损失总共达到近乎 480 亿美元，并且消费者受害者报告了 50 亿美元的预算外支出。

身份盗用的一种形式是利用被称为“网络钓鱼 (phishing)”的社交工程攻击来进行的。网络钓鱼根据因特网百科全书 Wikipedia 被定义为通过伪装成某个确实需要诸如口令和信用卡细节之类的敏感个人信息的可信者，从而通过诡计来欺骗性地获取这种信息。网络钓鱼欺骗方案一般使用伪装网站来生成电子邮件消息，这种电子邮件消息看起来就像是源自受信

任的服务提供者（例如银行或商家）的对必要信息的请求一样。网络钓鱼电子邮件消息一般包含去往看起来属于服务提供者但实际上却为“网络钓鱼者”所用的网站的链接。伪装网站尝试诱骗用户给出他们的口令或其他敏感个人信息。

另一种形式的身份盗用是利用被称为“网址嫁接（pharming）”的攻击来进行的。在这类攻击中，完全只用于犯罪目的的软件一般通过 DNS 劫持或投毒，来将无辜的用户误导到欺骗性站点或代理服务器。网址嫁接利用了 DNS 服务器软件的弱点，该弱点使得黑客可以获取站点的域名，并且使定向到该网站的流量被重定向到另一网站。DNS 服务器是负责将因特网名称解析成为其真实地址的机器；它们充当因特网的“路标”。如果接收到被重定向的流量的网站是伪造的网站，例如银行网站的拷贝，那么它就可以被用于“钓取”或窃取计算机用户的口令、PIN 号码、账户号码和/或其他机密信息。

已知各种其他的用来获取用户输入的机密信息的欺骗性手段。例如，包括键盘记录器、鼠标点击记录器和屏幕捕捉记录器在内的间谍软件是公知的并且被用于此目的。另外，诸如窥探软件、密探软件、非病毒恶意软件、黑客工具、监视工具和木马之类的其他类型的间谍软件是公知的。又例如，“邪恶双子星”攻击正变得常见起来。邪恶双子星是自制的无线接入点，也被称为“热点”，它伪装成合法的接入点，以在末端用户不知晓的情况下收集个人或团体信息。攻击者将自身定位在接入点附近，并且令其计算机发现合法的接入点使用的是什么名称和无线电频率。然后他利用相同的名称在该频率上发出他自己的无线电信号。就本发明而言，间谍软件是任何帮助未经授权地获取信息的软件程序，所述信息例如是个人或组织的信息。间谍软件一般还对用户隐藏。间谍软件一般未经同意就将自身安装在用户的计算机上，然后监视或控制对设备的使用。每当用户按键时，所有的聊天交谈、所有被访问的网站、用户利用浏览器进行的每次交互、每一个被执行的应用、每一个被打印的文档以及所有文本和图像都可能被间谍软件所捕捉。间谍软件一般能够在本地保存捕捉到的数据，并且/或者经由因特网将其发送到第三方，这常常是在用户不知晓或者未经用户

同意的情况下进行的。键盘记录器和鼠标点击记录器还可以采取连接在键盘/鼠标线缆与计算机之间的硬件或者键盘/鼠标设备内的硬件的形式。

另一种获取机密敏感个人信息并用其来进行欺骗的获取者被称为“越肩”（over-the-shoulder）密探。该密探暗中读取用户的显示屏以获取信息，例如字母数字或其他形式的信息。例如，将小键盘和/或键盘图像用于用户数据输入的传统图形用户界面也很易受鼠标点击记录器、屏幕捕捉记录器和其他方案的攻击。图形界面中的每个字母数字字符由一个唯一的图形图像来表示，例如构成数字 1 的像素。屏幕捕捉记录器利用光学字符识别（OCR）技术来解译鼠标点击和相应的字母数字图形，以确定用户的 ID 和口令的实际字母数字文本字符。先进的屏幕捕捉记录器还能够利用图形图像的校验和以及大小特性来确定与数据输入期间用户的鼠标点击所选择的每个图形图像相对应的是哪个标识符字母数字字符。这样，即使当图形用户界面重新安排了小键盘或键盘上的字母数字字符的顺序时，屏幕捕捉记录器也可以获取个人信息。

已知的反病毒和反密探软件的软件产品尝试使得用户能够对抗某些身份盗用者。但是，这些产品不能提供对盗用的安全防护，因为它们本质上是反应性的。这样，它们都依赖于可再现的签名。它们必须被不断地更新，并且只在其被更新的范围内有用。它们总是易受新病毒或新形式攻击的影响。因此，对于防护计算机数据以对抗外界威胁来说，使用过时的反病毒和反密探软件文件最多只会提供最低限度的保护。因此，这些产品的缺点在于反病毒和反密探软件程序所使用的信息必须被不断更新以反映新发现的方案。除了保持病毒信息最新之外，系统还必须周期性地进行扫描以寻找潜在的感染。

防火墙软件提供了可供用户使用的另一条防护线。防火墙软件被安装在用户的计算机上（个人或团体防火墙），以便如果用户的计算机中的程序正在用户不知晓或者未经用户同意的情况下访问网络，则警告用户。但是，如果木马损害了经授权的程序和端口，则防火墙会允许木马通过该端口传输数据。

传输控制协议/因特网协议（TCP/IP）是因特网和一些私有网络的基本

通信协议。安全超文本传送协议（HTTPS）是基于 TCP/IP 的安全因特网通信协议，其使用安全套接层（SSL）协议来允许使用加密数据流的安全数据传送。HTTPS 的主要目的在于以安全方式从远程主机取得超文本对象，例如网页。SSL 协议允许了包括 web 浏览器和 HTTP 服务器在内的客户端经由安全连接进行通信。SSL 提供了加密、源认证和数据完整性，作为用于对经由不安全的公共网络交换的信息进行保护的手段。许多电子商务应用将这些协议用于保护服务器和因特网之间的传输。

已知系统的另一缺点在于它们在不同程度上依赖于人类来维护它们的安全状态。如上所述，尽管采取了安全预防措施，用户的信息和/或系统还是可能受到损害。解决身份盗用问题的其他已知方法包括要求用户使用“令牌”或者在用户的系统上保存数字证书以用于登录过程期间的认证。令牌一般是为了访问服务提供者的系统而需要的像信用卡或钥匙圈那样大小的认证设备。令牌通常显示随着时间而改变的数字，并且与网络上的认证服务器同步。令牌还可与服务器使用质询/响应方案。该方法要求用户拥有令牌，并且除了输入口令和/或个人标识号码（PIN）外，还要输入来自令牌的信息以便进行认证。令牌的问题在于用户除了要保持所需的口令和/或 PIN 安全之外，还必须保持令牌安全。另外，与令牌丢失或损坏相关联的客户支持成本也造成了额外的问题。因此，也需要一种用于防止身份盗用的系统和方法，其不要求创建和维护成本高昂的硬件设备来为机密信息提供安全性。

发明内容

根据本发明的系统和方法通过在将数据输入到用户计算机时提供能够加密用户认证证书的用户界面，以及在由计算机实现的网络上提供对可由诸如字母数字之类的符号、字处理或其他软件一般提供的其他符号和能够在这种网络上或经由这种网络被处理的任何其他符号所表示的几乎任何数据的加密和解密，从而克服了已知系统和方法的缺陷。

根据本发明的系统和方法的一个优点在于它们允许了用户利用服务器提供的由用户定义的图形界面来认证被访问的服务器的身份。尝试登录到

其账户的用户从而能够验证看起来源自特定服务提供者的网站或消息确实是真实的服务提供者。敏感的个人信息从而不会被用户提供给伪装为实际提供者的欺骗性实体。

另一个优点在于这些系统和方法不依赖于令牌、卡和其他类似的硬件设备、数字证书、反病毒软件或者个人防火墙解决方案来保护末端用户以对抗在线身份盗用。

本发明的认证界面系统的一个实施例可包括：连接到计算机网络的服务器，连接到计算机网络的用户计算机，用户计算机和服务器被编程为经由计算机网络向彼此发送信息和从彼此接收信息；以及图形界面，该图形界面包括组合图像和输入界面，该组合图像包括与用户名相关的特定于用户的图像，其中服务器被编程为响应于从用户计算机接收到的用户名向用户计算机发送图形界面，并且用户计算机被编程为接收和显示由服务器发送的图形界面。输入界面可采取小键盘或键盘的形式，并且可被叠加在特定于用户的图像上。图形界面还可包括个性化元素，例如特定于用户的文本元素、动画元素和照片。

用户和保存用户的信息的实体之间的认证方法的一个实施例包括实体向用户提供图形界面，该图形界面包括组合图像和输入界面，该组合图像包括与用户相关的特定于用户的图像，从而用户在尝试访问信息之前可以认证实体。图形界面可在用户输入用户名或者要访问的信息的身份的某种其他指示之后被提供给用户。用户仅在认证了提供组合图像的实体的身份之后才输入标识符以访问信息。图形界面还可包括特定于用户的文本元素，并且可被实体加密。另外，实体可以仅在将标识特定计算机的标识符与预先定义的与用户相关联的计算机的列表上列出的计算机相匹配之后才提供图形界面。

参考以下描述、所附权利要求书和附图，将会更好地理解本发明的这些和其他实施例、特征、方面和优点。

附图说明

通过结合附图参考以下详细描述，将更容易理解本发明的以上优点和

特征以及其伴随而来的优点，附图中：

图 1 是图示出用于输入用户认证的示例性现有技术系统的示图；

图 2 图示出用于使得能够输入认证信息的示例性现有技术小键盘图形用户界面；

图 3 图示出用于使得能够输入认证信息的示例性现有技术键盘图形用户界面；

图 4 是图示出使得用户能够在经由网络连接到计算机/服务器的用户计算机上输入信息的典型现有技术系统的框图；

图 5 图示出用来帮助说明图 6-10 所示的本发明实施例的新颖特征的现有技术键盘图像；

图 6 图示出根据本发明一个实施例的优选的安全性更高的扭曲型键盘图形认证界面；

图 7 图示出根据本发明替换实施例的优选的安全性更高的扭曲型键盘图形认证界面；

图 8 图示出根据本发明替换实施例的优选的安全性更高的扭曲型键盘图形认证界面；

图 9 图示出根据本发明替换实施例的优选的安全性更高的文件大小型键盘图形认证界面；

图 10 图示出根据本发明替换实施例的优选的安全性更高的文件大小型键盘图形认证界面；

图 11 图示出根据本发明替换实施例的优选的安全性更高的散列型键盘图形认证界面；

图 12 图示出根据本发明替换实施例的优选的安全性更高的散列型键盘图形认证界面；

图 13 图示出用来帮助说明图 14-18 所示的本发明实施例的新颖特征的现有技术键盘图像；

图 14 图示出根据本发明替换实施例的优选的安全性更高的移位型 (shift type) 键盘图形认证界面；

图 15 图示出根据本发明替换实施例的优选的安全性更高的移位型键

盘图形认证界面；

图 16 图示出根据本发明替换实施例的优选的安全性更高的移位型键盘图形认证界面；

图 17 图示出根据本发明替换实施例的优选的安全性更高的校验和型键盘图形认证界面；

图 18 图示出根据本发明替换实施例的优选的安全性更高的校验和型（checksum type）键盘图形认证界面；

图 19 是图示出用于图 6-18 的实施例的用户端和服务器端的优选加密和解密过程的框图；

图 20 图示出根据本发明替换实施例的优选的安全性更高的动态图形轮盘型（wheel-type）键盘图形认证界面；

图 21 图示出根据本发明实施例的优选的安全性更高的动态滑块型键盘图形认证界面；

图 22 是图示出用于本发明的非用户个性化标记（marker）的优选加密和解密过程的框图；

图 23 是图示出用于本发明的用户个性化标记的优选加密和解密过程的框图；

图 24 是图示出用于本发明实施例中的加密和解密的优选客户端过程流的框图；

图 25 是图示出用于认证过程的现有技术客户端/服务器交互的框图；

图 26 是图示出用于本发明实施例中的具有加密的优选客户端/服务器交互的框图；

图 27 是图示出认证过程的框图；

图 28 是图示出用于本发明实施例中的优选认证过程的框图；

图 29 是用于加密和解密过程的优选实现方式的伪源代码列表；

图 30 是用于本发明的图形的优选实现方式的伪源代码列表；

图 31A-E 图示出根据本发明替换实施例的优选图形认证界面。

参考符号或名称在附图中被用来指示这里示出的某些组件、方面或者特征。多个图中共有的参考符号指示这里示出的类似组件、方面或者特

征。

具体实施方式

根据本发明的实施例，在不迟于将用于因特网或其他网络上的交易的诸如用户认证信息和/或代表其他信息的数据之类的信息或数据输入到用户的计算机时对信息或数据进行的加密和解密是通过实时的图像处理过程来实现的，并且/或者是通过预先创建要被实时地、无设定顺序地随机使用的图像，以使得通过界面进行的数据输入几乎不可预测因此几乎对任何非法的图像解码尝试免疫，从而来实现的。因此，信息或数据不易受到任何这种网络上的盗用的攻击。参考图 1-28 和 31 A-E，将对比现有技术方法和系统来描述本发明的加密和解密系统和方法的若干实施例。正如将会说明的，本发明的最优先实施例用于认证用户，以便可以保证因特网或其他网络交易的安全性。但是，本发明具有宽广得多的范围，并且可用于对容易用符号来表示并且将在计算机实现的网络上传输的信息进行加密和解密。

为了在因特网或其他计算机网络上进行在线交易，用户一般使用键盘、鼠标或其他输入设备，来利用连接到因特网或其他计算机网络的 web 浏览器输入其敏感个人信息。图 1 是图示出用于基于唯一用户名（用户 ID）和口令来验证用户的证书的示例性传统系统 20 和认证过程。在此示例中，用户将要输入的认证信息包括用户 ID 和口令，其中每一个包括若干个元素。就本发明的各种实施例而言，术语“标识符”可以指几乎任何与用户知晓的信息和/或用户拥有或与用户相关的某种属性相关的信息。例如，这种标识符可包括名称、账户号码、社会安全号码、地址、口令、个人标识号码（PIN）。另外，就本发明的各种实施例而言，在标识符的上下文中使用的术语“元素”可以是几乎任何可被系统识别的符号。一般来说，为了用作用户 ID 和/或口令，元素优选是以特定顺序表述的字母数字符号。一般来说，用户 ID 和口令是由当在计算设备 24 上运行 web 浏览器时经由键盘 22 输入的字符串组成的。浏览器在如 28 处所示的显示屏上向用户提供典型的用户输入界面 26。或者，用户的数据输入可经由数字小键盘 30 的图形图像上的鼠标点击来进行，如图 2 所示，或者在键盘 32 的图

像上进行，如图 3 所示。图 2 是数字界面的典型表示，该数字界面可被末端用户使用，以通过点击界面上的适当位置来输入口令/代码/PIN。该界面只允许输入字母数字信息，但是小键盘也可被修改，以提供其他符号或图标。图 3 是字母数字界面的典型表示，该字母数字界面可被末端用户使用，以通过点击该界面输入口令/代码/PIN（在此实例中只是字母数字，或者也可以是其他符号/图标）。

在本发明的优选实施例中，为尝试获得对提供者的服务（例如用户在银行的账户）的访问权限的用户具体地个性化界面。虽然银行的总网站可以具有针对所有用户的标准格式，但是一旦特定用户识别了他的要被访问的账户，认证过程就开始，以允许用户和银行都能确认彼此的身份。用户首先将用户名输入到由服务提供者发送并被显示在用户计算机上的标准界面中。响应于用户名，服务提供者随后发送特定于具体用户的个性化认证界面，并且它被显示在用户的计算机上。

图 31A 图示出个性化的图形界面 160，其中采取传统小键盘 163 和数据窗口 162 形式的输入界面被叠加在雏菊田野的背景图像 161 上。输入界面可包括如图 31A 所示的小键盘 163、图 31C 的键盘 164、其变体，或者这里论述的其他输入设备中的任何一种的表示。输入界面可以以叠加、嵌入、结合、邻近或其他方式与用户先前选择的图像相关联。图 31B 图示出具有输入界面的图形界面的实施例，该输入界面仅由结合了猫的图像 161 的数据窗口 162 构成。在此实例中，数据是由用户经由另一输入设备（例如计算机的键盘）输入的。图 31D 图示出一个实施例，其中输入界面由个性化问题 167 和多个潜在的答案 168 构成。可以设想，输入界面的确切形式可以不同于输入设备的传统物理实施例，而是实际上可以由图像本身元素构成。

个性化属性可以由图形界面上显示的颜色/形状/实际数据构成。个性化属性也可能在于输入界面本身的表示中。图像 161 例如可以是照片、图表、图画或其他可视的表现图。图像还可包括图案、几何形状、动画元素或帮助区分图像的其他可视标记，从而为界面提供了用户可识别的平台。文本元素 165 也可以被设置在图像之中、图像周围或以其他方式与图像相

关联，以进一步个性化界面。这些文件元素可以采取单独的字母、单词、短语或特定于用户的其他文本分组的形式。

另外，图像和界面本身可利用这里描述的方法被加密，以阻止对界面的捕捉和欺骗性再建。

个性化属性优选地是由用户在以下时间选择的：当在设立用户的账户时用户和服务提供者之间的关系首次建立之时，或者至少在用户首次尝试通过服务提供者访问账户之前。在此，用户可以从选择列表或从自己的优选项中指定将被结合到图形界面中的个性化元素。因此，一旦被创建，个性化的图形界面就将立即能被用户识别，并且几乎不可能被尝试猜测许多可能的变体之中的哪一种被结合到了界面中的欺骗性实体所再建。

界面的个性化使得用户可以知道，界面不是经由自动地向用户大规模分发假冒界面（在诸如“网络钓鱼”之类的计算机攻击中就可能尝试这么做）来无智能地创建的。在网络钓鱼中，欺骗性实体尝试再建真正的界面，以期引诱没有疑心的用户透露敏感或机密的信息。但是，通过使真实界面个性化，在假冒界面被显示时，用户就被警告可能存在该界面。因为网络钓鱼者将会很难或无法复制这种个性化界面，这有助于阻挠网络钓鱼，从而帮助了赢得末端用户的依赖。这种应用被称为相互认证过程。

对抗网络钓鱼和其他攻击的进一步保护可通过用户名与用于访问合法服务器的特定计算机的初始相关来提供。尝试从除了预先识别的特定计算机之外的计算机输入用户名将会导致质询界面的呈现，该质询界面向尝试访问用户账户的人提出质询问题。例如，如图 31D 所示，质询界面 166 可提出一个或多个问题 167，这些问题的答案将只为合法用户所知晓。成功的回答质询问题将使得用户可以访问个性化认证界面，而未能回答则将会使认证过程终止。也可利用这里描述的方法来使质询界面个性化，以向系统添加额外的安全性。

如果用户需要从若干个不同的计算机访问合法服务器，则用户可以向合法服务器提供将被用来访问服务器的计算机的列表。从而，只有从未列出的计算机访问用户账户的那些尝试会导致呈现质询界面。

设想了这些个性化属性可结合不同类型的界面来实现，以提供对抗迄

迄今为止描述的计算机攻击的保护，并且不限于具体描述的实施例。可以利用个性化属性来修改这里描述的界面实施例中的任何一种。例如，对如图 31B 所示的具有标准文本输入框的界面的个性化将会向这些类型的界面提供保护。允许对归档的文档进行在线检索的另一类界面也可以这种方式被个性化，以便作为对来源的核实以及只允许特定用户访问。图 31E 公开了允许检索归档的文档 169 的界面 168 可在用户认证之前隐藏其一些或所有特征。

图 4 是图示出典型的现有技术系统 39 的框图，该系统包括用户计算机 40 和在计算机 40 上输入信息的用户 42。计算机或互联网都不是在考虑到安全性的情况下设计的。安全性只是事后的想法，用户利用其计算机在互联网上进行的典型交易内的示例性的不同缺点和可能的数据弱点被示出。计算机 40 经由网络连接到计算机/服务器 49。如图 4 所示，敏感性信息在经由网络从用户的计算机系统 39 传输到远程服务器 49 之前，可以利用例如 HTTPS 在 47 处被加密。但是，系统 39 和计算机 40 易受信息盗用的攻击，因为在输入到计算机 40 中的点和调用加密过程的点之间，信息保持其原始形式。就本发明而言，输入点和加密点之间的所有点都被总称为漏洞 41。如图 4 中示意性所示，机密数据由用户在 42 处创建，并且以未加密的形式在 41A 处通过 IO 设备 43 被输入到用户的计算机系统中，然后在 41B 处经由 45 处的 CPU 和内核以及支持芯片流到操作系统 (OS) 44 中，然后在 41C 处流到应用 46。传出的、未经加密的数据随后在 41D 处流动，其中它在 47 处被加密，并且经由路径 41E、41F 被传递到 OS 44 和 I/O 设备 43，然后在 41G 处被计算机 40 利用如 48 中所示的路由器或其他网络工具经由路径 41H 传输到服务器 49。由于以上所述的示例性的特定漏洞 41A-41H 以及网络弱点，诸如交叉站点脚本程序、键盘记录器、鼠标点击记录器、屏幕捕捉器和中间软件中的人之类的威胁可以以敏感性信息的原始的、加密前的形式捕捉到敏感性信息。从而，即使离开用户计算机系统的数据已被加密，网络也可能受到损害。这是因为在漏洞中的任何一处，例如在沿着位置 41A-41H 所示的数据流上的任何一点，都可能绕过或损害加密协议。本发明的实施例提供了用于使得能够输入诸如用户认证证

书之类的数据的系统和方法，所述数据在不迟于输入点处对认证信息进行加密，从而填补了漏洞。

图 5 图示出用于描述本发明的若干个实施例的传统图形小键盘 52。图 6-8 图示出优选的安全小键盘图形界面 54、56、56，该界面被配置用来通过包括用于数据输入选择的小键盘的扭曲，来提供比传统上更高的安全性。此实施例被称为扭曲型，这种类型被称为“图像扭曲”，因为与图 5 的小键盘界面 52 相比，用户的小键盘已被扭曲了。这种扭曲使得人类用户可以很容易地识别图像中的数字或其他符号，但却防止了屏幕捕捉器/OCR 和 x-y 坐标记录器将鼠标或其他指点设备的点击链接到界面的特定键。虽然图 6 示出了一种示例性扭曲，但实际上可以生成几乎无穷多种扭曲排列和组合，以在 X 轴和 Y 轴上使窗口界限内的数字、字母或其他符号的图像扭曲，从而降低对图像进行未经授权的解码的可能性。例如，图 7 图示出一种小键盘 56，该小键盘 56 已经被扭曲，从而与图 5 的现有技术小键盘 52 中所示的空间关系相比，提供或显示了小键盘上的数字和特征相对于彼此的不同空间关系。在图 8 中，示出了另一个扭曲的图形小键盘 58 界面。在此实施例中，小键盘 58 的背景特征由虚线表示，以表明采用了与图 5 的传统小键盘中采用的不同的颜色或灰度阴影。成功盗用信息的基础在于能够捕捉屏幕显示并随后将其用于预测未来的信息输入。当服务器每次认证会话开始都向客户端发送不同的图像时，使用捕捉的信息作为预测未来行为的基础就变得很困难了。对于图 6-8 的实施例来说，扭曲图像可以使用诸如纹理/扭曲/噪声/像素/等等滤波器之类的各种传统数学算法来进行。这些图像随后可在服务器上被实时随机选取，然后被显示给末端用户。各种算法可以被实时应用或者预先应用到图像，并被存储在数据库中。

图 9 和 10 图示出其他类型的有用的图形界面以及本发明的优选替换实施例。在图 9 中，界面 60 被示为具有灰色背景 62，用于围绕着各自包含数字的键的小键盘表面。在此类实施例中，界面 60 被加了阴影，并且阴影的程度是由一个或多个随机图像处理算法来提供的。这样，提供了大量可能的小键盘排列和组合。任何图形图像的文件的实际大小都完全是图

像的分辨率或图像内表示的像素/英寸的函数。这些也决定了图像的质量。服务器随后可以基本上随机地将额外的值填充到同一图像，以生成不同的文件大小，这一点进而又不能被欺骗性实体用来准确地识别被显示给末端用户的图像，因为相同的视觉上类似的图像的文件大小可能每次是不同的。

也很明显的是，这类实施例并不限于结合小键盘使用。而是可以使用键盘或其他类型的界面。另外，在这里论述（但未示出）的这种类型和其他类型的界面实施例中，可在计算机屏幕内使 X 轴和/或 Y 轴位移一个较小的量。这种位移使得数据记录器型软件更加难以准确地捕捉屏幕上示出的数据，而这种数据很容易被用户和用户的计算机经由网络所连接到的合法服务器所理解。参考图 10，图示出了加密/解密的另一种阴影类型。小键盘 64 具有背景 66，该背景 66 被示为不同于图 9 或图 5 的小键盘的背景中的任何一种。

这里描述的在本发明中使用的计算机屏幕图像的位移、背景改变、抖动和扭曲可由传统的编程技术来生成。与用户和合法服务器所知晓的输入数据的图像在传统图像显示屏上的外观相比，这些位移、背景改变、抖动和扭曲能够改变出现在用户屏幕上的输入数据的图像的空间关系。这些空间关系改变优选为较小，即都保持在主窗口之内，并且优选为随机的，正如下文将描述的。这样，这些空间关系改变足够神秘，从而能够阻止计算机程序对加密后的数据进行解码。

参考图 11 和 12，将描述本发明的另一类图形界面实施例。这些图像描绘出了响应于用户对数字的点击而经由网络发送的实际数据。这些值是由服务器实时生成的，然后与图像一起发送给客户端。在解释点击后，客户端随后向服务器发回预先指派的数据。服务器基于预先存储的值来很容易地识别相应的图像。从而，这两个图图示出了本发明的散列型加密、解密。在图 11 的左侧，显示了传统的小键盘图像 68。在右侧，在虚线中示出了被散列的显示 70，其中每个键具有按随机顺序安排的若干个字母。在此实施例中，服务器被用于向客户端发送映射指令，从而当例如用户输入“0”时，客户端将“0”映射到“ej”，并且将“ej”发送到服务器。对

于每个认证会话，服务器发送不同的一组映射指令，从而对于每个认证会话，发生真正数据的完全不同的映射和发送。图 12 通过在左侧示出同一传统小键盘图像 68，但利用另一不同的被散列的显示 72 来表示由服务器发送到用户的客户端计算机的不同的一组映射指令，从而图示出了此特征。被散列的显示 72 对于每个键具有不同的、优选为随机的一组字母。可以清楚看出，其他符号可用于被散列的小键盘。同样，由于在每个认证会话期间使用的随机的、不同的映射，对于用户的安全性信息的盗用几乎是不可能的。

参考图 13-16，示出了使用移位型加密/解密的本发明的其他优选实施例。这些图像表示在更大的外部背景内使数字界面的 x 和 y 值位移的效果。x 和 y 值被随机地“抖动”或“调整”一有限值，从而净效果是鼠标位置的 x, y 坐标的值在被捕捉到时无法很容易地被用于推断/识别末端用户所点击的确切数字。例如，在图 13 中，现有技术小键盘 74 被示出作为图 14-16 的参考点。在图 14 中，服务器将所示出的小键盘（在 78 处的虚线中）发送到客户端计算机的在 76 处的实线中所示的位置。对于每个认证会话，使用一个不同的映射算法，从而，用户的标识符输入无法被很容易地再现。在此图中，所映射的小键盘被示为实线小键盘图像显示 76，该显示被示为相对于客户端计算机上显示的小键盘位置 78 向下和向右移位。在对于一不同的认证会话表示一不同的映射的图 15 中，在服务器上创建的实线小键盘图像 80 相对于客户端计算机显示屏上的小键盘图像 82 已经被向右和向上移位。在图 16 中，在服务器上创建的所映射的实线图像 84 已经被向客户端计算机显示屏上示出的小键盘图像 86 的下侧和左侧移位。就本发明而言，术语“抖动”被定义为意指图 14-16 所示的那类扭曲，并且该术语被用在诸如“抖动”界面中。

图 17 和 18 图示出了本发明的另一实施例，该实施例被称为校验和型加密/解密。图像中的每个像素具有由其在图像中的位置所决定的“x”和“y”值限定的唯一的 2 维标识符。该图绘出了像素的 RGB 值，这些值用于表示数字界面中的数字 7 内的样本像素。通过取 R、G、B 的唯一值，并将它们加起来，并且将 x, y 值也与这相加，可以发现即使只在像素的

R、G 或 B 值之一中有细微差别，一个图像所表示的总值也可能不等于另一图像。如图 17 所示，小键盘 88 具有键 90，该键 90 承载着以某种预定的颜色示出的数字 7。该数字 7 具有与之相关联的 x 和 y 位置以及颜色的红（R 或 “r”）、绿（G 或 “g”）和蓝（B 或 “b”）值。通过取被指派给 R、G 和 B 中每一个的唯一值并将它们与 x, y 值相加起来，可以确定一个总值来表示该像素。通过对每个像素或者选定数目的像素重复该过程，可以确定图像或图像的一部分的总值。图 17 中的框 92 表示六个相邻的像素，其中不同的阴影示出了五个值中的至少一个的某种差别。如图 17 所示，在框 94 中，“x”值被指派以“70”，所指派的“y”值是“111”，所指派的红或“r”值是“211”，所指派的绿或“g”值是 211，所指派的蓝或“b”值是 211。该像素的总值由“j”表示。相邻的像素被以相同的方式指派了值，如框 96 中所示，唯一地差别在于“x”值改变了“1”得到总“x”值“71”，从而产生了相差 1 的不同总值“w”。类似地，图 18 图示出了小键盘图像 94，并且其“7”位于相同的位置，但是具有不同的“r”、“g”和“b”值，从而总值“j”不同。另外，相邻的像素与图 17 中的相应像素相比具有不同的“r”、“g”和“b”值。因此，即使在像素的 R、G 或 B 值之一上有细微差别，一个图像的文件大小也可能不等于另一图像的文件大小。这些变化也可被应用到灰度级图像或非 RGB 型图像。

对于图 9-10 和 17-18 所示的文件大小和校验和型加密，键盘图像可被扭曲，使得整个图像或者键盘上的每个键的图像将产生不同的校验和和/或文件大小，以避免先进的屏幕捕捉记录器识别出每个键。

图 19 是图示出图 5-18 所示的那些类型的加密和解密以及图 20-21 所示的动态加密/解密方法和系统（下文将描述）的典型实现过程。如图所示，在服务器上和在客户端计算机上，即在用于图 5-18 所示的小键盘和/或键盘图形认证界面实施例的系统的客户端和服务器端，使用单独的过程。从图 19 可见，加密和解密过程在性质上不对称的，因为与加密方相比，在解密方涉及的步骤更少。

优选地，三步过程被用来创建图 5-18 所示的那些类型的安全、唯一键

盘图形认证界面。在第一步骤中，生成加密的键并将其映射到键盘的唯一图形字符。在下一步骤中，在较大的 X、Y 轴的界限内随机地使键盘图形图像位移。在第三步骤中，利用已知的图像处理算法来有限地扭曲图形图像。这些图像被扭曲即加密的级别或程度仅由末端用户在视觉上解译各个键盘键图像的能力所限。优选使以上加密步骤对于界面的每个使用实例唯一，以便随着时间过去而增大解译图像的难度。

在图 19 中可以看出，解密过程优选地包括两个步骤。在第一解密步骤中，用户在视觉上解译被加密的键盘（X、Y 位移和图形扭曲是用于加密的两个步骤），并且选择键盘界面上的键以用于输入认证信息。在第二解密步骤中，在服务器上通过查找确切映射来解密键盘映射。

图 20 图示出根据本发明另一实施例的动态图形轮盘多因素界面，用于使能对（最好是）利用鼠标点击和/或键盘导航输入到计算机系统中以用于对齐字母数字和图形符号的认证信息进行加密/解密。用作轮盘上的标记的颜色/图标/形状/形式也可以基于由末端用户预先确定或者由服务提供者预先决定的逻辑来生成。这使得末端用户能够创建然后识别轮盘，因为是他个性化了轮盘。由于用户实时地选择标记，因此该实施例被称为动态系统和方法。图 20 所示的轮盘图形用户界面（GUI）200 是利用传统技术在服务器上生成的，并且优选地包括两个同心的轮盘 202 和 204，用于在数据输入时创建加密。用户简单地经由对“右箭头”按钮 206 进行导航性鼠标点击和/或使用键盘以便逆时针旋转，以及对“左箭头按钮”208 进行导航性鼠标点击以便顺时针旋转，来将内轮盘 202 上的参考点导引到外轮盘 204 上的用户名字段 210 或口令字段 212 的下一元素，以输入每个数据元素。位于内轮盘 202 上的参考点，也称为标记，是由用户在输入时选择的，并且仅为用户所知晓。因此，用户 ID、口令等等的特定元素的身份对于外人是不可辨别的，包括对于各种间谍软件和“越肩”密探是不可辨别的。换言之，用户最初在其头脑中选择参考点标记，即虚拟标记。用户简单地将内轮盘 202 上的所选参考点/虚拟标记导引到外轮盘 204 上的所选标识符元素，例如用户名字段 210 或口令字段 212，以便输入标识符。标识符在这里也被称为代码或访问代码。标记仅为末端用户所知晓，并且在标

识符的所有元素被输入到系统的会话期间保持恒定。用户从输入代码的第一元素开始，例如用户 ID。然后用户顺次输入代码的每个下一元素。在点击“输入”按钮以输入这样加密的代码元素后，用户随后点击“下一个”按钮。轮盘 202 和 204 上的符号随后优选地被随机化，并且用户随后旋转内轮盘 202，使得内轮盘上的所选符号顺次匹配代码的下一元素或被置于代码的下一元素附近，并且点击“输入”。系统随后向服务器发送与下述实际度数或旋转位移相对应的数据：该实际度数或旋转位移是内轮盘 802 从在第一元素被选择和输入后屏幕显示被随机化时起到轮盘 802 变得静止为止移动的实际度数或旋转位移。换言之，位移信息是以度数形式或者表示在用户选择第二元素时发生的轮盘 802 的位移的某种其他形式被发送到服务器的。点击“下一个”、随机化显示、旋转内轮盘 202 以使所选标记与下一代码元素顺次匹配的这个过程被重复，直到特定标识符的所有代码元素都被输入到系统中为止。

通常由服务器容宿在数据库中的图像符号和标记符号的顺序经由网络以阵列形式被发送到要显示的 GUI。服务器还可被编程以在通过网络发送符号图像之前对其应用前述其他形式的加密中的任何一种。优选地，标记符号的顺序在每次代码的元素被输入到系统中时被随机化，并且这可以通过传统的技术来完成。虽然标识符元素的顺序也可被随机化，但对于大多数应用来说，最好它们不在每次会话期间被随机化。

作为没有图示出的另一实施例，可以使标记和/或数据元素的符号的集合个性化，或者使之唯一并且基于用户的偏好，或者由服务提供者来设置。这种唯一性还确保了用户正使用正确的认证设备/GUI。这种可选的特征几乎消除了相同的、非法的或伪装的 GUI 被发送到末端用户以便用于输入其证书、认证数据或其他代码的可能性。在一种优选实现方式中，通过用户交互来进行的 GUI 上的位移是通过使标记阵列的索引相对于标识符阵列移位来计算或确定的。所得到的每个元素的每个标记索引的位移值随后经由网络被发送到服务器。由于服务器已被编程为知晓正确的代码，因此它随后可以使用与标识符的第一元素的输入相对应的位移来确定用户为该会话选择了哪个标记。服务器随后可通过验证后续的位移仅对应于用户为

该会话选择的标记的位移，来认证每个后续元素。

可选地，如图 20 所示的“输入”按钮可用来指示用户字段 210 或口令字段 212 的所有元素已经被输入。所示出的按钮指示器只是示例性的；其他按钮指示器也可用在本发明的实施例中。或者，在其他实施例中可以去除“输入”按钮。例如，对于其中认证标识符（例如用户名或口令）具有预定的固定长度的系统，可能不需要“输入”按钮。

另外，对于图 20 所示的那类的加密/解密，优选地，所输入的元素不被显示在用户名字段 210 或口令字段 212 中，以用于帮助防止“越肩”密探看到该信息。星号或其他适当的符号可被显示在每个字段中，以表明元素的输入。就图 20 类型的本发明实施例而言，术语“标识符”优选地指用户 ID、口令和/或 PIN。但是，如上所述，该术语可以指用户可能希望加密并输入到系统中的几乎任何信息。例如，这种标识符可以包括名称、账户号码、社会安全号码、地址和电话号码。另外，如上所述，术语“元素”可以是被系统识别的几乎任何符号。一般来说，当在用户 ID 和口令的上下文中使用时，元素是以特定顺序表述的字母数字符号。就本发明的各种实施例而言，术语“标记”也可指被系统识别的几乎任何符号。为了方便，优选地，标记是非字母数字符号。

根据如图 20 所示的那类优选实施例，使服务器用随机化的标记序列来预先填充标记阵列。可选地，可以生成多组随机生成的标记候选，以考虑到要使用的标识符的数目并提供在使用重置按钮的情况下可用的额外的多组，重置按钮在图 20 中示出但未被编号，并且将在下文中参考图 21 来进一步说明。例如，优选地，对于四元素标识符，例如“BANK”，候选标记的组数至少为 20，从而为每个标记提供五个重置。

如上所述表示与标识符的每个元素的输入相关联的位移的值被发送到服务器并被服务器解码。因为服务器知道用户为特定认证信息选取的任何特定标记的正确元素和可能的标记行为，以及图像细节，例如“抖动”、文件大小、校验和、扭曲、移位和这种类型的图像细节的组合，所以服务器基于预期的逻辑来推知标记元素。在这种应用中，图 19 的框图也将适用，但是必须包括涉及并应用这里描述的轮盘类型加密/解密动作的步骤。

用于轮盘型过程的逻辑是知晓预期标识符的第一字母的服务器寻找用户选择来输入第一标识符元素的标记。然后，服务器也知道用于该会话的可能标记。对于标识符的第二和每个后续输入，服务器识别并验证相同的正确标记被使用。从而，系统能够判定用户是否为该会话输入了正确的认证标识符。位移坐标是特定于会话的，并且一旦会话结束就不可使用了。可以意识到，这种加密和解密始终使用在输入到系统时随机生成的唯一的、几乎是防盗用的位移信息。

对于为每个会话创建多组随机化标记的示例，图 20 示出了内轮盘 202 具有最初以随机顺序组织的第一组可能的标记。在标识符数据的元素的每个输入实例之后，如上所述，根据服务器进行的定义或随机化，内轮盘上的标记被替换为已在下一组中随机化的标记。

作为输入标识符的元素的示例，参考图 20，假定用户的标识符是单词“B A N K”。为了根据本发明来输入此标识符，用户在内心选择以上称为内轮盘 202 的包括所有标记的界面的环形区域的 16 个划界部分或者扇区之一中的标记。如图 20 所示，使用了十六个这样的扇区；但是，具有更少或更多的扇区的界面是有用的，并且在本发明的范围内。对于使用六十个标识符元素的应用，最优选的标记数目是十六个，因此最优选的扇区数目是十六个。

如图 20 还示出的，♥ 或“心形”标记位于扇区 214 中。扇区 214 与外轮盘 204 相邻并在外轮盘 204 内部从 214A 到 214B 延伸。图 20 中的轮盘上的元素和扇区的类型、性质、形状、颜色、配置和数目是示例性的。可以使用几乎任何类型、性质和数目的元素，并且它们可具有几乎无限数目形状、大小和配置。如图 20 还示出的，内轮盘 202 的扇区 214 与一组随机选择的标识符元素“b a Z Y”相邻，该组随机选择的标识符元素被置于外轮盘 204 的一个扇区中，该扇区在扇区 204 的径向外部，并且一般具有相同的形状和大小。在此示例中，用户通过促动按钮 206 和/或 208 以便旋转它来导引内轮盘 202，从而使得用户选择的标记，即♥，与标识符的第一元素，即“B”，对齐。系统优选地被编程为使得标记可以位于标识符的元素的任一方向的几度之内而仍被认为是对齐的。换言之，用户选择的标记

和标识符的每个元素之间的某个距离范围可被定义为可接受的。一旦达到对齐，用户就停止旋转内轮盘 202 并点击“下一个”按钮。然后，系统提供第二组标记，即相同的标记，但是在顺序上已被服务器随机化。系统随后准备好供用户输入标识符的下一元素。接下来，用户再次促动用于旋转内轮盘的按钮 206 和/或 208 以将所选的标记与下一元素对齐，并且点击“下一个”按钮，如上所述。用户随后对标识符的每个剩余元素重复此过程，以便每个元素被以适当的顺序一个接一个地输入。

优选地，提供了“输入”按钮供用户促动，以指示标识符的最后一个元素已被输入。或者，在不需要时，例如在标识符具有预定的固定长度的情况下，可无需使用“输入”按钮。

根据替换实施例，内和外轮盘以及扇区可以是可互换的。换言之，标记可被置于外轮盘的扇区上并可从外轮盘的扇区来选择，并且内轮盘可包括标识符元素。在其他替换实施例中，可以使外轮盘可旋转。

图 21 图示出另一类型的动态图形认证界面，其中用户实时选择标记，并且包括标识符元素的区域相对于标记运动的相对运动是线性的而不是旋转性的。用作下方横条上的标记的颜色/图标/形状/形式或者指派给末端用户的 PIN 也可以基于可由末端用户或服务提供者预先确定的逻辑来生成。这也使得末端用户能够识别个性化滑块。

在图 21 的实施例中，界面被称为滑块显示 216。图 20 和图 21 中所示的字母数字和非字母数字符号是示例性的，即其他图形符号和图像也可用于实现本发明。另外，各种区域的形状、风格、配置、颜色、朝向可被改变，只要标识符元素和标记之间的相对运动被提供并可测量。滑块显示 216 可选地包括用户名输入字段 218、口令输入字段 220 和用于在促动时将显示的一个部分或区域移动到另一部分或区域的可选择的箭头按钮 222 和 224。在图 21 的实施例中，两个区域被放置成行，一个在另一个上面。每个区域或行被划分成单元，优选地具有相同的大小，以便能够将一个放置在另一个上面。从而，滑块显示 216 包括可移动的下方行 226 和固定的上方行 230，其中可移动的下方行 226 具有多个单元，每个单元具有标记，例如在此示例中位于字母“B”下面的黑桃或王牌符号 228，即

“♣”。在通过对“左箭头”按钮 222 和“右箭头”按钮 224 使用导航性鼠标点击来进行操作时，下方行 226 可滑动位移。以与上面所论述的图 20 所示的轮盘实施例的位移相同的方式来测量滑块显示 216 的可移动的下方行 226 相对于固定的上方行 230 的位移，只不过此类显示的位移是线性的而不是放射性的。一旦用户利用“下一个”按钮 232 来表明数据的输入，将标识符元素输入到系统中的每次实例的位移值就被发送到服务器并被服务器解码。从而，针对滑块显示 216 将位移信息发送到服务器的动作与图 20 的轮盘实施例的类似，只不过滑块显示 216 的位移是线性的，而轮盘显示 200 的位移是旋转性的。

优选地，提供“重置”按钮 234，以使得用户可以重新开始输入用户名、口令或其他标识符或代码。类似地重置能力也可结合其他实施例使用，例如结合图 20 中示出但未编号的使用。在用于显示标识符字段的元素的输入状态的图像中，可选地并且优选地提供了图标 236，以指示用户名或口令的多少个元素已被输入。优选地，所输入的元素不被显示在用户名字段 218 或口令字段 220 中，以帮助防止“越肩”密探看到字段信息。星号可被显示在输入部分中，以表明每个元素的输入。

根据替换实施例中，图 21 所示的行可以互换，即可以使得能够从上方行选择标记，可以使该上方行能被用户滑动以便导引标记，并且可以使下方行包括标识符的可能元素。

或者，图 20 的轮盘上的元素和/或标记，以及图 21 的行中的元素和/或标记可针对用户定制，例如动物、人、场景的图片或任何其他图像。或者，标记可使用为用户所知晓并由服务器预先定义的逻辑。

图 20 和 21 中的用户界面被示为具有两个标识符，例如用户名和口令。但是，本发明并不限于两个因素；在本发明的范围内可包括额外的因素。例如，可添加 PIN 代码，以使得加密/解密系统成为三标识符系统。

以上图中示出的每个图形界面优选地是利用本领域普通技术人员已知的传统软件来生成并发送到用户设备的，所述传统软件例如是 MACROMEDIA FLASH 品牌软件或 JAVA 品牌软件或使用 SVG 标准。在优选实施例中，FLASH 软件被用于生成图形界面。

虽然已经描述了本发明的具体实施例，但是各种修改、更改、替换构造和等同也被涵盖在本发明的范围之内。

因此，说明书和附图被认为是示例性的而不是限制性的。但是，很明显，可对其进行修改、减去、删除和其他修改和改变，而不脱离权利要求所述的本发明的更宽广精神和范围。

参考图 22-28，将描述本发明的各种过程特征的框图。在图 22 中，服务器获得用于图 20 或 21 的实施例中并在其中示出的非用户个性化标记，并从可用标记列表中随机选择某一数目“n”的标记，然后将这些标记返回到客户端计算机的显示屏。如图 23 所示，使用实质上相同的过程，只不过在获得标记时，包含用户个性化标记的数据库被访问，并且在将多组标记返回到客户端计算机之前，这多组标记被选择并随机化。客户端计算机处的优选过程流程在图 24 的框图中图示出，其中过程开始于用户进入登录页面。然后，客户端从服务器或从本地机加载应用，并且取得随机的标记列表之一。用户随后输入标识符或其他代码的元素，客户端计算机随后经由网络将数据发送到服务器。服务器随后处理数据，并且如果对于该标识符的数据输入成功，则去往下一页。如果未成功，服务器将控制返回到客户端计算机，以获得另一个随机化标记列表，并像之前那样继续该过程。

图 25 是图示出在无加密情况下发生的客户端/服务器交互的框图，图 26 图示出有加密的交互。在图 25 中，客户端计算机作出请求，并且服务器获得标记列表并将列表返回到客户端。在图 26 中，发生类似地交互，但是除此之外，服务器在获得标记列表并将标记列表返回到客户端之前对从客户端接收到的数据进行解密。图 27 图示出了认证交互，其中一旦已可选地对从客户端接收到的数据进行了解密，就执行对该数据的认证，然后服务器向客户端返回成功或失败指令。在图 28 中，详述了认证过程，从而在开始过程后，利用为该会话选择的标记列表，发生将位移映射到标记下的字符的操作。然后，利用存储在数据库中的口令作为尝试性判定的基础，判定用户为标识符（例如口令）的第一元素选择了哪个标记。接下来，系统通过顺次为每个元素比较标记所作出的位移以及标识符的正确元

素是否已被匹配，来核实其余的标识符元素。如果作出了所有正确匹配，则服务器发送“成功”指令，以指示认证成功。如果没有作出所有正确匹配，则服务器发送“失败”指令并且该过程重新开始。

作为可用在图 16-19 的实施例中所使用的那类映射的代码的示例，图 29-30 提供了一些可以使用的典型优选伪代码。但是，对于实现这里参考附图描述的本发明的特征所需的代码来说，相信所有这种代码都在本领域的普通技术的范围内，并且相信可以很容易提供具体应用。

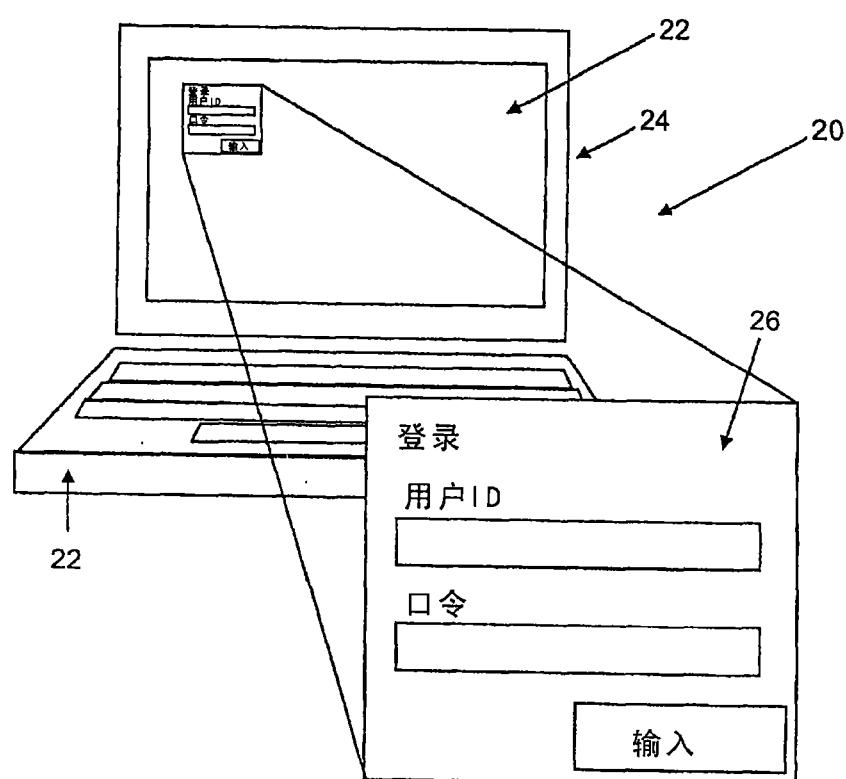


图1
(现有技术)

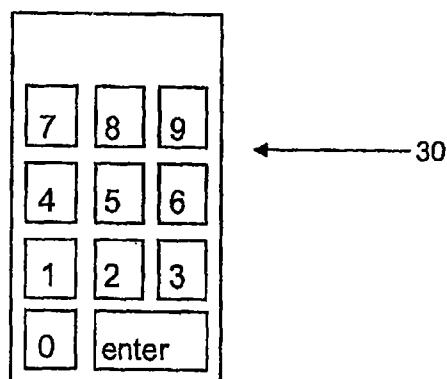


图2
(现有技术)

32

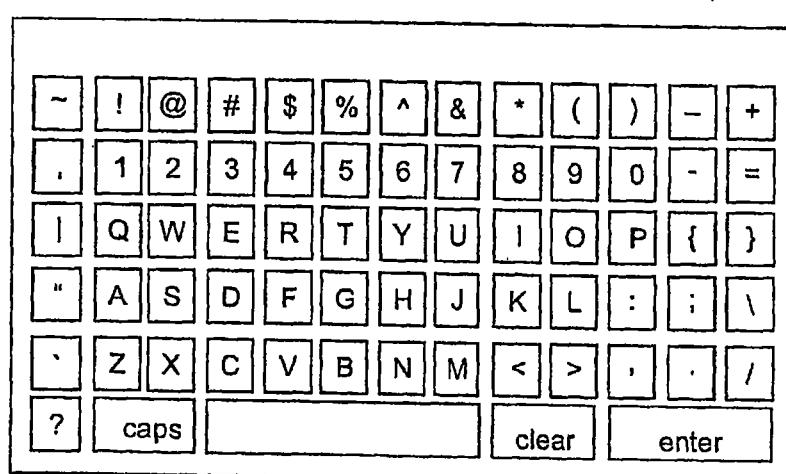


图3
(现有技术)

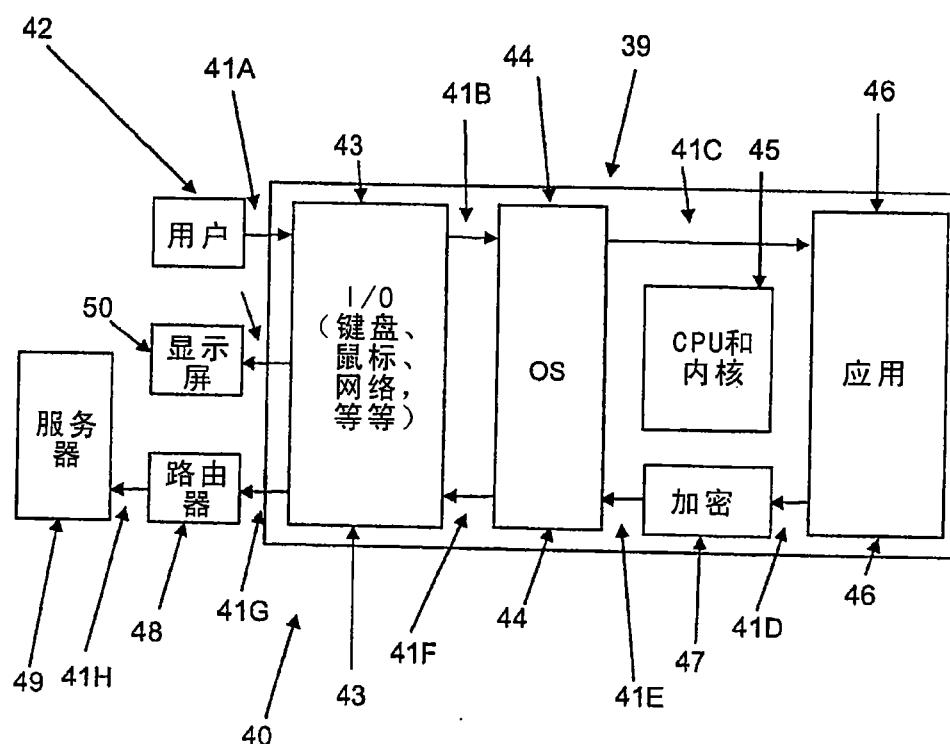


图4
(现有技术)

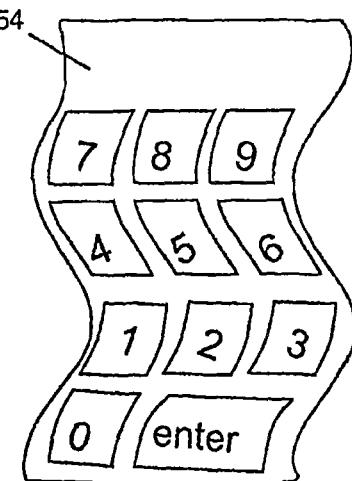
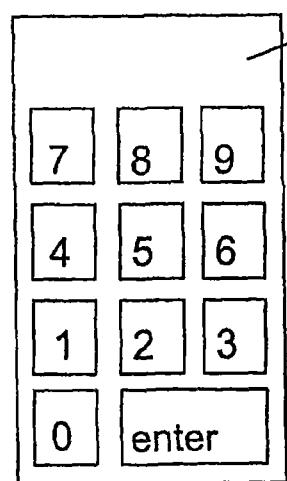


图5
(现有技术)

图6

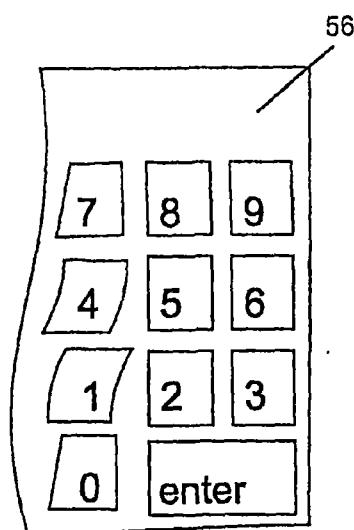


图7

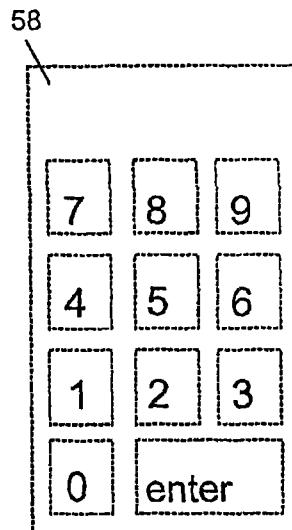


图8

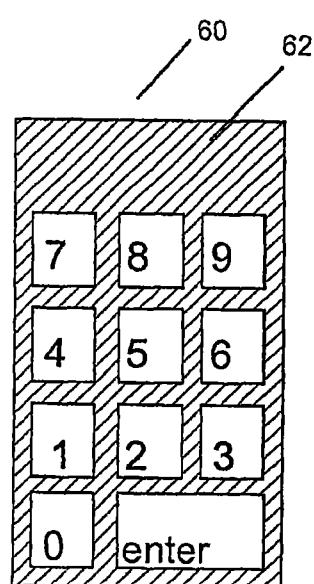


图9

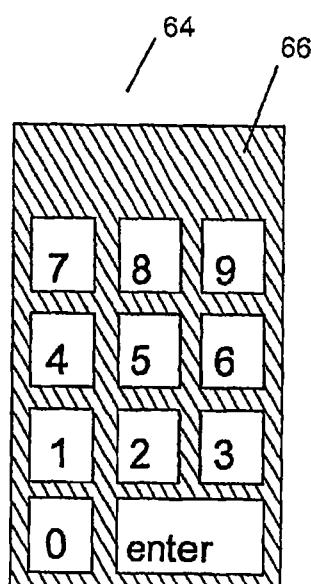


图10

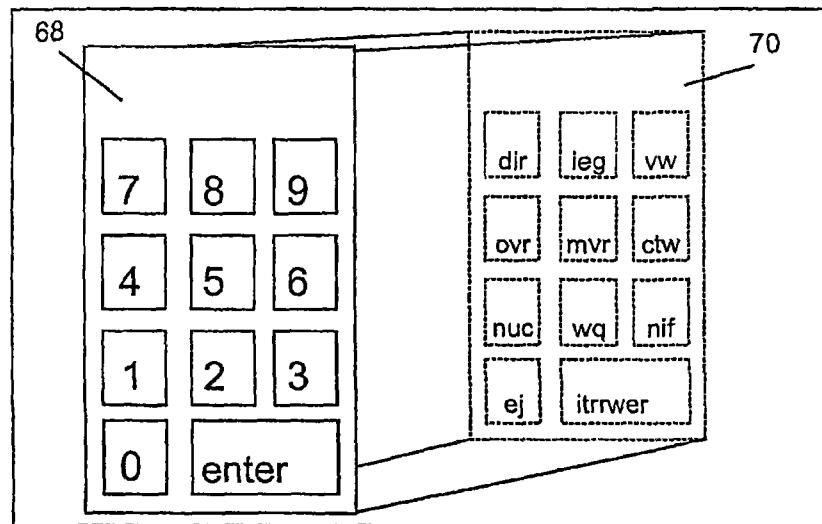


图 11

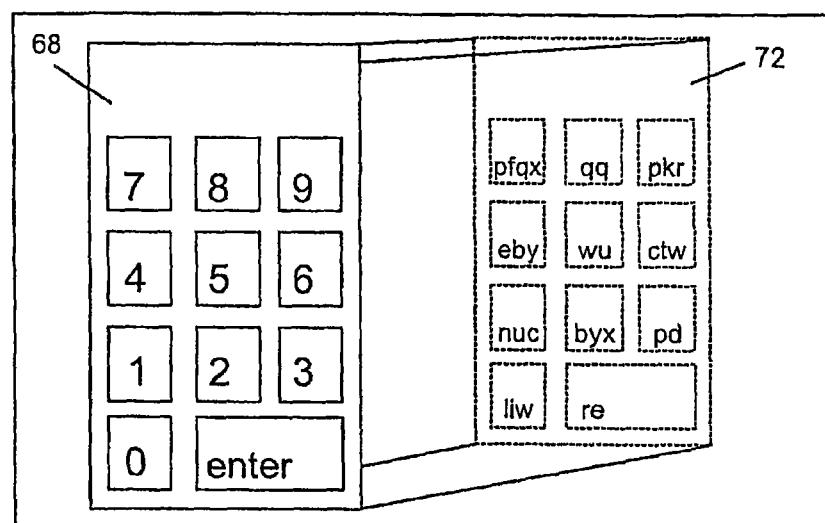


图 12

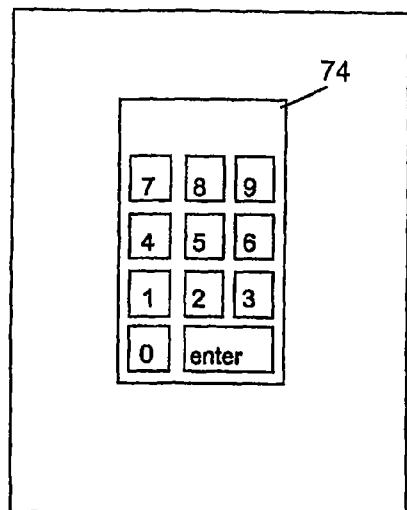


图13
(现有技术)

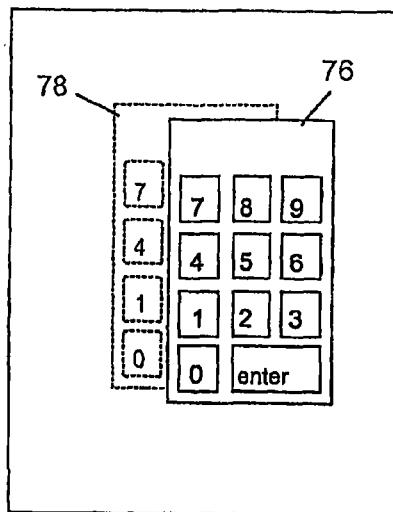


图14

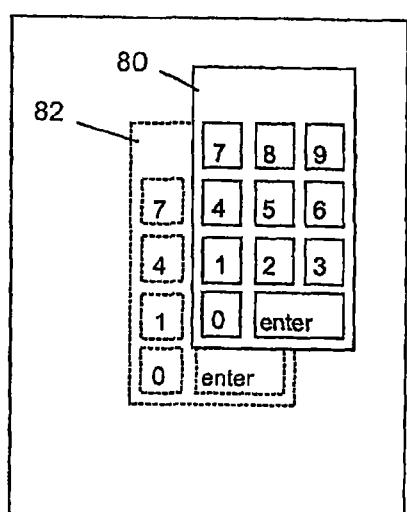


图15

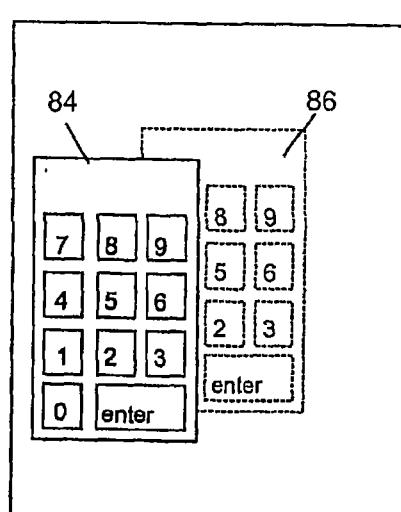


图16

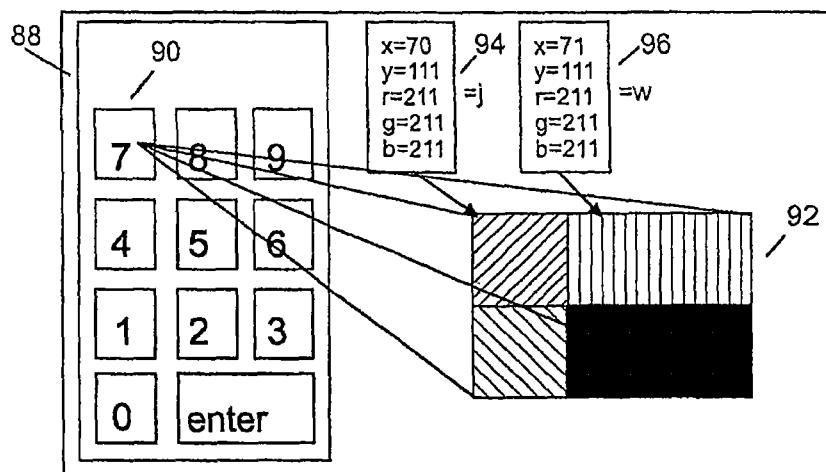


图 18

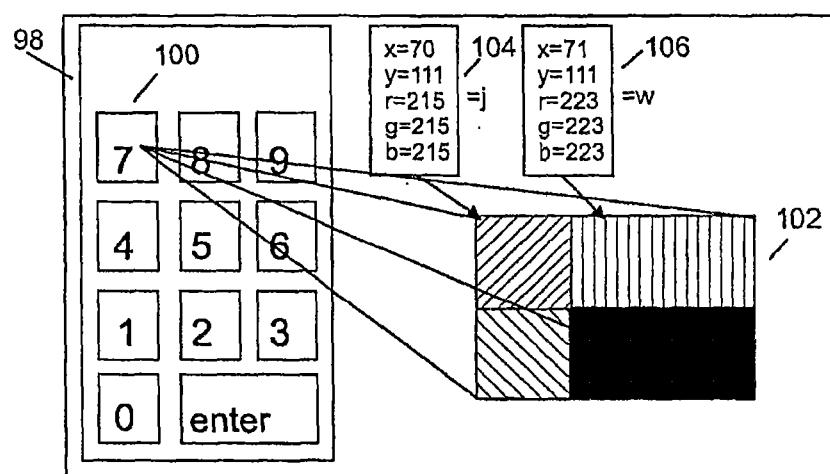


图 17

数字/字母数字/滑块/轮盘的典型实现过程

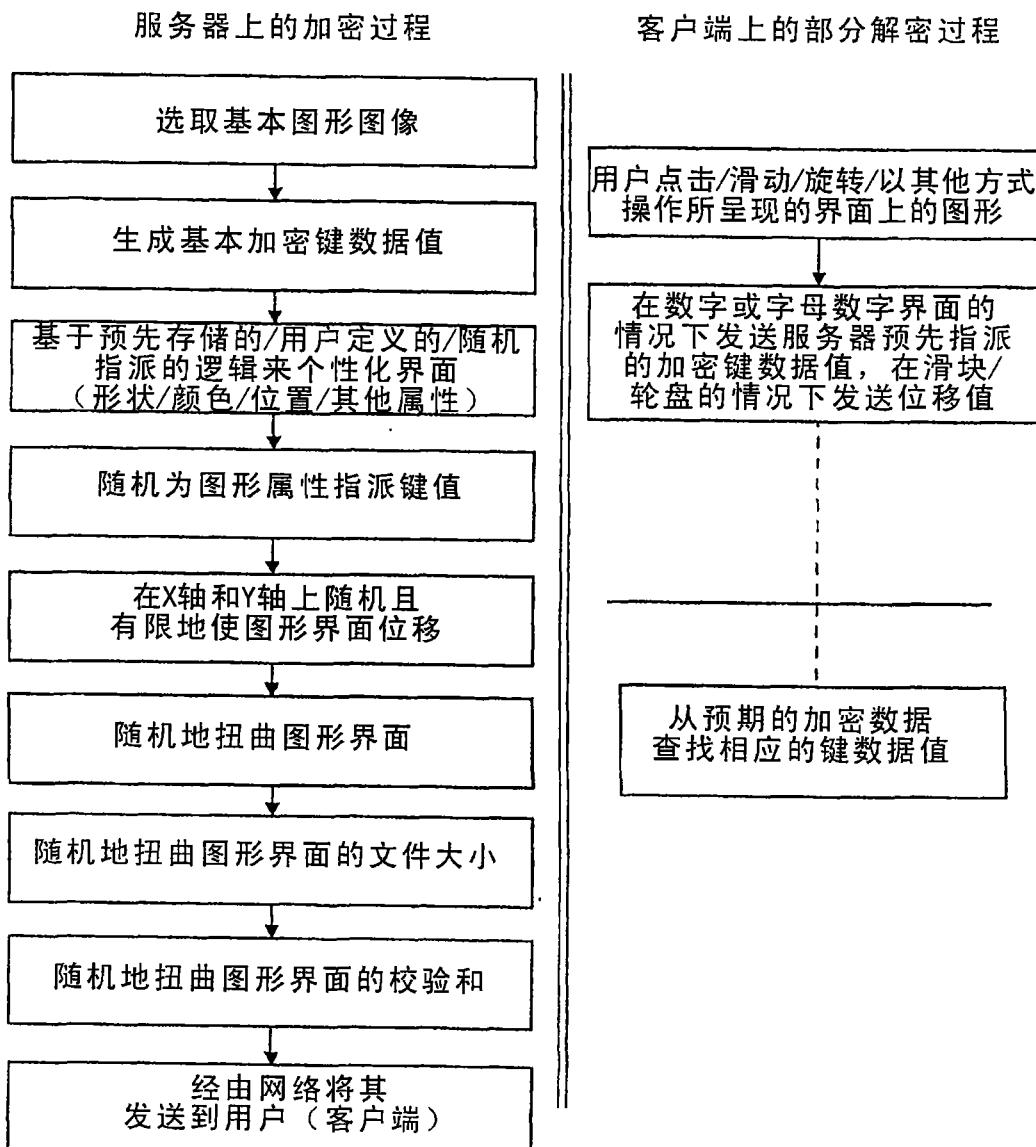


图19

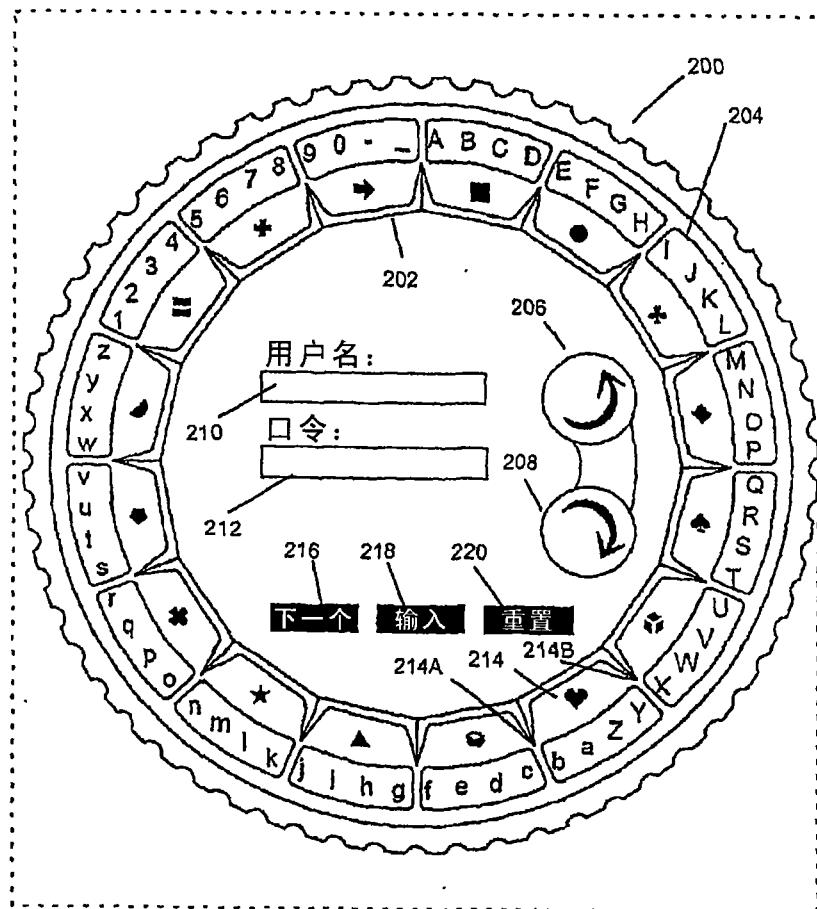


图20

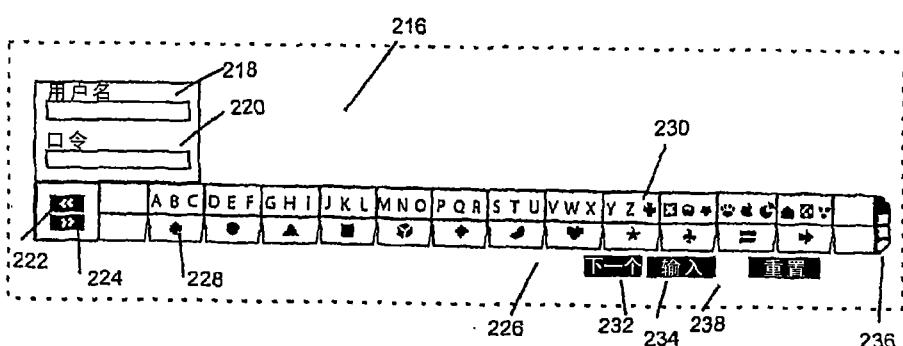


图21

非用户个性化标记



从列表中随机地选择n个标记
(其中n是屏幕上图标/
标记的数目)

随机化
所选标记

返回标记

用户个性化标记
(用户始终获得
相同的标记列表)



从用于用户的
数据库获得
标记列表



随机化
所选标记

返回标记

图22

图23

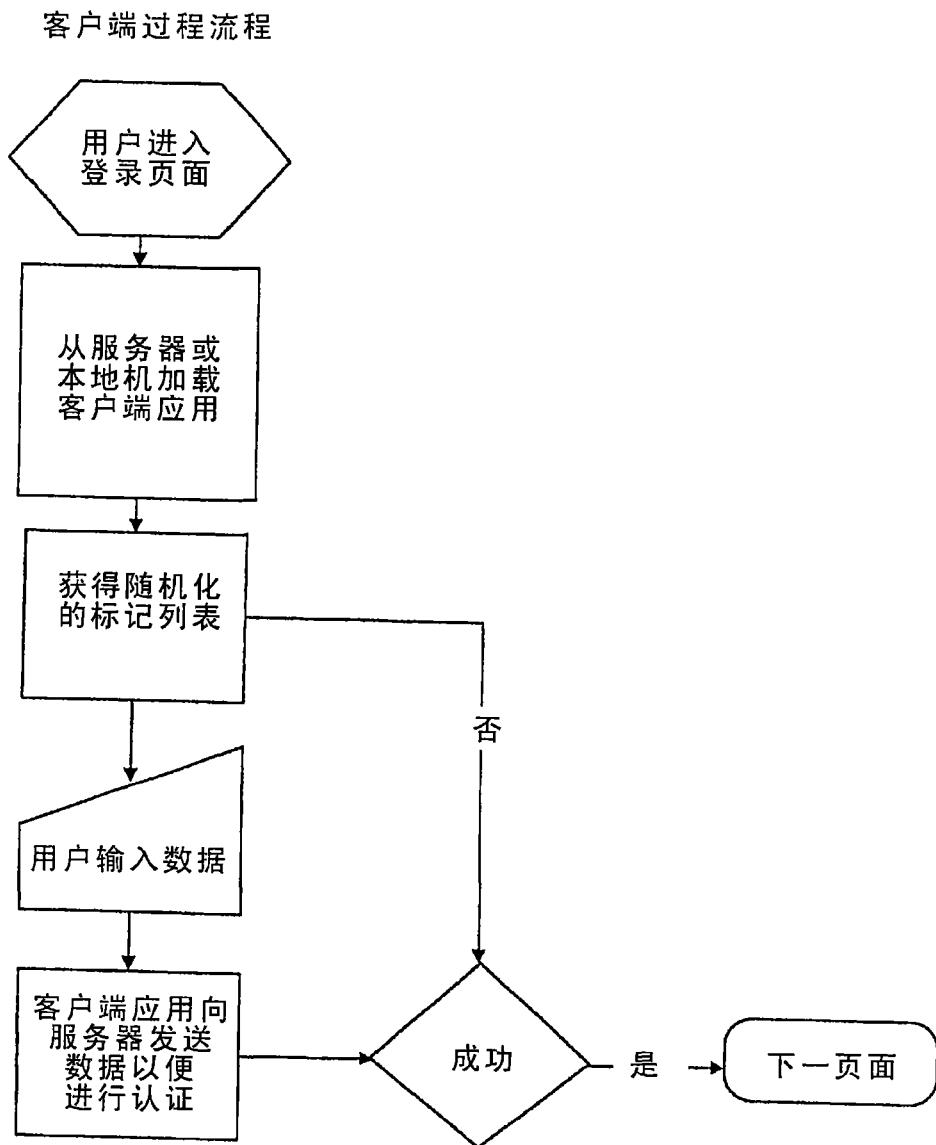


图24

客户端/服务器
交互（没有加密）

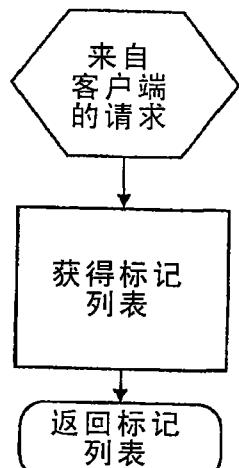


图25

客户端/服务器
交互（有加密）

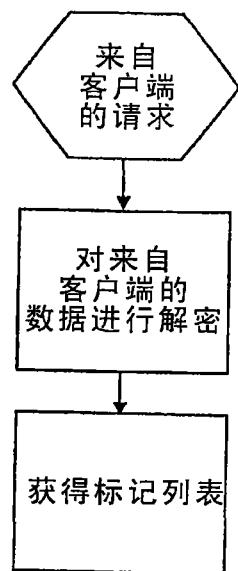
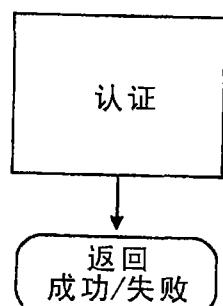


图26

认证
来自客户端的用户输入



返回标记列表

图26

对来自客户端的数据进行解密（可选）

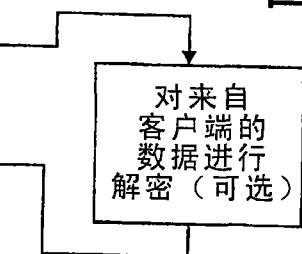


图27

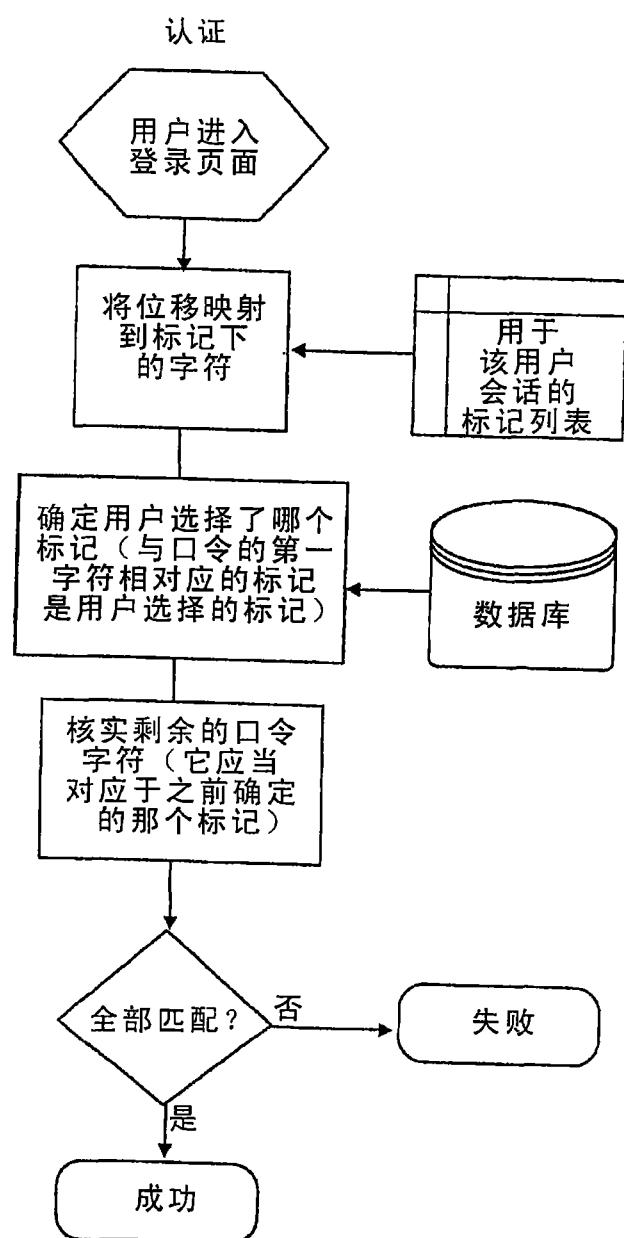


图28

//流程序列：

1. 调用登录页面（参考：function printLoginPage()）
2. 从浏览器请求Flash客户端。参考反编译Flash代码是此文档的一部分。
3. Flash代码对服务器发出呼叫以获得要显示的轮盘。
参考函数genWheel生成被格式化并提供给flash代码的轮盘。
4. 用户利用滑块输入PIN
5. Flash向服务器发送位移
6. 服务器代码解析http请求并调用 authenticatePIN函数
7. 基于authenticatePIN响应，用户被给予访问许可或不被给予访问许可

//HTTP代码片断

```
function printLoginPage () {
    //将以下代码嵌入到html页面中
    print "<html></body>
<table width="100%" border="0" cellpadding="24" cellspacing="0">
<tr>
    <td align="center">
        <form method="post" action="authSimple.jsp">
            <table>
                <tr>
                    <td>Login ID:</td>
                    <td><input type="text" name="usr" size=10></td>
                </tr>
                <tr>
                    <td>Password:</td>
                    <td><input type="password" name="pwd" size=10></td>
                </tr>
                <tr>
                    <td><input type="submit" value="Submit"></td>
                </tr>
            </table>
        </form>
    </td>
<td><object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444535400001"
codebase="http://fpdownload.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,0,0" width="1" height="1" align="middle">
</object>
</td>
</tr>
</table>"
```

图29

```

<param
name="allowScriptAccess" value="sameDomain" /> <param name="movie" value=""
SliderBarAuth.swf

        <param name="quality" value="high" />
        <param name="becolor" value="#ffffff" />
        <embed src="SliderBarAuth.swf" quality="high"
becolor="#ffffff" width="1" height="1" align="middle"
allowScriptAccess="sameDomain" type="application/x-shockwave-flash"
pluginspage="http://www.macromedia.com/go/getflashplayer" />
    -/object>
    </td>
</tr>

}

</table>
</body></html>

```

```

//用于生成轮盘标记的代码
function String genWheel( int length, int sets, long seed ){
    Random rand = seed != 0 ? new Random( seed ): new Random();
    String wheelstr = "".*;

    int[] outerWheel = new int[length];
    for (int i=0; i<length; i++) { outerWheel[i] = i; }

    wheelList = new Vector();
    for( int wheelCount = 0; wheelCount < sets; wheelCount++ ) {
        wheelstr += "&wheel"+wheelCount+"=";
        int arr[] = new int[length ];

        Integer newWheel[] = new Integer[ length ];
        for( int i = 0; i < length; i++ ) {
            arr[i] = 0;
        }
        for( int i = 0 ; i < length; i++ ) {
            int value = rand.nextInt( length );
            if ( arr[value] != 0 ) {
                --i;
                continue;
            }
            arr[value] = 1;
        }
    }
}

```

图29

```

        newWheel[i] = new Integer(outerWheel[value]);}
        wheelstr += formatIntegerArray(newWheel);
        wheelList.add(newWheel);
    }
    return wheelstr;
}

//认证过程
function int authenticatePin( Vector displacements, String PIN, Vector wheelList) {
    rcode = SUCCESS;

    Integer marker = null;
    for (int x=0,i=0; rcode = 0 && i<PIN.length(); i++,x++) {
        Integer[] of = (Integer[])wheelList.elementAt(x);int
        offset =
        ((Integer)displacements.elementAt(x)).intValue;
        String currchar = ""+PIN.charAt(i);
        int index = outer.indexOf(currchar);
        int bignum = 100000 *outer.length();
        int m;

        if (offset<0){
            m = (offset*-1)%outer.length;
            m = (bignum+index-offset)%outer.length;
        }else if (offset>0) {
            m = offset%outer.length();
            m = (bignum+index-offset)%outer.length;
        }else{ m=index; }

        if (marker==null){
            marker = of[m/3];
        }
        else{

if (a!= b){           int a = of[m/3].intValue ;
                    int b = marker.intValue;
                    rcode = WRONG_PIN;
                    break;
                }
}
}

```

图29 (续)

```
        }  
    }  
  
    if (rcode = SUCCESS){  
        //成功  
    } else {  
        //失败  
    }  
  
    return rcode;  
}  
}
```

图29（续）

```
movie 'SliderBarAuth.swf {
H flash 6, total frames: 2, frame rate: 12 fps, 672x56 px, compressed

frame 1 {
stopo; }

frame 1 {
    un - if;
    pw =
    trace('sss');
}

button 7 {

on (release, keyPress '<Enter>') {
    var serverUrl ='authDemo.jsp';
    var errorUrl ='errordemo.jsp';
    var resultauth_ly = new LoadVarso;
    resultauth_lv.onLoad = function (success) {
        if (success) {
            trace('datareceive');
            var v1 = resultauth_lv.redirect.charAt(0);
            trace(v1.length);if
                (V1 = 0) {
                    trace('success');
                    gotoAndStop(2);}
            if (resultauth_lv.redirect = 1) {
                getURL(resultauth_lv.url1, '');
            }
        } else {
            getURL(errorUrl,)}
    }

    var sendauth_ly = new LoadVarso;
    sendauth_lv.un = _root.un;
    sendauth_lv.pw = _root.pw;
    sendauth_lv.sendAndLoad(serverUrl, resultauth_lv,'POST');}
}
```

图30

```
movieClip 9 { }

instance HandleKeys of movieClip 9 {

    onClipEvent (keyDown) {
        if (joot.firstkey = 0) {
            _root.firstkey = 1;
            _root.timer[0] = getTimer();
        }
        if (Key.isDown(37) && _root.onceonly = 0) {
            _root.onceonly = 1;
            _root.startmoving(0);
        }
        if (Key.isDown(39) && _root.onceonly = 0) {
            _root.onceonly = 1;
            _root.startmoving(1
        }}}

        onClipEvent (keyUp) { _root.onceonly = 0;
    keycode = Key.getCode();
    if (keycode = 37) {
        _root.stopmoving(0);
    } else {
        if (keycode = 39) {
            _root.stopmoving(1);
        }
    }
}

frame 2 {
    stop();
    _root.un.tabindex = 1;
    _root.pw.tabindex = 2;
    _root.enterbtn.tabindex =
    3;root. resetbtn.tabindex =
    4;}

frame 2 {
    function fadeout0 {
        var v I =this;
        if (inout = 0) {
            v1.-alpha -= faderatio;
```

图30（续）

```

updateAfterEvent();
if
(A .alpha < 0) {
  updatecharso ;
  inout = 1;
}
} else {
  v1._alpha += faderatio;
  updateAfterEvent();
  if(v1._alpha > 100) {
    inout = 0;
    delete _root.fl.onEnterFrame;
  }
}
}

function updatecharso {
var v2 = root;
var v1 = 0;
while (v1 < symbolcount) {
  eval('char' + v1) = chars[randomarray[vcount][v1]];
  ch[v1].setRGB(col[v2randomarray[vcount] [v1]]);
  rch[v1].setRGB(col [v2.randomarray[vcount] [v1]]);
  lch[v1].setRGB(col[v2.randomarray[vcount] [v1]]);
  ++v1;
}
}

function outputo {
var v1 = _root;
outx = fl_x;
outx2 = outx - centered;
outx3 = outx2 / SS;
if (outx3 >= numberofchars) {
  v1.angle = outx3 - numberofchars; }
else {
  v1.angle = -(numberofchars - outx3); }
if (v1.angle == 3611 v1.angle = -36) { v1
.angle = 0;
}
}

function movementO {
sliderx = root.slide.slider_x;
newx = sliderx * ratio;
usedx = newx + centered;
}

```

图30（续）

```
_root.fl._x = usedx;
updateAfterEvent();
}

function postsliderQ {
    thex = fl._x;
    Q = thex % SS;
    halfSS=SS/2;  if
    (Q = 0) {
        outputQ;
        delete this.onEnterFrame;
        delete root.fl.onEnterFrame;
    } else {
        if (Q > haOS) { right: = fl._x;
            rightx - Q; right: += SS;
            fl.onEnterFrame = stopright;
        } else {
            leftx = fl._x;
            leftx -= Q;
            -root.fl.x = leftx;

            fl.onEnterFrame = stopleft;
        }
    } function
    stamoving(rightleft) { speed
    = startspeed;
    if (rightleft = 1) {
        fl.onEnterFrame = startright; }
    else {
        fl.onEnterFrame = startleft;
    }}

function stopmoving(rightleft) {
    if (rightleft = 1) {
        right: = fl._x;
        Q = rightx % SS;
        rightx -= Q;
        rightx += SS;
        fl.onEnterFrame = stopright;
    } else {
        leftx = fl._x;
        Q = leftx % SS;
```

图30（续）

```
leftx -= Q; fl.onEnterFrame =
stopleft;
}

function rightQ {
    currentx = fl._x;
    newx = currentx - centered;
    usedx = newx / ratio;
    if (usedx > slidermax) {
        usedx = slidermax;}
    _root.slide.slider._x = usedx;
    updateAfterEventQ;
    fl._x += speed;
    if (currentx >= maxright) {
        speed =
minSpeed;fl._x =
maxright;
        delete this.onEnterFrame;
    }
}

function stoprightQ {
    var v1 = -root;
    currentx = fl._x;
    newx = rightx - centered;
    usedx = newx / ratio;
    if (usedx > slidermax) {
        usedx = slidermax;}
    v1.slide.slider._x = usedx;
    updateAfterEventQ;
    if (currentx >= maxright) {
        speed =
minSpeed;fl._x =
maxright; outputQ;
        delete this.onEnterFrame;
        delete v1.fl.onEnterFrame;
    } else {
        if (currentx >= rightx - SS) {
            speed =
minSpeed;fl._x =
rightx;
            outputQ;
            delete this.onEnterFrame;
            delete v1.fl.onEnterFrame;
        }
    }
}
```

图30（续）

```
function startrighto {
    currentx = fl._x;
    newx = currentx - centered;
    usedx = newx / ratio;if (usedx
> slidennax) {
        usedx = slidermax;}
    _root.slide.slider._x = usedx;
    updateAfterEvento; stopping
    = false;
    speed *= startFactor;
    fl._x += speed;
    if (currentx >= maxright) {
        speed = minSpeed;fl._x =
        maxright;delete
        this.onEnterFrame;
    } else {
        if (speed > maxSpeed) {
            speed = maxSpeed;
            fl.onEnterFrame = right;
        }
    }
}

function lefto {
    currentx = fl._x;
    newx = currentx - centered;
    usedx = newx / ratio;
    root.slide.slider._x = usedx;
    if (usedx < slidermin) {
        usedx = kdermin;}
    updateAfterEvento;
    fl._x -= speed;
    if (currentx <= maxleft) {
        speed = minSpeed;fl._x
        = maxleft;
        delete this.onEnterFrame;
    }
}

function stoplefto {
    var v 1 = -root;
```

图30（续）

```
currentx = fl._x;
newx = leftx - centered;
usedx = newx / ratio;if
(usedx < slidermin) {
    usedx = slidermin;}
v1.slide.slider._x = usedx;
updateAfterEvent();
currentx = fl._x;
stopping = true;
speed *= stopFactor;
fl._x -= speed;
if (currentx <= maxleft) {
    speed = minSpeed;fl._x =
maxleft;
outputo;
delete this.onEnterFrame;
delete v1.fl.onEnterFrame;
} else {
if (currentx >= leftx) {
    speed = minSpeed;
    fl._x = leftx;
    outputo;
    delete this.onEnterFrame;
    delete v1.fl.onEnterFrame;
}
}
}

function startlefto { currentx = fl._x;
newx = currentx - centered;
usedx = newx / ratio;
if (usedx < slidermin) {
    usedx = slidermin;}
root.slide.slider._x = usedx;
updateAfterEvent();
stopping = false;speed
*= startFactor; fl._x -
= speed;
if (currentx <= maxleft) {
    speed =
minSpeed;fl._x =
maxleft;
    delete this.onEnterFrame;
} else {
```

图30（续）

```
        if (speed > maxSpeed) {
            speed = maxSpeed;
            fl.onEnterFrame = left;
        }
    }

}

frame 2 {
    fadeoutx = 100;
    fadeinx = 0;
    faderatio = 50;
    inout = 0;
    dcUrl ='dc';
    firstkey = 0;
    sliderpress = 0;
    rightpress = 0;
    leftpress = 0;
    timecount = 0;
    timer = new Array();
    _root.unbox.tabIndex = 1;
    _root.pwbox.tabIndex = 2;
    _root.enterbtn.tabindex = 3;
    _root.resetbtn.tabIndex = 4;
    onceonly = 0;
    maxright = 1530;
    maxleft = 450;
    centered = 450;
    ratio = 4;
    slidercenter = 135;
    slidermax = 270;
    slidermin = 0;
    numberofchars = 36;
    angle = 0;
    symbolcount = 12;
    stopFactor = 0.9;
    startFactor = 1.3;
    minSpeed = 0.1;
    maxSpeed = 15;
    startspeed = 4;
    starting = false;
    stopping = false;
    speed = minSpeed;
    SS = 15;
    vcount = 0;
```

图30（续）

```

vcrypt1 = "
vcrypt2 = ",
vcrypt3 = '
vcrypt4 = "
var local_so = sharedobject.getLocal('shdcUrl');
var send_var;
if (local_so.datashdcUrl != null) {
    send_var = local_so.data.shdcUrl;
} else {
    send var =
}
var result_ly = new LoadVarso;
result_lv.onLoad = function (success) {
    if (success) {
        local_so.data.shdcUrl = retval;
    } else { }

var send_ly = new LoadVarso;
send Iv.client ='v&;
sendlv.v = send_var;
send_lv.sendAndLoad(dcUrl, result_iv, 'POST'); cot
= new Arrayo;
col[0] ='0x207EAE;
col[1] ='0xDE791A;
col[2] ='0xB 11616%
col[3] ='0038C29%
col[4] ='0x394068;
col[5] ='0x207EAE';
col[6] ='0xDE791A;
col[7] ='0xB 11616`;
col[8] ='0038C29%
col[9] ='0x39406B;
col[10] ='0x207EAE';
col[11] ='0xDE791A;
chars = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L'];
chm0 = new Color('fl.mi.chars.lrg_char0');chml =new
Color('fl.ml.chars.lrg_char1');
chm2 = new Color('fl.mi.chars.lrg_char2');
chm3 = new Color('fl.ml.chars.lrg_char3');
chm4 = new Color('fl.ml.chars.lrg_char4');
chm5 = new Color('fl.ml.chars.lrg_char5');
chm6 = new Color('fl.mi.chars.lrg_char6');
chm7 = new C olor('fl. mi. chars. lrg_char7');
chm8 = new Color('fl.mi.chars. lrg_char8');
chm9 = new Color('fl.mi.chars. lrg_char9');

```

图30（续）

```

chm 10 = new Color('fl.ml.chars.lrg_charl O');
chml 1=new Color('fl.ml.chars.lrg_charl 1');
ch = [chm0, chml, chm2, chm3, chm4, chm5, chm6, chm7, chm8, chm9, chm10,
chm 11];
rchm0 = new Color('fl.mlr.chars.lrg_charO');
rchml =new Color('fl.mlr.chars. lrg_char1');
rchm2 = new Color('fl.mlr.chars. lrg_char2');
rchm3 = new Color('fl.mlr.chars.lrg_char3');
rchm4 = new Color('fl.mlr.chars.lrg_char4');
rchm5 = new Color('fl.mlr.chars.lrg_char5');
rchm6 = new Color('fl.mlr.chars. lrg_char6');
rchm7 = new Color('fl.mlr.chars.lrg_char7');
rchm8 = new Color('fl.mlr.chars.lrg_char8');
rchm9 = new Color('fl.mlr.chars.lrg_char9');
rchm10 = new Color('fl.mlr.chars.lrg_charlO');
rchml 1=new Color('fl.mlr.chars.lrg_charl 1');
rch = [rchm0, rchml, rchm2, rchm3, rchm4, rchm5, rchm6, rchm7, rchm8, rchm9,
rchm 10, rchm 11 ];
lchm0 = new Color('fl.mll.chars.lrg_charO');
lchml = new Color('fl.mll.chars.lrg charl');
lchm2 = new Color('fl.mll.chars.lrg_char2');
lchm3 = new Color('fl.mll.chars.lrg_char3');
lchm4 = new Color('fl.mll.chars. lrg_char4');
lchm5 = new Color('fl.mll.chars. lrg_char5');
lchm6 = new Color('fl.mll.chars.lrg_char6');
lchm7 = new Color('fl.mll.chars.lrg_char7');
lchm8 = new Color('fl.mll.chars.lrg_char8');
lchm9 = new Color('fl.mll.chars.lrg_char9');
lchm10 = new Color('fl.mll.chars.lrg_charlO');
lchml 1=new Color('fl.mll.chars.lrg_charl 1');
lch = [lchm0, lchml, lchm2, lchm3, lchm4, lchm5, lchm6, lchm7, lchm8, lchm9,
lchm 10, lchm 11 ];
trace('sss' + resultauth_lv.redirect); wheel0
= resultauth_lv.wheel0.split('X'); wheel1 =
resultauth_lv.wheel 1.split('X'); wheel2 =
resultauth_lv.wheel2.split('X'); wheel3 =
resultauth_lv.wheel3.split('X'); wheel4 =
resultauth_lv.wheel4.split('X'); wheel5 =
resultauth_lv.wheel5.split('X'); wheel6 =
resultauth_lv.wheel6.split('X'); wheel7 =
resultauth_lv.wheel 7.split('X'); wheel8 =
resultauth_lv.wheel 8.split('X'); wheel9 =
resultauth_lv.wheel9.split('X'); wheel 10 =
resultauth_lv.wheel10.split('X'); wheel 1 =
resultauth_lv.wheel11.split('X'); wheel 12 =
resultauth_lv.wheel12.split('X');

```

图30 (续)

```
wheel 13 = resultauth_lv.wheel 13.split('X');
wheel 14 = resultauth_lv.wheel14.split('X');
wheel 15 = resultauth_lv.wheel15.split('X');
wheel16 = resultauth_lv.wheel 16.split('X');
wheel 17 = resultauth_lv.wheel17.split('X');
wheel 18 = resultauth_lv.wheel 18.split('X');
wheel19 = resultauth_lv.wheel 19.split('X');
_root.passedkey = resultauth_lv.key;
randomarray = [wheel0, wheel 1, wheel2, wheel3, wheel4, wheel5, wheel6, wheel7,
wheel8, wheel9, wheel 10, wheel 11, wheel 12, wheel 13, wheel 14, wheel 15, wheel 16,
wheel 17, wheel 18, wheel 19];
var i = 0;
while (i < symbolcount) {
    eval('char' + i) = chars [randomarray[vcount][i]];
    ch[i].setRGB(colLroot.randomarray[vcount] [i]);
    rch[i].setRGB(colLroot.randomarray[vcount] [i]);
    lch[i].setRGB(colLroot.randomarray[vcount] [i]);
}
}

movieClip 16 {
    frame 1 {
        stop();
    }
    frame 2 {
        stop();
    }
}

movieClip 21 {}

movieClip 23 {}

movieClip 25 {}

movieClip 27 {}

movieClip 29 { }
```

图30 (续)

```
movieClip 31 { }

movieClip 33 { }

movieClip 35 { }

movieClip 37 { }

movieClip 39 { }

movieClip 41 { }

movieClip 43 { }

movieClip 44 { }

movieClip 45 { }

movieClip 46 {
}

movieClip 51 {

frame 1 {
stop(); }

frame 2 {
stop(); }

frame 3 {
stop(); }

frame 4 {
```

图30（续）

```
        stopo;  
    }  
  
    frame 5 {  
        stopo;  
    }  
  
button 91 {  
  
    on (press) {  
        ++rightpress;  
        if (_root.firstkey = 0) {  
            _root.firstkey = 1;  
            root.timer[0] = getTimer();  
            startmoving(1);  
        }  
        ,  
    on (release) {  
        stopmoving(1);  
    }  
  
    on (dragOut) {  
        stopmoving(1);  
    }  
  
button 94 {  
  
    on (press) {  
        ++leftpress;  
        if (_root.firstkey = 0)  
            {_rootfirstkey = 1;  
            root.timer[0] = getTimer();  
            startmoving(0);}  
    on (release) {  
        stopmoving(0);  
    }  
  
    on (dragOut) {  
        stopmoving(0);  
    }  
}
```

图30 (续)

```

button 95 {

on (press, keyPress '<Enter>') {
    if (_root.firstkey = 0) {
        -root.firstkey = 1;
        timer[0] = getTimero;
        ++timecount;
        timer[1] = timer[0];
    } else {
        ++timecount;
        timer[timecount] = getTimero ;
    }
    ++vcount;
    _root.gauge. gotoAndStop(vcount + 1);
    eval('_root.angle' + vcount) = _root.angle;
    -root.fl._x = 990;
    _root.slide. slider._x = 135;
    _root.outputo;
    if (vcount = 4) {
        timeout = new Arrayo ;
        timeout[0] = 0;
        var i = 1;
        while (i < timer.length) {
            timeout[i] = timer[i] - timer[i - 1];
            ++i;
        }
        var ToSend = new LoadVarso;
        ToSend.angle1 = _root.angle1;

ToSend.angle2 = _root.angle2;
ToSend.angle3 = _root.angle3;
ToSend.angle4 = _root.angle4;
ToSend.timer1 = timeout[1];
ToSend.timer2 = timeout[2];
ToSend.timer3 = timeout[3];
ToSend.timer4 = timeout[4];
ToSend.rightpressed = _root.rightpress;
ToSend.leftpressed = _root.leftpress;
ToSend.sliderpressed = _root.sliderpress;
ToSend.pw = _root.pw;
ToSend.un = _root.un;
ToSend.key = _root.passedkey;
ToSend.reset = 0;
ToSend.send(_root.serverUrl, '_self); }

else {
}
}

```

图30（续）

```
        root.fl.onEnterFrame = _root.fadeout;
    }
}
}

button 96 {

on (press) {
    var ToSend = new LoadVarso;
    ToSend.angle1 = _root.angle1;
    ToSend.angle2 = _root.angle2;
    ToSend.angle3 = _root.angle3;
    ToSend.angle3 = _root.angle4;
    ToSend.timer1 = timeout[1];
    ToSend.timer2 = timeout[2];
    ToSend.timer3 = timeout[3];
    ToSend.timer4 = timeout[4];
    ToSend.rightpressed = _root.rightpress;
    ToSend.leftpressed = _root.leftpress;
    ToSend.sliderpressed = root.sliderpress;
    ToSend.pw = _root.pw;
    ToSend.un = root.un;
    ToSend.key = _root.passedkey;
    ToSend.reset = 1;
    ToSend.send(_root.serverUrl, self);
    loadMovieNum(_level0._url, 0);}
}

movieClip 101 { }
```

图30（续）

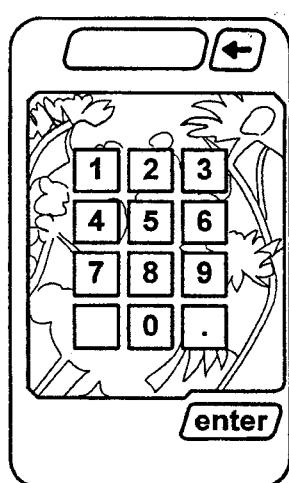


图31A

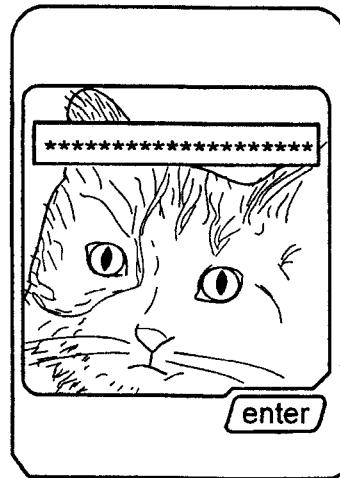


图31B

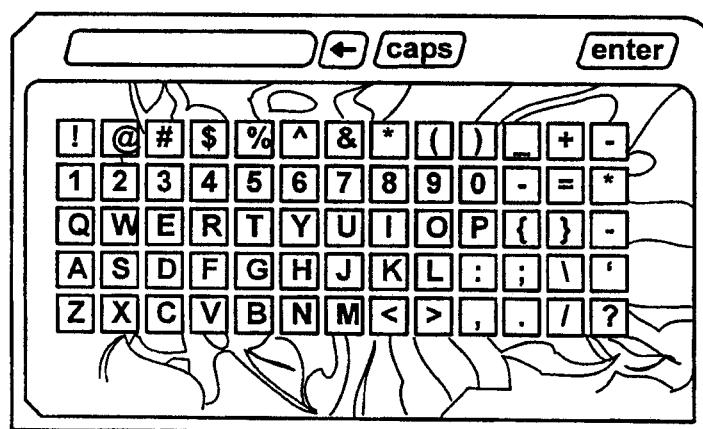


图31C

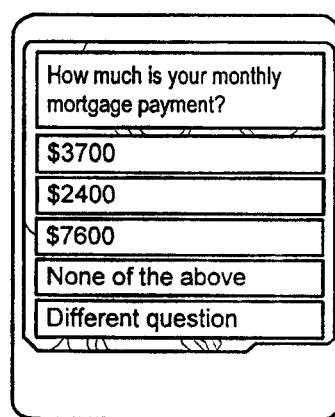


图31D

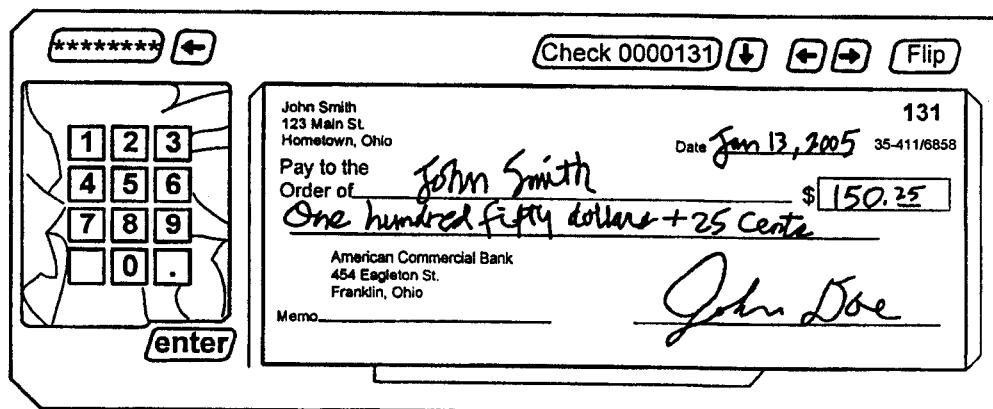


图31E