(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 111512590 A
(43)申请公布日 2020.08.07

(21)申请号 201880082736.4

(22)申请日 2018.12.06

(30)优先权数据
17208570.6 2017.12.19 EP

(85)PCT国际申请进入国家阶段日
2020.06.19

(86)PCT国际申请的申请数据
PCT/EP2018/083710 2018.12.06

(87)PCT国际申请的公布数据
WO2019/121026 EN 2019.06.27

(71)申请人 皇家飞利浦有限公司
地址 荷兰艾恩德霍芬

(72)发明人 S·J·A·德雷赫 A·佩斯特林

(74)专利代理机构 永新专利商标代理有限公司 72002
代理人 刘兆君

(51)Int.Cl.
H04L 9/00(2006.01)
H04L 9/32(2006.01)

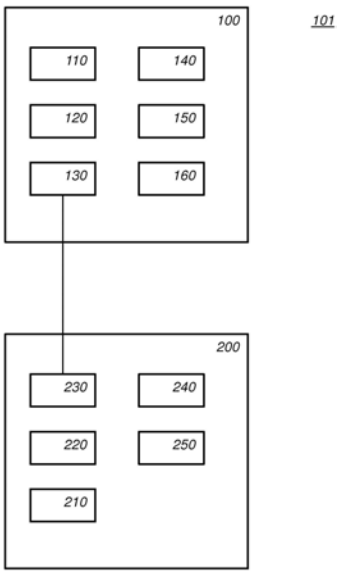
权利要求书2页 说明书14页 附图5页

(54)发明名称

用于密码认证的同态加密

(57)摘要

一些实施例涉及服务器设备(100)和客户端设备(200),其被布置为认证客户端设备(200)的用户。所述用户具有对认证字符串的访问。所述服务器设备(100)被配置为根据同态加密算法来对字符/位置数据的集合进行加密。所述客户端设备允许用户从加密的集合中选择子集,使用同态运算可以根据所述子集计算出验证号码。



CN 111512590 A

1.一种用于从客户端设备(100)认证用户的服务器设备(200),所述用户具有对认证字符串(326)的访问,所述认证字符串具有在多个位置(410)处的多个字符,所述字符选自字符集(420),所述服务器设备包括:

存储设备(110),其存储根据字符/位置数据的集合(310)的子集(320)计算出的第一验证号码(340,$g^{p}y$),所述子集由认证字符串(325,p)指示,字符/位置数据的所述集合(310,$g^{i\cdot 10n-1-j}$)包括针对来自所述字符集的字符与认证字符串中的位置的组合的数字(430);

处理器电路,其被配置为:

根据同态加密算法对字符/位置数据的所述集合进行加密,并将所加密的集合(350)发送到所述客户端设备;

从所述客户端设备接收第二验证号码(380),所述第二验证号码是通过对来自所加密的集合的子集的加密的第二验证号码进行同态计算来计算的,所述子集由所述认证字符串指示;并且

验证在所述第一验证号码与所述加密的第二验证号码之间的对应关系以认证所述用户。

2.根据权利要求1所述的服务器设备,其中,所述处理器电路被配置为:

生成所述字符集的置换(500);

获得针对所述字符集中的所述字符的显示数据(510、511),并且将所述显示数据与对应于所述字符的加密的字符/位置数据(530、520-529)相关联;

将所述显示数据与所述加密的字符/位置数据相关联地并且以根据所生成的置换的次序发送到所述客户端设备。

3.根据权利要求1所述的服务器设备,其中,

相同的显示数据用于所有位置,或者

针对至少两个不同的位置生成不同的显示数据。

4.根据权利要求2所述的服务器设备,其中,没有显示数据被发送到所述客户端设备以用于后续认证,所述客户端设备使用高速缓存的显示数据。

5.根据前述权利要求中的任一项所述的服务器设备,其中,所述第一验证号码利用盲数(y)而被盲化,并且其中,计算所述第二验证号码包括利用所述盲数进行盲化。

6.根据前述权利要求中的任一项所述的服务器设备,其中,所述字符/位置数据中的至少一些是随机数。

7.根据前述权利要求中的任一项所述的服务器设备,其中,

所述字符/位置数据中的至少一些被计算为基数的幂的倍数,

所述字符/位置数据中的至少一些被计算为幂,其中,指数是基数的幂的倍数。

8.根据前述权利要求中的任一项所述的服务器设备,其中,验证所述对应关系包括对所述第二验证号码进行解密。

9.根据前述权利要求中的任一项所述的服务器设备,其中,所述同态加密算法是概率同态加密算法。

10.根据前述权利要求中的任一项所述的服务器设备,其中,所述验证号码是Pederson承诺,并且/或者其中,所述加密是ElGamal加密。

11.一种用于向服务器设备认证用户的客户端设备,所述用户具有对认证字符串的访

问,所述认证字符串具有在多个位置处的多个字符,所述字符选自字符集,所述客户端设备包括:

处理器电路,其被配置为:

接收根据同态加密算法加密的字符/位置数据的集合,字符/位置数据的所加密的集合($g^{i\cdot 10^{n-1-j}}$)包括针对来自所述字符集的字符与认证字符串中的位置的组合的数字;

对来自字符/位置数据的所加密的集合的子集的加密的第二验证号码($g^{p}y$)进行同态计算,所述子集由所述认证字符串(p)指示,

将所述加密的第二验证号码发送到所述服务器设备。

12.根据权利要求11所述的客户端设备,其中,所述处理器电路被配置为:

从所述服务器设备接收与加密的字符/位置数据相关联的显示数据;

针对所述认证字符串的每个位置,显示接收到的显示数据;

接收用户输入,所述用户输入从所显示的显示数据中选择显示数据;

选择与所选择的显示数据相对应的子集。

13.根据权利要求11-12中的任一项所述的客户端设备,包括用于存储盲数的存储设备,所述处理器电路被配置为利用所述盲数(y)来对所述加密的第二验证号码进行盲化。

14.一种用于从客户端设备(100)认证用户的服务器方法(600),所述用户访问认证字符串(326),所述认证字符串具有在多个位置(410)处的多个字符,所述字符选自字符集(420),所述服务器方法包括:

存储(610)根据字符/位置数据的集合(310)的子集(320)计算出的第一验证号码(340,$g^{p}y$),所述子集由所述认证字符串(325,p)指示,字符/位置数据的所述集合(310,$g^{i\cdot 10^{n-1-j}}$)包括针对来自所述字符集的字符与认证字符串中的位置的组合的数字(430);

根据同态加密算法对字符/位置数据的所述集合进行加密(620),并且将所加密的集合(350)发送到客户端设备;

从所述客户端设备接收(630)第二验证号码(380),所述第二验证号码是通过对来自所加密的集合的子集的加密的第二验证号码进行同态计算来计算的,所述子集由所述认证字符串指示;并且

验证(640)所述第一验证号码与所述加密的第二验证号码之间的对应关系以认证所述用户。

15.一种用于向服务器设备认证用户的客户端方法(700),所述用户能够访问认证字符串,所述认证字符串具有在多个位置处的多个字符,所述字符选自字符集,所述客户端方法包括:

接收(710)根据同态加密算法加密的字符/位置数据的集合,字符/位置数据的所加密的集合($g^{i\cdot 10^{n-1-j}}$)包括针对来自所述字符集的字符和认证字符串中位置的组合的数字;

对来自字符/位置数据的所加密的集合的子集的加密的第二验证号码($g^{p}y$)进行同态计算(720),所述子集由所述认证字符串(p)指示;

将所述加密的第二验证号码发送(730)给所述服务器设备。

16.一种包括表示指令的瞬态或非瞬态数据(1020)的计算机可读介质(1000),所述指令用于使处理器系统执行根据权利要求14或15所述的方法。

用于密码认证的同态加密

技术领域

[0001] 本发明涉及客户端设备、服务器设备、服务器方法、客户端方法和计算机可读介质。

背景技术

[0002] 美国专利US 7731251公开了一种在通信网络上安全地通信信息的已知系统。该已知方法可以用于输入认证信息，例如PIN码。

[0003] 该已知系统包括通过远程通信网络耦合到网络服务器的远程计算设备。用于认证用户的网页包括虚拟面板，用户可以从该虚拟面板输入私有信息。虚拟面板包括可以从用户的远程计算设备的显示器中直接选择的多个字母数字键以输入私有信息。字母数字字符随机显示在虚拟面板中的不同位置。虚拟面板上显示的实际字符与存储在认证服务器上的数据库中的替代字符集相关。然后，使用安全套接字层(SSL)对来自替代字符集中与用户输入的私有信息相对应的字符进行加密，并将其通信给认证服务器。

[0004] 存在与已知系统相关联的若干问题。例如，网络服务器容易受到存储器刮取攻击。在解密来自用户的响应之后，用户的密码在存储器中可见。有权访问服务器的存储器的攻击者可以从中获得密码。同样，在客户端，攻击者可以记录从用户设备发送的数据，然后将其重播以向服务器模仿用户。

发明内容

[0005] 如权利要求中所定义地提供了客户端设备和服务器设备，其被布置用于认证。服务器设备使用同态加密来根据同态加密算法对字符/位置数据的集合进行加密。客户端设备根据加密的字符/位置数据计算验证号码。因为使用同态加密对字符/位置进行了加密，所以无需解密就可以对加密数据执行验证号码的计算。最后，只需要返回验证号码。服务器因此可以验证接收到的和存储的验证号码是否彼此一致，而不必知道选择了哪个字符/位置数据，也就是说，服务器不需要知道认证字符串，例如用户的密码或PIN码。

[0006] 在实施例中，服务器设备获得用于字符集中的字符的显示数据，并且将显示数据与对应于该字符的加密的字符/位置数据相关联。用户可以在客户端设备上选择与认证字符串中的字符相对应的显示数据。因此，即使在客户端设备上，在其存储器中也不需要认证字符串。显示数据优选地对于机器而言是难以读取的，例如，非机器可读的。即使很难读取显示数据，即使不是不可能，也会使自动攻击复杂化。

[0007] 本文描述的认证方法可以在广泛的实际应用中应用。这样的实际应用包括例如用于保护消息的真实性的通信系统，例如用于保护金融交易的金融系统，例如用于保护数据的真实性的存储系统，等等。

[0008] 根据本发明的方法可以在计算机上作为计算机实现的方法来实现，或者可以在专用硬件中或者以两者的组合来实现。用于根据本发明的方法的可执行代码可以存储在计算机程序产品上。计算机程序产品的示例包括存储器设备，光学存储设备，集成电路，服务器，

在线软件等。优选地，计算机程序产品包括存储在计算机可读介质上的非瞬态程序代码，用于在所述程序产品在计算机上运行时执行根据本发明的方法。

[0009] 在优选实施例中，计算机程序包括计算机程序代码，当所述计算机程序在计算机上运行时，所述计算机程序代码适于执行根据本发明的方法的所有步骤。优选地，所述计算机程序被体现在计算机可读介质上。

[0010] 本发明的另一方面提供了一种使计算机程序可用于下载的方法。当计算机程序被上传到例如苹果的App Store、谷歌的Play Store或微软的Windows Store时以及当可从此类商店下载计算机程序时，将使用该方面。

附图说明

[0011] 将仅通过示例的方式，参考附图来描述本发明的其它细节、各方面和实施例。为了简单和清楚起见示出了图中的元件，但元件不一定按比例绘制。在附图中，与已经描述的元件相对应的元件可以具有相同的附图标记。

[0012] 在附图中：

[0013] 图1示意性地示出了认证系统的实施例的示例；

[0014] 图2示意性地示出了数据依赖性的实施例的示例；

[0015] 图3示意性地示出了字符/位置数据的实施例的示例；

[0016] 图4示意性地示出了显示数据的实施例的示例；

[0017] 图5示意性地示出了服务器方法和客户端方法的实施例的示例；

[0018] 图6a示意性地示出了根据实施例的具有包括计算机程序的可写部分的计算机可读介质；

[0019] 图6b示意性地示出了根据实施例的处理器系统的表示。

[0020] 图1-4中附图标记列表：

[0021] 100 服务器设备

[0022] 101 认证系统

[0023] 110 验证号码存储设备

[0024] 120 加密单元

[0025] 130 通信接口

[0026] 140 比较单元

[0027] 150 置换生成器

[0028] 160 显示数据单元

[0029] 200 客户端设备

[0030] 210 显示器

[0031] 220 用户接口

[0032] 230 通信接口

[0033] 240 同态计算单元

[0034] 250 存储设备

[0035] 310 字符/位置数据的集合

[0036] 320 字符/位置数据集合的子集

[0037] 325、326 认证字符串

[0038] 330 第一非盲化验证号码

[0039] 340 第一验证号码

[0040] 345、346 盲数

[0041] 350 字符/位置数据的加密集合

[0042] 360 字符/位置数据的加密的集合的子集

[0043] 370 第二非盲化验证号码

[0044] 380 字符/位置数据的第二集合

[0045] 410 多个位置

[0046] 420 字符集

[0047] 430 字符/位置号码

[0048] 500 置换

[0049] 510 显示数据

[0050] 511 显示数据

[0051] 530 加密的字符/位置数据

[0052] 520-529 加密的字符/位置数据

具体实施方式

[0053] 虽然本发明可以有许多不同形式的实施例,但是在附图中示出并且在本文将详细描述一个或多个特定实施例,应理解,本公开被认为是本发明原理的示例,并不旨在将本发明限制于所示出和描述的特定实施例。

[0054] 在下文中,为了理解,在操作中描述实施例的元件。然而,显而易见的是,各个元件被布置为执行被描述为由它们执行的功能。此外,本发明不限于实施例,并且本发明在于本文描述或在互不相同的从属权利要求中叙述的每个新颖特征或特征的组合。

[0055] 流行的认证机制是用户名和密码(也称为用户证书)的组合。典型地,将证书发送到认证服务器进行验证。当攻击者获得某个用户的证书时,则他能够在服务器上模仿该用户,也就是说,攻击者可以让服务器相信他是他从其窃取证书的用户。遗憾的是,在实践中,许多用户在多个应用中使用相同的密码。因此,攻击者能够在更多应用中模仿用户,而不仅仅是在他破解的应用中。

[0056] 问题在于,当用户输入他的/她的证书时,一些攻击者能够读取存储器。这些攻击称为存储器刮取。现有的解决方案主要基于多因子认证,其中用户必须优选地在单独的设备上执行多个认证步骤。结果,仅刮取证书不足以使攻击者模仿服务器上的用户,但另一方面,用户会丢失敏感数据(他的密码,他很可能在多个应用中使用该密码)。

[0057] 一种可能的解决方案是确保密码永远不会出现在存储器中。例如,一种解决方案可以使用户使用安全键盘以以下方式输入数字序列:所按按钮的位置不会泄漏数字的值,并且由按下按钮产生的值不是明文中的值,而是该值的编码。

[0058] 该安全键盘确保从存储器刮取器中隐藏密码的所有信息(例如PIN)。这样的解决方案不会将敏感的个人密码泄露给这种攻击者。然而,攻击者可能只是将加密的密码刮取并重播给服务器,从而仍然模仿用户。

[0059] 为了向服务器进行认证，客户端通常通过安全信道发送其证书。然后，服务器使用其数据库来验证用户名和密码组合是否匹配。例如，数据库包含用户名u和对应密码$\mathcal{H}_U$的(加盐)散列的组合。当服务器获取用户名U和密码p时，服务器将计算H(p)并验证是否$H(p)=\mathcal{H}_U$。

[0060] 如果攻击者能够攻击服务器，则他可以找到包含用户证书(或其衍生物)的数据库。存在很好的解决方案来防止攻击者从数据库获取敏感密码，例如前面示例中使用密码散列。然而，当攻击者能够刮取活动服务器的存储器时，在大多数场景中，他将看到服务的存储器中的明文密码。在下面的实施例中，描述了在客户端设备和服务器设备上在认证期间使用同态加密来保护认证字符串(例如密码)。

[0061] 图1示意性地示出了认证系统101的实施例的示例。认证系统101包括客户端设备200和服务器设备100，它们已经被配置为使得客户端设备200可以向服务器设备100认证客户端设备200的用户。通过证明他知道认证字符串来认证用户。

[0062] 客户端设备200和服务器设备100可以是被配置为本发明的实施例的电子设备。客户端设备200包括通信接口130，所述通信接口130被布置为与服务器设备通信，并且服务器设备100包括通信接口130。这些通信接口被布置为例如通过计算机网络以数字格式彼此通信，例如通过交换数字消息。例如，计算机网络可以是互联网、内联网、WLAN等。计算机网络可以是因特网。例如，通信接口130和/或230可以是有线接口，例如以太网端口。例如，通信接口130和/或230可以是无线接口，例如Wi-Fi接口，例如，包括Wi-Fi天线。

[0063] 客户端设备200和服务器设备100合作，使得客户端设备可以向服务器设备100认证客户端设备200的用户。例如，服务器设备100可以被配置用于用户被授权而其他用户未被授权的某种服务。例如，服务器100可以是：文件服务器，其存储用户有权访问的文档；流服务器，其提供对有价值的内容的访问；网络商店，其中存储了用户的信用卡信息，等等。

[0064] 客户端设备200和服务器设备100各自包括处理器电路。它们还可以包括存储器，所述存储器包括计算机处理指令。所述处理器电路被布置为执行存储在存储器中的指令。认证的运行在处理器电路中实现，其示例在本文示出。图1示出了可以是处理器电路的功能单元的功能单元。例如，图1可以用作处理器电路的可能功能组织的蓝图。处理器电路未与图1中的单元分开示出。例如，图1中所示的功能单元可以全部或部分地以存储在设备100或200处(例如，在设备100或200的电子存储器中)的计算机指令实现，并由设备100或200的微处理器执行。在混合实施例中，功能单元部分地在硬件中实现，例如作为协处理器，例如密码协处理器，并且部分地在设备100或200上存储和执行的软件中实现。

[0065] 服务器设备100被配置为从客户端设备200认证用户。该用户访问认证字符串。认证字符串具有在多个位置处的多个字符，其中这些字符是从字符集中选择的。例如，认证字符串可以是PIN，在这种情况下，字符集是数字{0,1,2,...,9}。例如，认证字符串可以是字母数字字符串，在这种情况下，字符集可以是{a,…,z,A,…,Z,0,…,9}。此外，可以将一个或多个标点字符添加到字符集中，如认证系统101所希望的那样。

[0066] 服务器设备100包括被配置为存储第一验证号码的存储设备110，例如，验证号码存储设备110。在设置阶段，已经从认证字符串中计算出第一验证号码。然而，在实施例中，知道第一验证号码不足以获得认证字符串。此外，第一验证号码本身不能用于向服务器100

进行认证。换句话说，即使攻击者获得了第一验证号码，这也无助于他获得认证字符串，也无助于向客户端设备200进行认证。客户端设备200被配置为使得在认证过程期间，从认证字符串计算第二验证号码。客户端设备200可以这样做，因为用户基于他知道认证字符串来提供输入。然后将第二验证号码发送到服务器设备100。验证在第一验证号码和第二验证号码之间的对应关系，这反过来验证了两个验证号码是从相同的认证字符串计算出的。因此，用户被认证。

[0067] 验证号码存储设备110可以包括用于多个用户的多个验证号码。例如，验证号码可以与用户名相关联。例如，可以存储成对的验证号码和用户名。该关联可以通过用户名的派生物，例如散列或加盐散列。在这种情况下，验证号码存储设备110可以将验证号码与派生物一起存储。

[0068] 图2示意性地示出了可能的数据依赖性的实施例的示例。该图解释了在第一和第二验证号码与认证字符串之间的可能关系。

[0069] 图2示出了字符/位置数据的集合310。集合310包括针对来自字符集的字符与认证字符串中的位置的每种组合的数字。字符/位置数据可能是秘密的，但这不是必需的。这可以是公共数据。通常，例如从所使用的同态加密方案所使用的有限组、环或字段中选择数字。同态加密方案允许对例如组中的数据(例如数字)进行加密。此外，存在可以对加密数字执行的至少一个二进制运算。例如，支持加法的同态加密方案允许在加密时计算一组数字的总和。例如，对于这样的加密E，存在f，使得$f(E(x_1),\cdots,E(x_n))=E(x_1+\ldots+x_n)$。也就是说，可以从加密的被加数计算出总和的加密，此外，这可以在不知道解密密钥的情况下进行。一些同态加密方案可能支持乘法。还存在支持一种以上运算(例如加法和乘法)的同态加密。虽然在实施例中可以使用这样的方案，但是由于仅需要一种运算，所以不需要这样的复杂类型的同态加密。同态加密方案的示例包括：(未填充的)RSA，ElGamal，GoldwasserMicali，Benaloh，Paillier等。例如，数字可以从对质数取模的整数组中选择。其它示例是由某种生成器生成的椭圆曲线上的组。一组可逆数以复数为模，等等。

[0070] 图3示意性地示出了字符/位置数据的实施例的示例。示出了多个位置410。可以通过从字符集420中为认证字符串的每个位置选择字符来获得认证字符串。对于多个位置410中的位置的每个组合以及对于字符集420中的每个字符，选择字符/位置号码。如果存在n个位置和m个可能的字符，则通常存在nm个字符/位置号码。可能省略一些组合，但后果是在认证字符串中不可能使用那些字符/位置组合。

[0071] 在实施例中，认证字符串可以具有预定长度，或者认证字符串可以具有小于某个最大长度的任何长度，至少为2。然而，不要求认证字符串使用多个位置中的所有位置。例如，可以通过打开一些位置来支持不同长度的认证字符串。替代地，可以通过将一个字符编码为空位置来减少字符/位置数据的数量。通过在字符集中包括填充字符，还可以支持不同长度的认证字符串。可能要求认证字符串在认证字符串的末尾最多具有作为连续字符串的填充字符。图3示出了位置和字符的一种组合：数字430。

[0072] 在实施例中，字符/位置数据是随机数。使用随机数意味着存在可能存在给出相同验证号码的两个不同的认证字符串的小的风险。因此，在认证该用户时，可能存在多个认证字符串被接受。然而，发生这种情况的风险很小，并且替代认证字符串未知。此外，通过为同态加密选择较大的组等，可以使风险尽可能小。

[0073] 在实施例中，选择字符/位置数据的集合310，使得任何子集为同态加密所支持的运算产生不同的结果。例如，如果同态加密支持加法，则可以将字符/位置数据的集合310选择为基数的幂的倍数。可以选择基数作为字符集中字符的数量。例如，如果所支持的运算是加法，则该选择将为每个子集提供不同的总和。

[0074] 例如，如果同态加密支持乘法，则可以选择字符/位置数据的集合310为幂，其中指数是基数的幂的倍数。可以将基数选择为字符集中的字符的数量。例如，如果所支持的运算是乘法，则该选择将为每个子集提供不同的乘法。例如，幂可以是生成子的幂。选择字符/位置数据作为生成子的幂的另一优点在于，取幂用作额外的加密层。特别地，即使是非盲化的(见下文)，也较难从验证号码中获得认证字符串。

[0075] 在实施例中，字符/位置数小于组的大小。在实施例中，字符/位置数是例如生成子的基数的幂，并且指数小于组的大小或由生成子生成的组的阶。在实施例中，字符/位置数或指数小于组的阶除以字符集的大小。

[0076] 字符/位置数据可以是公开的，但不必是公开的。特别地，为了增加抵抗密码攻击的安全性，字符/位置数据对于客户端设备可以是私有的。例如，如果字符/位置数据是随机数据，则攻击者分析在服务器和客户端设备之间交换的数据得不到观察到的数据对应于哪个字符/位置数据。在设置阶段以及每个后续认证阶段都使用字符/位置数据的相同集合。可以针对每个用户随机选择字符/位置数据，因为服务器设备100支持认证。

[0077] 认证字符串325确定字符/位置数据的集合310的子集320。特别地，与实际在认证字符串的不同位置处使用的字符相对应的字符/位置数包含于子集320中。从子集320中得出第一验证号码340。

[0078] 例如，可以通过执行同态加密所支持的运算(例如相乘)来从子集320获得第一验证号码。为了增加安全性，所述结果可能会被盲化。例如，可以通过执行所支持的运算(例如，通过将子集中的数字相乘)来从子集320获得第一非盲化验证号码330。接下来，可以用盲数340来盲化第一非盲化验证号码330。后者是(盲化的)第一验证号码340。例如，所支持的运算可以用于包括盲数；例如，可以将盲数添加或乘以第一非盲化验证号码330。盲数是可选的。在没有盲数的情况下，获得第一非盲化验证号码330的攻击者可能会进行字典攻击来确定认证字符串。盲数减小非盲化验证号码330在认证字符串325上包含的信息，例如，可能将其减小为零。下面我们将假设第一个认证号码是盲化的，但这不是必需的。盲数可以是例如由随机数生成器生成的随机数。

[0079] 注意，服务器设备在认证阶段期间不需要认证字符串325、非盲化验证号码330或盲数345。例如，服务器设备可以在设置阶段期间计算第一验证号码340，然后删除不需要的信息。例如，服务器设备可以在设置阶段期间例如从客户端设备或从受信任的第三方接收第一验证号码340。在认证阶段期间，服务器设备需要字符/位置数据310和第一验证号码，无论是否为盲化。如果字符/位置数据310不是随机的，则可以在需要时计算而不是存储它。

[0080] 服务器设备100包括加密单元120。加密单元120被配置为根据同态加密算法对字符/位置数据的集合310进行加密。加密是在每元素的基础上的。在图3中，这被示为加密集合350。在认证期间，加密集合350被发送到客户端设备200。为了避免重放，当对于不同的认证该加密的结果是不同的时，这是优选的。例如，这可以通过在每次加密集合310时使用不同的密钥来确保，例如通过选择随机密钥。例如，这可以通过选择概率加密方案(或两者)来

确保。例如，加密方案可以是ElGamal加密，其是同态和概率加密方案。

[0081] 客户端设备200包括同态计算单元240。同态计算单元240被配置为从字符/位置数据的加密集合350的子集360计算加密的第二验证号码。从服务器设备100接收加密集合350。例如，客户端设备200的用户可以基于他/她对认证字符串326的了解来选择接收到的集合350的子集360。如果认证字符串325和认证字符串326是相同的，则子集360与子集320相同，不同之处在于子集360被加密而子集320未被加密。例如，接收到的集合350可以包括哪个加密的字符/位置号码对应于哪个字符/位置的指示。例如，集合350可以被排序。例如，用户可以在客户端设备200的用户接口220处(例如在触摸屏的键盘等处)输入认证字符串，以选择子集360。

[0082] 该指示可以是机器可读的或不是机器可读的。以机器可读的方式指示集合350中的哪个加密的字符/位置号码对应于哪个字符/位置的缺点在于，即使服务器设备100不是可刮取的，但客户端设备200可能变得可刮取。避免在服务器100处的攻击是值得的，因为在服务器100处的攻击通常意味着许多认证字符串被破坏。由于认证字符串经常被重用，因此这可能会具有很大的安全后果。下面提出了一种解决方案，其减少了在客户端设备200处攻击的风险。

[0083] 同态计算单元240被配置为从字符/位置数据的加密的集合的子集中同态计算加密的第二验证号码。例如，同态计算单元240可以执行与在集合320上进行的相同的计算以获得数字330，不同之处在于同态计算单元240对加密的数据执行计算。第二验证号码370被加密但未是非盲化的。

[0084] 如果使用盲化，则同态计算单元240可以被配置为以盲数346对加密的第二验证码进行盲化。如果同态计算单元240执行相同的计算和盲化(虽然对加密数据)，并且使用相同的盲数(但经过加密)，则第二验证号码380与第一验证号码340相同，虽然已加密。

[0085] 客户端设备200可以以几种方式获得加密的盲数。例如，可以在设置阶段期间获得加密的盲数，并为将在认证阶段使用的每个密钥加密。加密的盲数可以在设备200的存储设备中。在这种情况下，可以在认证期间使用相同的密钥或有限数量的密钥。替代地，加密方案可以是公共/私有密钥加密方案。客户端设备200可以接收公共密钥，所述公共密钥也用于加密集合310以获得集合350。在这种情况下，同态加密单元240可以对盲数本身进行加密。盲数可以从用户或从客户端设备200的存储设备250获得。

[0086] 将第二验证号码(无论是非盲数370还是盲数380)发送到服务器设备100。服务器设备100包括比较单元140。比较单元140被配置为验证在第一验证号码和加密的第二验证号码之间的对应关系，以认证用户。例如，如果客户端设备200执行与服务器设备100相同的计算(但是在加密数据上)，则比较单元140可以被配置为解密第二验证号码以将其与第一验证号码进行比较相等性。虽然这不是必须的，但是例如同态加密单元240可以配置为计算不同的数，例如，同态加密单元240可以配置为计算第一验证号码的两倍，或者向其添加常数。更一般地，如果v是非盲化验证号码，则同态加密单元可以对于常数a和b来计算av、v+b或av+b等。在相加计算之前或之后，这些可能会被盲数进行盲化。例如，同态加密单元240可以计算avy，并且比较单元140可以验证第二验证号码比第一验证号码大a倍。

[0087] 如上所述，以上实施例防止在服务器上进行刮取，但是可以在客户端设备上被刮取。在实施例中，如下提供了附加的可选保护。在实施例中，服务器设备100包括置换生成器

150和显示数据单元160。置换生成器150被配置为生成字符集的置换。置换生成器150可以为每个位置生成置换，但是也可以生成比位置少的置换，其中重新使用一些置换。例如，可以对每个位置使用相同的置换。优选地，为每个认证至少生成新的置换。服务器设备100可以包括随机数生成器，以生成字符/位置数据和/或置换。

[0088] 显示数据单元160被配置为获得字符集中的字符的显示数据。显示数据以非机器可读的方式描绘了字符集的字符。在实施例中，从显示数据存储中取回显示数据，或从外部显示数据生成器设备中取回显示数据。显示数据还可以在服务器设备100上生成。生成显示数据本身是已知的。例如，用于生成所谓的验证码的已知算法。例如，对于每个位置获得显示数据的不同集合。可能会生成比位置数少的显示数据集，在这种情况下，一些显示数据被重复使用。例如，相同的显示数据可以用于所有位置。甚至不需要为每个认证使用不同的显示数据。在这种情况下，客户端设备高速缓存在后续认证中使用的显示数据。显示数据会消耗相对大量数据，这会通过高速缓存而减少。显示数据可以是图像，例如，对于每个字符，可以存在图像。显示数据可以是电影剪辑。例如，电影使人们可以从图像或视频剪辑中识别字符，但计算机可能无法识别。至少要求攻击者分析图像或剪辑，这使攻击更容易出错，自动化程度更低并且更加资源密集。因此，对于攻击者，将攻击扩展到许多用户而不是几个选定的用户变得更加困难。例如，显示数据可以是jpg、png等，或mp4、avi等。

[0089] 显示数据与加密的字符/位置数据的集合350一起发送到客户端设备200。显示数据可以以置换次序发送，但与正确的字符/位置数据相关联。图4示出了用于PIN码的字符集的置换500。对于较大的字符集，可以这样做。置换500可以指示将字符集中的字符的显示数据发送到客户端设备200的次序。显示数据510也在图4中示出。显示数据510中示出的是以非机器可读的方式按照置换所指示的次序的字符集中的字符。在这种情况下，通过选择随机字体并以随机定向叠加两个黑条来生成显示数据。可以替代使用本领域中已知的生成非机器可读数据的任何其它方式。在显示数据下方，示出了对应于字符的对应的加密的字符/位置数据520-529。例如，如果针对第一位置生成了显示数据510，则字符/位置数据520-529是用于字符集中的字符和第一位置的组合的字符/位置数据。对于下一位置，使用新的字符/位置数据，并且可选地还可以使用新的置换和/或新的显示集合。在更安全的实施例中，每个位置具有其自己的字符/位置数据、显示集合和置换。

[0090] 客户端设备200被配置为从服务器设备接收与加密的字符/位置数据相关联的显示数据。客户端设备200包括显示器210。显示器210被配置为显示每个位置的接收到的显示数据。在显示了特定位置的显示数据之后，用户可以在认证字符串中选择与该位置相对应的正确字符。当选择显示数据时，还为子集360选择了相关联的加密的字符/位置。例如，如果认证字符串的第一字符为9，则用户可以选择显示数据511，并且加密的字符/位置数据521被添加到子集361。因此，用户可以选择正确的字符/位置数据，即使如此攻击者不知道它代表什么：字符/位置数据既被加密，又对应于机器可读的显示图像。

[0091] 例如，可以根据以下伪代码来配置客户端设备200。

[0092] 对于认证字符串中的每个位置，进行：

[0093] 显示针对位置接收到的显示数据

[0094] 接收用户对所显示的显示数据的选择

[0095] 将位置和所选字符的加密的字符/位置数据添加到子集360。

[0096] 设备100和200的各种实施例可以包括输入接口，所述输入接口可以从各种替代中选择。例如，输入接口可以是键盘、触摸屏、鼠标等。通过输入接口可以选择显示数据。

[0097] 诸如存储设备110和250的存储设备可以被实现为电子存储器，例如闪存，或磁存储器，例如硬盘等。存储设备110或250可以包括一起构成存储设备的多个分立存储器。存储设备还可以实现为本地存储设备或外部存储设备，例如云存储设备。在后一种情况下，设备包括读取和写入单元，以例如通过网络连接读取/写入外部存储设备。例如，一部分存储设备可以是安全存储设备，而其余的可以是普通存储设备。存储设备还可以包括临时存储器，例如RAM。

[0098] 通常，设备100和200各自包括微处理器(未单独示出)，所述微处理器执行存储在设备100和200处的适当软件；例如，所述软件可能已被下载和/或存储在对应的存储器中，例如，诸如RAM的易失性存储器或诸如闪存的非易失性存储器(未单独示出)。替代地，设备100和200可以全部或部分地实现于可编程逻辑中，例如，作为现场可编程门阵列(FPGA)。设备100和200可以全部或部分地实现为所谓的专用集成电路(ASIC)，即针对其特定用途而定制的集成电路(IC)。例如，可以例如使用诸如Verilog、VHDL等的硬件描述语言在CMOS中实现电路。

[0099] 在实施例中，设备100和200可以包括一个或多个电路。电路实现本文描述的对应单元。这些电路可以是处理器电路和存储电路，所述处理器电路执行在存储电路中以电子方式表示的指令。

[0100] 处理器电路可以以分布方式实现，例如，作为多个子处理器电路。存储设备可以分布在多个分布式子存储设备上。存储器的一部分或全部可以是电子存储器、磁存储器等。例如，存储设备可以具有易失性和非易失性部分。存储设备的一部分可以是只读的。

[0101] 图5示意性地示出了服务器方法500和客户端方法600的实施例的示例。

[0102] 服务器方法600被配置用于从客户端设备(例如客户端设备200)认证用户，所述用户可以访问认证字符串。认证字符串在多个位置具有多个字符，这些字符是从字符集中选择的。服务器方法600包括：

[0103] 存储(610)从字符/位置数据集合的子集计算出的第一验证号码，所述子集由认证字符串指示，字符/位置数据集合包括用于来自字符集的字符与认证字符串中的位置的组合的数字；

[0104] 根据同态加密算法来对字符/位置数据集合进行加密(620)，并将加密后的集合发送到客户端设备；

[0105] 从客户端设备接收(630)第二验证号码，通过对来自加密的集合的子集的加密的第二验证号码进行同态计算来计算该第二验证号码，所述子集由认证字符串指示；以及

[0106] 验证(640)在第一验证号码和加密的第二验证号码之间的对应关系以认证用户。

[0107] 客户端方法700被配置用于向服务器设备(例如，服务器设备100)认证用户。用户可以访问认证字符串。认证字符串在多个位置具有多个字符，这些字符是从字符集中选择的。客户端方法700包括：

[0108] 接收710根据同态加密算法加密的字符/位置数据的集合，所述字符/位置数据的加密集合包括用于来自字符集的字符与认证字符串中的位置的组合的数字；

[0109] 从字符/位置数据的加密的集合的子集中同态计算(720)加密的第二验证号码，所

述子集由认证字符串指示；

[0110] 将加密的第二验证号码发送(730)给服务器设备。

[0111] 方法600和700可以包括额外的操作。特别地，它们可以包括获得、显示和选择字符集的显示数据。两种方法之间的依赖关系用虚线箭头指示。

[0112] 如本领域技术人员将显而易见的，执行该方法的许多不同方式是可能的。例如，步骤的次序可以改变，或者一些步骤可以并行执行。此外，在步骤之间可以插入其它方法步骤。插入的步骤可以表示如本文所述的方法的改进，或者可以与该方法无关。例如，一些步骤可以至少部分地并行执行。此外，在开始下一步骤之前，给定步骤可能尚未完全完成。

[0113] 可以使用软件来执行根据本发明的方法，所述软件包括用于使处理器系统执行方法600或700的指令。软件可以仅包括由系统的特定子实体采取的那些步骤。该软件可以存储在合适的存储介质中，例如硬盘、软盘、存储器、光盘等。软件可以作为信号沿着有线、无线或使用数据网络(例如，互联网)发送。可以使该软件可用于在服务器上下载和/或远程使用。可以使用比特流来执行根据本发明的方法，所述比特流被布置为配置可编程逻辑(例如，现场可编程门阵列(FPGA))以执行该方法。

[0114] 应当理解，本发明还扩展到适于使本发明付诸实践的计算机程序，特别是在载体上或载体中的计算机程序。该程序可以是源代码、目标代码、中间源代码和目标代码的形式，例如部分编译的形式，或者是适合用于实现根据本发明的方法的任何其它形式。与计算机程序产品有关的实施例包括与所阐述的方法中的至少一种方法的每个处理步骤对应的计算机可执行指令。这些指令可以被细分为子例程和/或存储在可以静态或动态链接的一个或多个文件中。与计算机程序产品有关的另一实施例包括与所阐述的系统和/或产品中的至少一个的每个单元对应的计算机可执行指令。

[0115] 如上所述，基于用户名和密码的认证机制容易受到存储器刮取攻击。这些存储器刮取攻击意味着两个威胁。首先，模仿用户；其次，丢失敏感的个人信息。已知的解决方案不能同时解决这两种威胁。此外，已知服务器是有吸引力的目标，因为它们包含许多敏感密码。在下文中，还使用更数学化的语言描述了进一步和/或更详细的实施例。

[0116] 考虑安全的键盘。这可以防止由于存储器刮取而导致证书丢失。安全键盘可以看作是一系列表$\{\mathcal{T}\}$，其中每一行代表0到9之间的数字。这种表中有2列。一列具有用于应用的数据以显示数字，例如，ω_i，第二列具有一些任意值。我们用$\mathcal{T}_i$表示存储器中的值，作为轻按表$\mathcal{T}$的行i上的值的结果，即，$\mathcal{T}$的行i上第二列中的值。行被置换为隐藏它们所代表的真实值。每次用户输入数字时，都会将新表加载到存储器中，以确保移除链接信息，例如两个输入的数字是否相同。

[0117] 以下认证方案可以防止重放攻击，但不使用同态加密。我们假设服务器具有在其中可安全存储用户证书的数据库。这意味着数据库本身不会泄漏任何用户的(完整)证书。

[0118] 1、用户输入其用户名，并将该用户名(的派生物)发送到服务器。

[0119] 2、服务器验证用户是否注册，如果已注册，则执行以下操作：

[0120] (a)生成10×n的随机矩阵$\mathcal{R}$，使得每列中的所有条目都是唯一的。

[0121] (b)生成作用于集合{0,1,2,…,9}上的n个随机置换π_i。

[0122] (c)生成显示数据$\omega_0^i,\ldots,\omega_9^i$的n个集合。

[0123] (d)针对i＝0,…,n-1,定义包含行j上的对$(\omega^{i}_{\pi_i(j)},\mathcal{R}_{\pi_i(j),i})$的表$\mathcal{T}^{i}$。

[0124] (e)用刚刚随机选择的键盘$\mathcal{K}=\mathcal{T}^{0},\ldots,\mathcal{T}^{n-1}$进行回复。

[0125] 3、向用户呈现$\mathcal{K}$,然后输入他的PIN＝$p_0,\cdots,p_{n-1}$,产生字符串$\mathcal{S}=\mathcal{T}^{0}_{j_0},\ldots,\mathcal{T}^{n-1}_{j_{n-1}}$,所述字符串将被发送回服务器。

[0126] 4、服务器使用$\mathcal{S}$和$\mathcal{R}$并使用关系$\mathcal{R}_{p_i,j}=\mathcal{S}_j$,之后根据用户选择他的表$T^{i}$的行$j_i$中的第i个PIN数字($p_i$)的事实,来重构PIN。因此,我们得出结论$\pi_i(j_i)$满足$(\omega^{i}_{\pi_i(j_i)}=\omega^{i}_{p_i})$,使得$T^{i}_{j_i}=R_{\pi_i(j_i),i}=R_{p_i,i}$。

[0127] 5、最后,服务器使用其数据库验证PIN,并从其存储器中去除所有生成的数据。

[0128] 由于服务器会生成新的随机键盘,因此服务器只会接受一次$\mathcal{S}$。因此,攻击者无法重播$\mathcal{S}$以模仿用户。此外,这种随机性可确保用户的设备上的活动存储器刮取器在一段时间内不会收集到有关用户输入的PIN的任何有意义的信息。

[0129] 仅通过移除注册检查并将用户证书安全地存储在数据库中来替换服务器侧的验证部分,可以将上述过程还用于确保用户的注册免受存储器刮取。

[0130] 为了在服务器上禁用存储器刮取器以获得敏感的密码或PIN,我们提出以下建议。

[0131] 设G＝<g>由大素数阶p的g生成的组。假设服务器具有包含用户名$\mathcal{U}$和密码派生的数据库。

[0132] $$P(U)=g^{PIN_U}y_U$$

[0133] 其中,PIN_U是用户的PIN并且y_U是G的一些未知元素。由于P(U)是Pedersen承诺,从理论上讲是隐藏的信息,当y_U未知时P(U)不包含有关PIN的信息[1]。

[0134] 设$[[x]]_k$表示具有秘钥k的x的El-Gamal加密[2]。从而

[0135] $[[x]]_k=(g^r,h^rx)$,

[0136] 其中r是统一的随机值对p取模并且$h=g^k$。

[0137] 我们假设用户U已经存储了在P(U)中使用的y_U。

[0138] 认证现在按以下进行:

[0139] 1、用户输入其用户名,并将该用户名(的派生物)发送到服务器。

[0140] 2、服务器验证用户是否注册,如果已注册,则执行以下操作:

[0141] (a)生成随机密钥$k\in\mathbb{Z}_p$。

[0142] (b)生成10×n随机矩阵$\mathcal{R}$,其中,$\mathcal{R}_{ij}=\left[\left[g^{10^{n-1-j}*i}\right]\right]_k$。

[0143] (c)生成作用于集合{0,1,2,…,9}上的n个随机置换π_i。

[0144] (d)生成显示数据$\omega^{i}_{0},\ldots,\omega^{i}_{9}$的n个集合。

[0145] (e)针对i＝0,…,n-1,定义包含行j上的对$(\omega^{i}_{\pi_i(j)},\mathcal{R}_{\pi_i(j),i})$的表$\mathcal{T}^{i}$。

[0146] (f)用刚刚随机选择的键盘$\mathcal{K}=\mathcal{T}^{0},\ldots,\mathcal{T}^{n-1}$和公共密钥$h=g^k$进行回复。

[0147] 3、向用户呈现$\mathcal{K}$,然后输入他的$PIN_U=p_0,\cdots,p_{n-1}$,产生字符串$\mathcal{S}'=\mathcal{T}^{0}_{p_0},\ldots,\mathcal{T}^{n-1}_{p_{n-1}}$。

[0148] 4、用户从存储器取回y_U，并使用公共密钥P_k.生成随机加密$[[y_U]]_k$。

[0149] 5、用户用$\mathcal{S} = [[y_U]]_k * \prod_{i=0}^{n-1} S'_i$进行回复。

[0150] 6、服务器解密δ，并验证其是否匹配P(U)。

[0151] 7、最后，从他的存储器中去除所有生成的数据。

[0152] 对于集合{0,1,2,…,9}，代替10xn矩阵，可以使用其它字符集。

[0153] 正确性：注意，数字字符串$PIN_U = p_0, \ldots, p_{n-1}$可以用整数表示：

[0154] $$\sum_{i=0}^{n-1} 10^{n-i-1} p_i.$$

[0155] 此外，根据

[0156] $$S'_i = T^i_{j_i} = R_{\pi_i(j_i),i} = R_{p_i,i} = \left[\left[g^{10^{n-1-j}*p_i}\right]\right]_k$$

[0157] El-Gamal的同态表示

[0158] $$\prod_{i=0}^{n-1} S'_i = \prod_{i=0}^{n-1} \left[\left[g^{10^{n-i-1}*p_i}\right]\right]_k = \left[\left[g^{\sum_{i=0}^{n-1} 10^{n-i-1}*p_i}\right]\right]_k = [[g^{PIN_U}]]_k$$

[0159] 使得

[0160] $$\mathcal{S} = [[y_U]]_k * \prod_{i=0}^{n-1} S'_i = [[P(U)]]_k.$$

[0161] 注册：现在注册如下地进行：

[0162] 1、用户输入其用户名，并将该用户名(的派生物)发送到服务器。

[0163] 2、服务器验证用户没有注册，如果是这样，则执行以下操作：

[0164] (a)生成随机密钥$k \in \mathbb{Z}_p$。

[0165] (b)生成10xn随机矩阵$\mathcal{R}$，其中$\mathbb{R}_{ij} = \left[\left[g^{10^{n-1-j}*i}\right]\right]_k$。

[0166] (c)生成作用于集合{0,1,2,…,9}上的n个随机置换π_i。

[0167] (d)生成显示数据$\omega^i_0, \ldots, \omega^i_9$的n个集合。

[0168] (e)针对i＝0,…,n-1，定义包含行j上的对$(\omega^i_{\pi_i(j)}, \mathcal{R}_{\pi_i(j),i})$的表$\mathcal{T}^i$。

[0169] (f)用刚刚随机选择的键盘$\mathcal{K} = \mathcal{T}^0, \ldots, \mathcal{T}^{n-1}$和公共密钥$h = g^k$进行回复。

[0170] 3、向用户呈现$\mathcal{K}$，然后输入他选择的$PIN_U = p_0, \ldots, p_{n-1}$，得到字符串$\mathcal{S}' = \mathcal{T}^0_{j_0}, \ldots, \mathcal{T}^{n-1}_{j_{n-1}}$。

[0171] 4、用户生成池化随机数y_U，并使用公共密钥h来生成随机加密$[[y_U]]_k$。

[0172] 5、用户用$\mathcal{S} = [[y_U]]_k * \prod_{i=0}^{n-1} S'_i$进行回复。

[0173] 6、服务器将δ解密到P(U)中，并存储用户U以及P(U)。

[0174] 7、最后，从他的存储器中去除所有生成的数据。

[0175] 服务器零知识：服务器仅看到P(U)的随机加密和P(U)本身。从El-Gamal[3]的语义安全性出发，可以推断出，将随机加密的y与键盘结果相乘也会将用户在键盘上所做的所有选择向服务器隐藏。这也称为盲化加密。此外，Pedersen承诺确保P(U)不能用于提取PIN。

[0176] 注意：作为椭圆曲线的两个素域都可以用来定义实现上述实施例的适当组。

[0177] 参考文献

[0178] [1]Pedersen.:Non-interactive and information-theoretic secure verifiable secret sharing.In Advances in Cryptology CRYPTO'91Springer

[0179] [2]T.ElGamal.:A public key cryptosystem and a signature scheme based on discrete logarithms.In Proceedings of CRYPTO 84on Advances in Cryptology, pages 10--18,New York,NY,USA,1985.Springer-Verlag New York,Inc.

[0180] [3]Y.Tsiounis and M.Yung.:On the security of ElGamal based encryption.In H.Imai and Y.Zheng,editors,Public Key Cryptography,volume 1431of Lecture Notes in Computer Science,pages 117--134.Springer,1998.

[0181] 图6a示出了根据实施例的具有包括计算机程序1020的可写部分1010的计算机可读介质1000,所述计算机程序1020包括用于使处理器系统执行客户端方法或服务器方法的指令。计算机程序1020可以作为物理标记或借助于计算机可读介质1000的磁化而体现在计算机可读介质1000上。然而,任何其它合适的实施例也是可以想到的。此外,可以理解,虽然计算机可读介质1000在这里被示为光盘,但是计算机可读介质1000可以是任何合适的计算机可读介质,例如硬盘、固态存储器、闪存等,并且可以不可记录的或可记录的。计算机程序1020包括用于使处理器系统执行根据实施例的客户端方法或服务器方法的指令。

[0182] 图6b以示意图示出了根据实施例的处理器系统1140。处理器系统包括一个或多个集成电路1110。在图6b中示意性地示出了一个或多个集成电路1110的架构。电路1110包括处理单元1120,例如CPU,用于运行计算机程序部件以执行根据实施例的方法和/或实现其模块或单元。电路1110包括用于存储编程代码、数据等的存储器1122。该编程代码将处理器系统配置为客户端设备或服务器设备的实施例。

[0183] 存储器1122的一部分可以是只读的。电路1110可以包括通信元件1126,例如天线、连接器或两者,等等。电路1110可以包括专用集成电路1124,用于执行该方法中定义的部分或全部处理。处理器1120、存储器1122、专用IC 1124和通信元件1126可以经由互连1130(例如,总线)彼此连接。处理器系统1110可以被布置为分别使用天线和/或连接器进行接触和/或无接触通信。

[0184] 例如,在实施例中,客户端设备、服务器设备、密钥生成设备和/或签名验证设备可以包括处理器电路和存储器电路,处理器被布置为执行存储于存储器电路中的软件。例如,处理器电路可以是Intel Core i7处理器、ARM Cortex-R8等。在实施例中,处理器电路可以是ARM Cortex M0。存储器电路可以是ROM电路,或者可以是非易失性存储器,例如闪存。存储器电路可以是易失性存储器,例如SRAM存储器。在后一种情况下,设备可以包括非易失性软件接口,例如,硬盘驱动器、网络接口等,其被布置用于提供软件。

[0185] 应当注意,上述实施例说明而非限制本发明,并且本领域技术人员将能够设计许多替代实施例。

[0186] 在权利要求中,放在括号之间的任何参考符号不应解释为对权利要求的限制。动词"包括"及其词形变化的使用不排除存在权利要求中未提及的其它元素或步骤。元件前面的词语"一"或"一个"不排除存在多个这样的元件。本发明可以通过包括若干不同元件的硬件以及通过适当编程的计算机来实现。在列举若干单元的设备权利要求中,这些单元中的若干单元可以由同一硬件来体现。尽管特定措施是在互不相同的从属权利要求中记载的,

但是这并不意味着不能有利地使用这些措施的组合。

[0187] 在权利要求中,括号内的引用是指示例实施例的附图中的附图标记或实施例的公式,因此增加了权利要求的可理解性。这些引用不应被解释为限制权利要求。

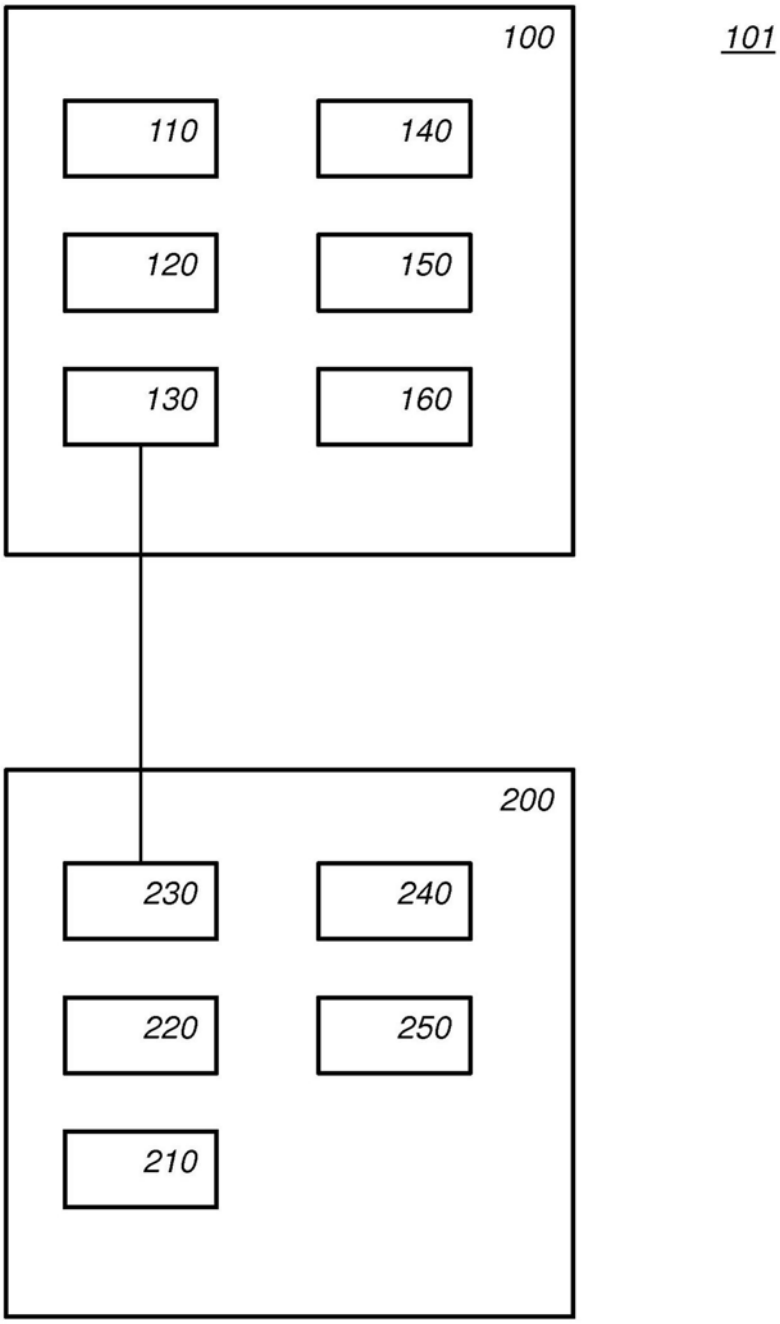


图1

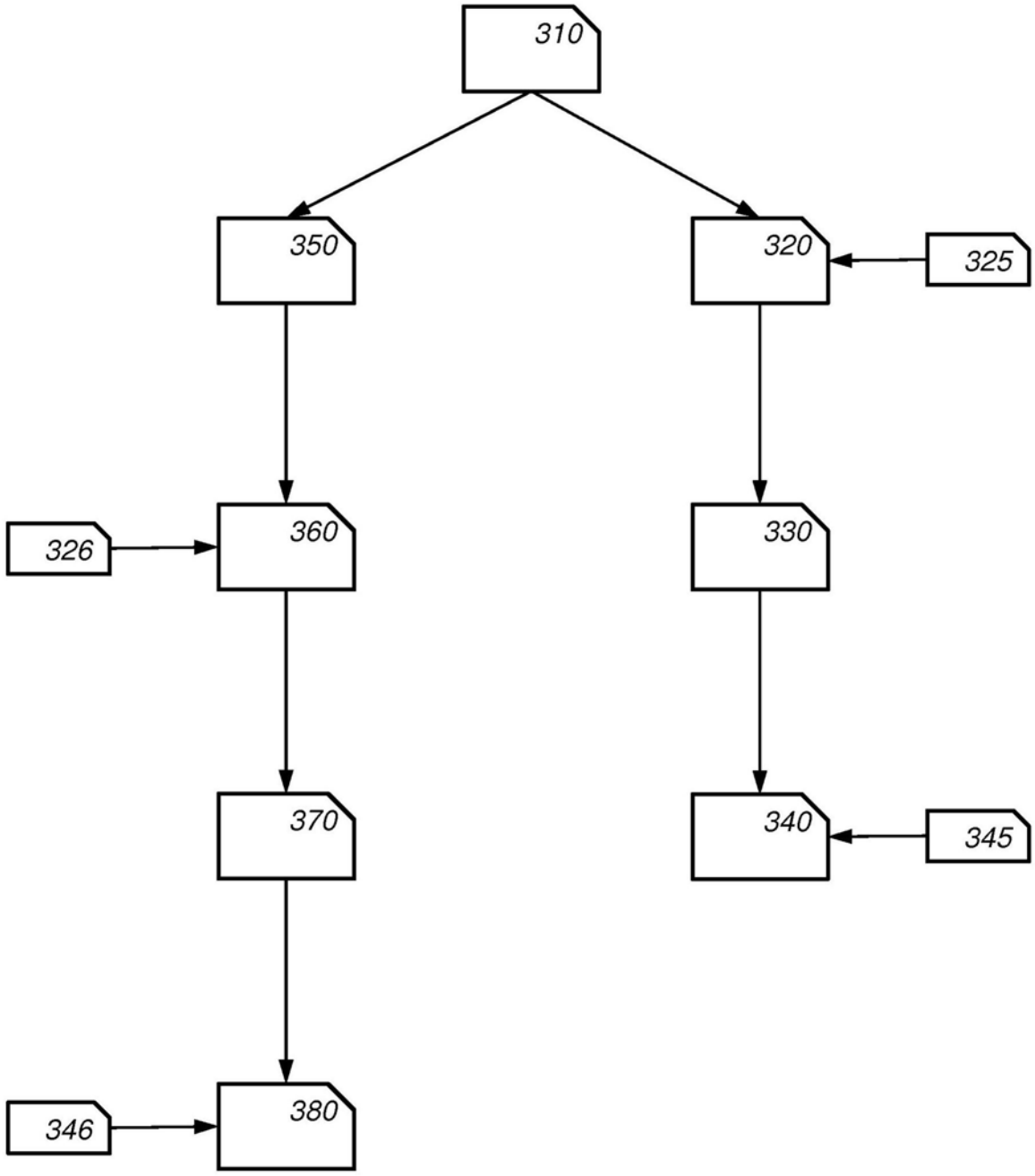


图2

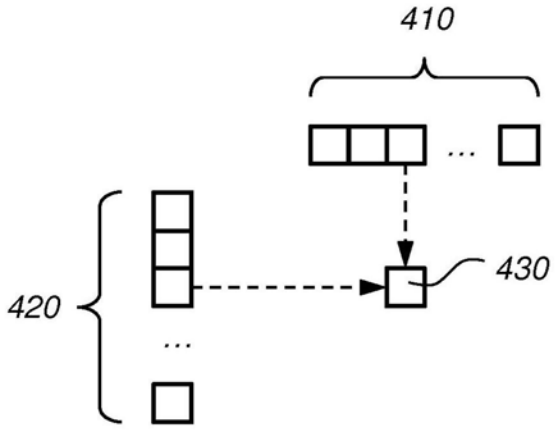


图3

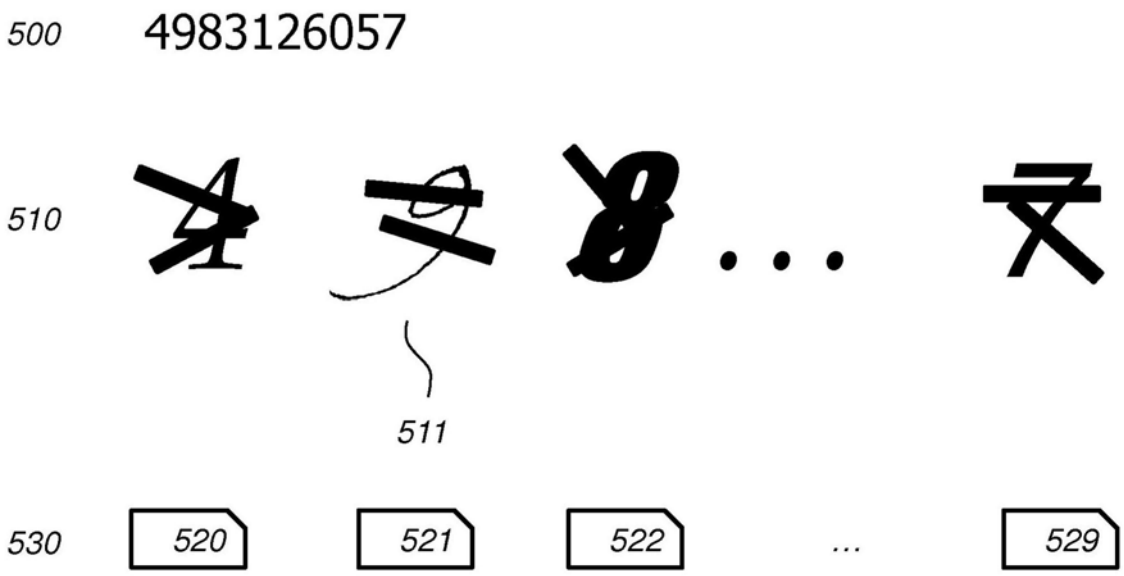


图4

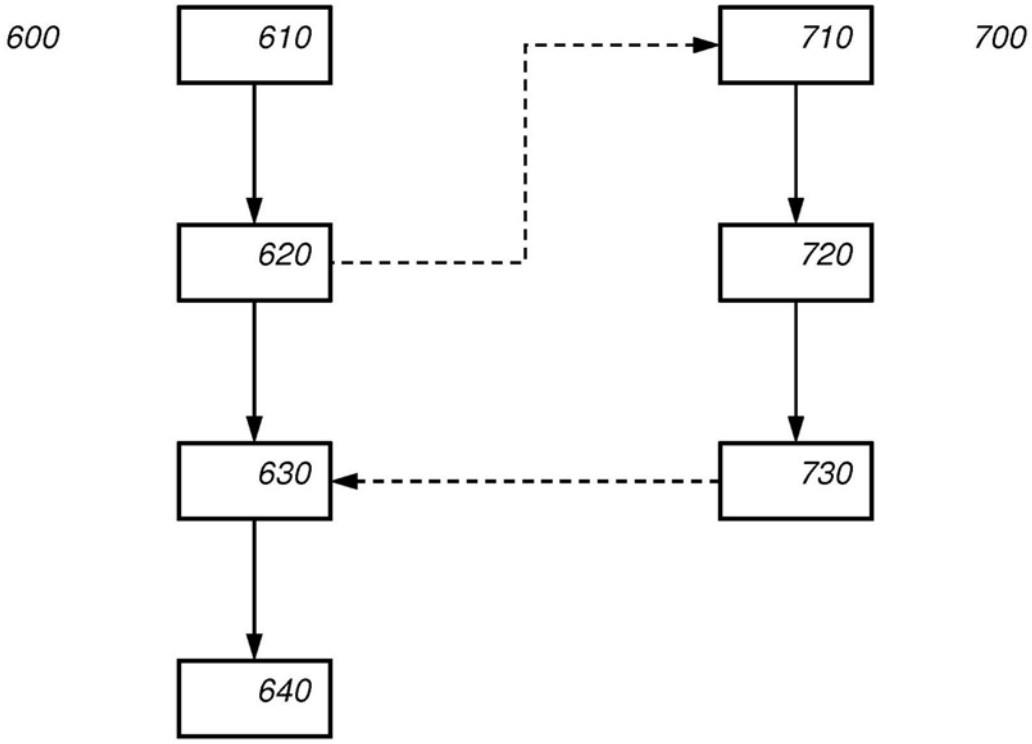


图5

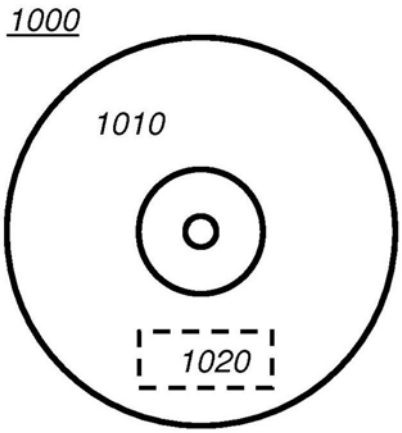


图6a

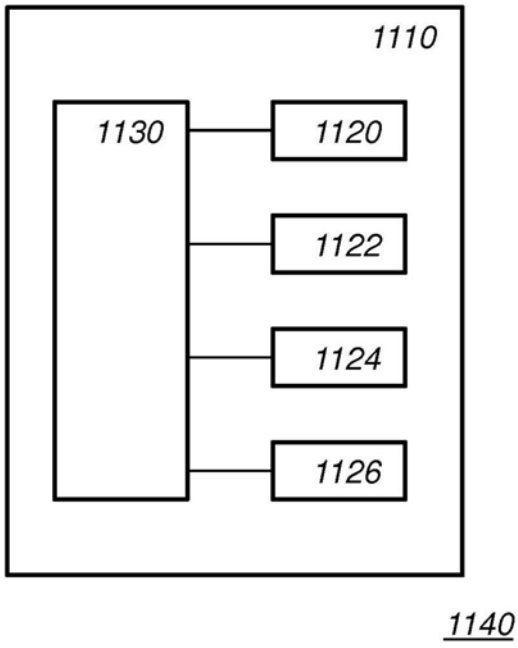


图6b