(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 115396139 B
(45)授权公告日 2024.11.29

(21)申请号 202210691523.X
(22)申请日 2016.09.28
(65)同一申请的已公布的文献号
申请公布号 CN 115396139 A
(43)申请公布日 2022.11.25
(30)优先权数据
14/925,769 2015.10.28 US
14/931,613 2015.11.03 US
(62)分案原申请数据
201680062621.X 2016.09.28
(73)专利权人 倪敏
地址 美国佐治亚州
(72)发明人 倪敏
(74)专利代理机构 余姚德盛专利代理事务所(普通合伙) 33239
专利代理师 周积德
(51)Int.Cl.
H04L 9/40(2022.01)
(56)对比文件
CN 102202067 A,2011.09.28
CN 102742211 A,2012.10.17
审查员 邹丽

权利要求书2页 说明书27页 附图22页

(54)发明名称

密码防盗的认证及加密的系统和方法

(57)摘要

本发明通过验证码和符章提供了一种防盗认证和加密的系统和方法,它利用销组成验证码,每个销都是一组符号中的一个,并通过由组成验证码的符号组中的至少两个符号组成符章。多个符章(一个“符章组”)被呈现给用户,用户预选的销(符号)被随机地插入部分或全部的符章中。用户对验证码中每个销选择符章组中的一个符章。通过用户选择的符章认证用户。因为每个所选符章中可能包括或不包括用户验证码预选的销中的一个,并且还包括了不属于该用户验证码预选销的其它随机产生的符号,所以即使有人观察用户选择哪些符章,他也不能确定用户的实际验证码是什么。

CN 115396139 B

1.一种利用预定的电子存储的验证码的信息加密/解密方法,该验证码包括从一组符号[符号集]中选择的预定数量的验证码符号,其中每个验证码符号由预定的销位置来表征,该方法包括:

接收待加密的原始信息,其中所述原始信息包括在相应信息元素位置中的预定数量的原始信息元素,其中所述原始信息元素选自一组信息元素;

将从该组信息元素中随机选择的随机信息元素在随机信息元素位置插入到原始信息中,以生成加密信息;

基于所述验证码为每个原始和随机信息元素生成符章组,其中每个符章组包括至少两个符章,其中每个符章组中的每个符章包括属于所述符号集的至少两个符号;和

基于验证码为加密信息中的每个原始和随机信息元素生成各自的密钥,其中每个密钥包括从为每个原始和随机信息元素生成的符章组中选择的至少两个密钥符章,由此为每个原始信息元素生成有效密钥并且为每个随机信息元素生成无效密钥;

其中,在解密过程中,每当信息元素的对应密钥有效时,如果满足下述条件,则加密信息中的信息元素被确定为有效信息元素:

信息元素的对应密钥中的密钥符章的数量等于密码中的符号数量;

信息元素的对应密钥中的至少一个密钥符章包含相应的一个验证码符号;和

对于每个销位置,所述销位置处的验证码符号或者包含在为该销位置选择的密钥符章中,或者根本不包含在信息元素的符章组中的任何符章中。

2.根据权利要求1所述的方法,其中为每个原始和随机信息元素生成的符章组包括随机插入到一些或所有符章中的一些或所有验证码符号。

3.根据权利要求1所述的方法,其中所述符号集被分成至少两个子集,并且其中每个符章和每个密钥符章包括来自所述至少两个子集中的一个符号。

4.根据权利要求3所述的方法,其中每个子集中的符号数量等于所述符章组中的符章数量,或者其中每个子集中的符号数量大于符章组中的符章数量。

5.根据权利要求1所述的方法,其中每个密钥符章包括属于符号集的四个符号,其中所述符号集被分成四个子集,并且其中每个符章和每个密钥符章包括来自四个符号子集中的一个符号。

6.根据权利要求1所述的方法,其中每个密钥符章包括属于符号集的五个符号。

7.根据权利要求6所述的方法,其中所述符号集被分成五个子集,并且其中每个符章和每个密钥符章包括来自五个符号子集中的一个符号。

8.根据权利要求1所述的方法,其中该组信息元素基于Unicode系统。

9.一种用于加密/解密信息的系统,该系统利用预定的电子存储的验证码,该验证码包括从一组符号[符号集]中选择的预定数量的验证码符号,其中每个验证码符号由预定的销位置来表征,该系统包括:

处理器;

处理器可访问的存储器;和

认证/加密模块(110),包括存储在存储器中的一组计算机可读指令,可由处理器执行以:

接收要加密的原始信息,其中原始信息包括在相应信息元素位置中的预定数量的原始

信息元素,其中原始信息元素选自一组信息元素;

将从该组信息元素中随机选择的随机信息元素在随机信息元素位置插入到原始信息中,以生成加密信息;

基于所述验证码为每个原始和随机信息元素生成符章组,其中每个符章组包括至少两个符章,其中每个符章组中的每个符章包括属于该符号集的至少两个符号;和

基于所述验证码为加密信息中的每个原始和随机信息元素生成各自的密钥,其中每个密钥包括从为每个原始和随机信息元素生成的符章组中选择的至少两个密钥符章,由此为每个原始信息元素生成有效密钥,并且为每个随机信息元素生成无效密钥;

其中,在解密过程中,每当信息元素的对应密钥有效时,如果满足下述条件,则加密信息中的信息元素被确定为有效信息元素:

信息元素的对应密钥中的密钥符章的数量等于密码中的符号数量;

信息元素的对应密钥中的至少一个密钥符章包含相应的一个验证码符号;和

对于每个销位置,所述销位置处的验证码符号或者包含在为该销位置选择的密钥符章中,或者根本不包含在信息元素的符章组中的任何符章中。

10.根据权利要求9所述的系统,其中为每个原始和随机信息元素生成的符章组包括随机插入到一些或所有符章中的一些或所有验证码符号。

11.根据权利要求9所述的系统,其中所述符号集被分成至少两个子集,并且其中每个符章和每个密钥符章包括来自所述至少两个子集中的一个符号。

12.根据权利要求11所述的系统,其中每个子集中的符号数量等于符章组中的符章数量,或者其中每个子集中的符号数量大于符章组中的符章数量。

13.根据权利要求9所述的系统,其中每个密钥符章包括属于符号集的四个符号。

14.根据权利要求13所述的系统,其中所述符号集被分成四个子集,并且其中每个符章和每个密钥符章包括来自四个符号子集中的一个符号。

15.根据权利要求9所述的系统,其中每个密钥符章包括属于符号集的五个符号,并且可选地,其中所述符号集被分成五个子集,并且其中每个符章和每个密钥符章包括来自五个符号子集中的一个符号,并且可选地,其中信息元素集基于Unicode系统。

密码防盗的认证及加密的系统和方法

[0001] 本申请是申请号为201680062621X、申请日为2016年09月28日、发明名称为“密码防盗的认证及加密的系统和方法”的发明专利的分案申请。

发明背景

发明领域

[0002] 本发明涉及认证及加密的系统和方法,更具体地讲,涉及密码防盗的认证及加密的系统和方法。

[0003] 相关技术的背景

[0004] 在现代社会中,日常生活需要使用各种信息设备,如移动电话,个人电脑,笔记本电脑,和自动取款机,如此之类。这些信息设备可以储存用户的个人数据。鉴于保护个人数据的重要性,有方法可以安全地锁定和解锁这些设备。

[0005] 目前,最常用的方法,以锁定和解锁这些设备是基于密码的认证过程,一个设备通常,在使用其服务之前,要求用户输入一个用户名称和用于识别身份的密码。这被称为是登录。这个登录过程的目的是防止用盗窃或欺诈手段改变用户的个人数据。

[0006] 随着每天网络更新范围和使用方便性的快速增加,黑客可更容易的窃取用户的密码从而得到他们的私人信息。此外,黑客们变得越来越有能力猜测和破解用户的密码。因此,简单的密码不再对网络威胁和间谍活动提供足够的保护。

[0007] 鉴于此,各种机制已被实施,以提供更好的保护。例如,用户需要创建一个满足密码长度,复杂性和不可预测性要求的密码,使密码的强度,理论上足以抵挡强力搜索攻击和字典攻击。而且,要求用户定期更改他们的密码,费弃旧密码,从而降低他们的密码被破解的可能性。这些机制能在一定程度上加强安全性,从而帮助用户保护自己的帐户。

[0008] 然而,每个组织都可能有一套不同的密码规则。一些需要密码长度至少为6个或8个字符。有些需要混合使用大小写字母以及数字。有些需要至少一个特殊字符,但有些不允许有特殊字符,所以当你觉得你已有了一个非常强大的可以在任何地方使用的金色密码时,又会有下一个具有不同要求的地方,使你的金色密码失效。

[0009] 由于这些不同的密码规则,用户很难,甚至不可能记住他们在不同网站/机构设置的众多密码。因此,用户通常把自己的密码存储在运行他们信息的设备里或应用程序的文件里。

[0010] 这些存储着的密码很可能被黑客有针对性的窃取,如果他们能进入这些存储密码的设备,他们将有机会获得所有的密码并进入所有用那些密码保护的账户/网站。因此实施严格的密码规则以避免过于薄弱的密码可能适得其反(增加更多暴露信息的风险)。

[0011] 鉴于传统密码的这些问题,有新的方法应运而生来解决这些问题。这些方法可能包括,但不限于,使用照片,图形图像,或者不同的形状和色调,使黑客更难偷看或窃取。一些技术甚至用手势及信息定位输入画面中的特定位置,以验证用户身份。然而这些方法都不能躲开一个隐藏的摄像头,它可以记录用户每次登录的一举一动。如果黑客可以播放所

有的记录并分析用户的一举一动,黑客最终还是可以获得用户的密码。

[0012] 现有认证方法的主要问题是:

[0013] (1)传统的密码和安全提问(最常用的法)不能防止偷看;

[0014] (2)图形图像和照片为基础的方法会要求用户上载图像或照片文件,并且系统必须保存并保持图像和/或照片。这不仅增加了用户和系统的负担,而且如果黑客摄像并回放登录过程,图像还是可以被认出的;

[0015] (3)新的图形,手势或基于位置的认证方法只能用于人和计算机之间使用,因而不能用于机器对机器的认证。

[0016] 因此,需要一种不出现上述问题的认证和加密的系统及方法。

发明内容

[0017] 本发明的一个目的是至少解决上述问题的缺点,并至少提供以下描述的优点。

[0018] 因此,本发明的一个目的是提供一种系统和方法,用于用户的认证。

[0019] 本发明的另一个目的是提供一种系统和方法,用于试图访问电子设备的用户的认证。

[0020] 本发明的另一个目的是提供一种系统和方法,用于该被请求访问的电子存储的信息的用户的身份认证。

[0021] 本发明的另一个目的是提供一种系统和方法,用于该被请求访问的装置的网络上的用户的认证。

[0022] 本发明的另一个目的是提供一种系统和方法,利用包括有预定数目的符号的验证码对用户身份的认证。

[0023] 本发明的另一个目的是提供一种系统和方法,利用多个符章对用户身份的认证,各个符章中至少有两个,用于创建用户验证码的符号。

[0024] 本发明的另一个目的是提供一种系统和方法,用于加密和解密电子信息。

[0025] 本发明的另一个目的是提供一种系统和方法,利用包括用于加密和解密电子信息的一个有预定符号数量的验证码,加密和解密电子信息。

[0026] 本发明的另一个目的是提供一种用于加密和解密电子信息的系统和方法,其利用包括有预定数目的符号与多个符章的组合,其中每个符章包括至少两个用于创建验证码的符号。

[0027] 为了实现至少部分或全部上述目的,有一种方法可用预先选出的一定数量的验证码(“验证码符号”)认证用户,它由预选的一组符号组成,其中,每个所述预定的验证码符号的特征在于每个预定的符号的位置,这一步骤包括呈现给用户一组符章,其中,所述符章组包括至少两个符章,而且符章组中的每一符章包括至少两个预定符号组中的符号,并要求用户从符章组中选出和预定的验证码符号位置相应的符章,并以该用户选出的符章来验证用户。

[0028] 为了实现至少部分或全部上述目的,还有一种系统可用预先选出的一定数量的验证码(“验证码符号”)认证用户,它由预选的一组符号组成,其中,每个所述预定的验证码符号的特征在于每个预定的符号的位置,它包括一个处理器,由处理器可访问的存储器,并包括一组计算机可读的存储在存储器中的指令的认证/加密模块,以用来向用户提供一组符

章,其中,所述符章组包括至少两个符章,而且符章组中的每一符章包括至少两个预定符号组中的符号,并要求用户从符章组中选出和预定的验证码符号位置相应的符章,并以该用户选出的符章来验证用户。

[0029] 其它优点,目的,以及本发明的特征的一部分将在下面描述,另一部分将会在本领域的普通技术人员对本发明的审查和实践中,因变得显而易见而学到。本发明的目的和优点可以如所附权利要求中特别指出的那样实现和获得。

附图说明

[0030] 本发明将被详细地进行说明,其中类似的附图标记指代以下类似的部分:

[0031] 图1A是一个描绘防盗验证码认证/加密系统的示例性框图,该系统可被用于一个设备或由客户机系统访问的服务器,是本发明的一个实例;

[0032] 图1B是一个描绘防盗验证码认证/加密系统的示例性框图,该系统被用于一个设备,是本发明的一个实例;

[0033] 图1C是一个描绘防盗验证码认证/加密系统的示例性框图,该系统被用于一个由客户端设备通过网络访问的服务器,是本发明的一个实例;

[0034] 图1D是一个描绘用硬件实现防盗验证码认证/加密系统的示例性框图,是本发明的一个实例;

[0035] 图2A是一个描绘被分为四维的符号的样板图示,是本发明的一个实例;

[0036] 图2B是一个描绘被分为五维的符号的样板图示

[0037] 图3是一个由认证/加密模块实施的示例性的,用于使用户能够选择验证码的流程图,是本发明的一个实例;

[0038] 图4是一个由认证/加密模块实施的示例性的,用于认证用户的流程图,是本发明的一个实例;

[0039] 图5是一个被认证/加密模块使用的示例性符章生成规则的表格,是本发明的一个实例;

[0040] 图6是一个被认证/加密模块使用的示例性符章选择规则的表格,是本发明的一个实例;

[0041] 图7是一个被认证/加密模块使用的示例性符章验证规则的表格,是本发明的一个实例;

[0042] 图8A和8B是在GATE_4实例中用户名创建(注册)过程的样本屏幕图示,是本发明的一个实例;

[0043] 图9A-9D是在GATE_4实例中用户登录过程的样本屏幕图示,是本发明的一个实例;

[0044] 图10A-10D是在GATE_4实例中用户登录过程的文字格式样本屏幕图示,是本发明的一个实例;

[0045] 图11A和11B是在GATE_5实例中用户名创建(注册)过程的样本屏幕图示,是本发明的一个实例;

[0046] 图12A-12D是在GATE_5实例中用户登录过程的样本屏幕图示,是本发明的一个实例;

[0047] 图13A-13D是在GATE_5实例中用户登录过程的文字格式样本屏幕图示,是本发明

的一个实例;

[0048] 图14是把未加密信息通过GATE_4使用发送端验证码进行加密处理的样本屏幕图示,是本发明的一个实例;

[0049] 图15A和15B是使用GATE_4实施成功解密处理的样本屏幕图示,是本发明的一个实例;

[0050] 图16A和16B是使用GATE_4实施解密处理,其中一个加密的假信息在接收端信息解密处理无效的,样本屏幕图示,是本发明的一个实例;

[0051] 图17是使用GATE_4实例屏幕截图,其中解密失败,因为接收器的验证码与发送器的验证码不同,是本发明的一个实例;

[0052] 图18是把未加密信息通过GATE_5使用发送端验证码加密处理的样本屏幕图示,是本发明的一个实例;

[0053] 图19A和19B是使用GATE_5成功实施解密处理的样本屏幕图示,是本发明的一个实例;

[0054] 图20A和20B是使用GATE_5实施解密处理,其中一个加密的假信息在接收端信息解密处理无效,的样本屏幕图示,是本发明的一个实例;

[0055] 图21是使用GATE_5实例屏幕截图,其中解密失败,因为接收器的验证码与发送器的验证码不同,是本发明的一个实例;

[0056] 具体实施方式的详细描述

[0057] 本发明提供了一个由特殊编程装置实现的新颖验证码防盗的认证和加密机制。认证时本发明利用设备中作为操作系统一部分的元代码组成的“验证码”。例如,一个验证码可能是这样的:Ⓘ♥ 2✉

[0058] 对于用户,上述验证码可能意味着“我喜欢发电子邮件”,这很容易记住,但别人很难知道。验证码中的每个符号将被称为“销”,相对于其它销具有一个相应的销的位置。在上面的例子中,符号“Ⓘ”在第一个销的位置,符号“♥”在第二个销的位置,符号“2”在第三个销的位置,符号“✉”在第四个销的位置。

[0059] 本发明最好是“在系统中”,它最好采用从设备操作系统中选定的一部分符号,因此它不要求用户上传任何照片或图像。用于创建验证码或加密信息的符号最好是分为两组或多组,这将被称为不同的“维度”。这将给用户在创建他们验证码时带来更多表达自己的方式。

[0060] 本发明通过把组成用户验证码的符号“隐藏”于不属于该用户验证码的其他符号中,来提供一种新的验证码防盗认证和加密机制。因此,在本质上是大海藏销。具体地讲,本发明利用在本文中将被称为“符章”的符号组。一个符章是一个其中至少有两个符号的小组。多个符章(一个“符章组”)会被呈现给用户,用户预先选定的销会部分或全部的出现在符章中。具体而言,用户验证码中的每个销(预选的符号)都可能被包括在呈现给用户的符章中。用户通过选择包括销在内的符章而输入验证码,使得每个选择的符章的销的位置对应于验证码中销的位置。因为每个所选的符章中不仅可能包括用户验证码中预选的一个销,也一定包括其它随机产生的没有在用户的验证码中被预选的其他符号,别人很难确定用户真正的验证码是什么。

[0061] 每一个用户在被要求提供验证码时,都会有一组随机产生的包括用户部分验证码的符章被呈现给用户。因此,用户每次被要求提供验证码时,都会输入一组不同的符章,从而防止旁观者通过用户输入的符章确定并盗取用户的验证码。

[0062] 作为一个说明性示例,可使用4组符号(4维),每个维度有36个符号。在登录过程中,36个符章会被呈现给用户,每个符章包括来自四个维度中每一维的一个符号。每个符号最好只显示在一个符章中(即,如果一个符号出现在一个符章中,它将不会再出现在另一个符章中)。因为此实例中显示36个符章,四个维度中的每个符号都被显示(36个符章中的每个符章都包括来自每一维的一个符号)。

[0063] 如果用户的验证码中销(预选的符号)的数量由变量“N”来表示,那么用户将需要选择N个符章(即包括各销构成的包括验证码的符章,每个所选符章的位置对应于它包括的验证码中的销的位置)。

[0064] 如上所述,本发明将因为用户对相应验证码中的每个销,都输入了一个包括4个符号的符章,而降低“偷看和拦截”用户验证码的可能性。因此即使一个用户的登录过程被黑客看见屏幕,或网络信息被黑客截取,黑客将不能确定每个符章的4个符号中的哪些符号是用户的验证码。因此,如果黑客试图登录到该用户的帐户,黑客将遇到另一组随机生成的符章,其中一些符章会包括构成该用户预选验证码的销,黑客就不知道应选择哪些符章。

[0065] 然而,如果黑客观察用户登录过程足够多的次数,黑客可以比较所有登录过程的记录,找出验证码中的每销是什么,因为每个销是一定会出现在符章中。如果黑客从不同登录记录中比较用户输入的所有第一符章,黑客将能最终确定第1销,而且如果黑客从登录记录中比较所有第二符章,黑客将能最终确定第2销,如此类推.

[0066] 因此为了防止黑客随着时间的推移找出验证码中的销,呈现给用户的随机生成的符章数量最好比每个维度中的符号数少。例如,如果每个所述一个或多个维度中有36个符号,可以只选择16个符章呈现给用户。这样做的结果是,不保证用户的每个销都一定会出现在符章中。在本实例中,用户试图登录时,验证码中的一个或多个销如果不在任何符章中,则该用户将选择任何一个符章作为对不存在的销的“通配符”符章(所选的通配符章的销的位置必须对应于不存在的销的位置)。这将使黑客的猜测更加困难,因为这有可能把一个随机选择的符章,放在没出现的销的位置上。

[0067] 使用比在一维或多维的符号组中数目更少的符章(例如,符号组中的一维或多维的数目是36时,使用16个符章)的另一个好处是会使用户更容易快速判断预选的销是否在符章中,而且屏幕给用户看起来更简单。

[0068] 本发明优选使用运用本发明的设备之操作系统中一部分的符号。因此不需要用户上载特殊的图形或照片到系统中,并因此减少了对用户与系统存储和维护那些信息的负载。

[0069] 本发明的系统和方法不仅可以用来验证用户,而且还可以认证多条信息,因此可以被用于信息加密。

[0070] 图1A描绘了本发明中一个可以由客户端系统访问的,一个设备或服务器的防盗认证/加密系统100框图的,一个优选示例。该系统100包括了认证/加密模块110,它提供防盗认证和/或加密的功能。系统100可有选择地连接到一个网络130。

[0071] 图1B描绘了本发明中一个连接到设备150的防盗认证/加密系统框图的一个优选

示例。该装置150最好包括一个用户界面模块120。认证/加密模块110提供了用户验证码的安全认证和/或信息加密,下面有更详细地说明。认证/加密模块110可有选择地连接到一个网络130。

[0072] 用户接口模块120可以由本领域中已知的,如一个图形用户界面,基于网络接口等的任何类型的用户界面来实现。在一般例子中用户接口模块120可以与,使用户能够与认证模块110进行交流的,任何类型接口连结来实现。

[0073] 图1C描绘了根据本发明的一个优选实施范例,由客户端设备170经由网络130连接的装在一个服务器160中的防盗认证/加密系统的框图。客户端设备170是任何类型的可以通过网络130访问服务器160的计算机或装置,并且优选地包括用户接口模块120使用户能与客户端设备170互动。认证/加密模块110提供了对用户验证码的安全验证和/或信息加密,以下将详细说明。

[0074] 通信链接140被用于认证/加密模块110与用户界面模块120,网络130和客户端设备170之间的通信。链接140可以是有线链接,无线链接,无线感应链接,电容链接或任何其他用于连接电子元件中公知的机制技术之间的通信。硬线链接可以适当地由总线实现,例如,工业标准体系结构(ISA)总线,微通道体系结构(MCA)总线,增强的ISA总线,视频电子标准协会(VESA)局部总线或外围组件互连(PCI)总线。

[0075] 无线链接的例子包括,但不限于,WAP(无线应用协议)链接,GPRS(通用分组无线电服务)链接,GSM(全球移动通信系统)链接,CDMA(码分多址)或TDMA(时分多址)链接,诸如移动电话的频道,一个GPS(全球定位系统)链接,CDPD(蜂窝数字分组数据),一个边(在运动的研究,有限公司)复式寻呼类型的装置,蓝牙无线链接,或基于IEEE 802.11的射频链接(WIFI)。

[0076] 该网络130可以是有线或无线网络,并且可以包括或对接任何一个或多个的,例如,互联网,内联网,PAN(个人区域网),LAN(局域网),WAN(广域网)或城域网(城域网),存储区域网络(SAN),幅中继连接,一个高级智能网络(AIN)连接,同步光网络(SONET)连接,数字T1,T3,E1或E3线路,数字数据服务(DDS)连接,DSL(数字用户线路)连接,以太网连接,ISDN(综合业务数字网)线,一个拨号端口,如V.90,V.34bis模拟调制解调器连接,电缆调制解调器,一个ATM(异步传输模式)连接,FDDI(光纤分布式数据接口)或CDDI(铜缆分布式数据接口)连接。

[0077] 用户接口模块120可以由本领域中已知的,如一个图形用户界面,基于网络的接口等的任何类型的用户界面来实现。在一般例子中,用户接口模块120可由,使用户能与认证/加密模块110进行交互的任何类型的接口来实现。

[0078] 如本文所使用的术语“模块”是指真实世界中的设备,组件,或由硬件实现的组件的配置,它可以包括例如一个专用集成电路(ASIC)或现场可编程门阵列(FPGA),或者一个微处理器系统和一组指令来实现该模块的功能,这(被执行的同时)使微处理器系统变成一个有特殊用途的用于实施该模块功能的设备。

[0079] 模块也可以被实施为单独的硬件及软件控制的硬件组合,某些功能由硬件单独提供,其它功能由硬件与软件的组合来提供。在某些实践中至少一部分,或者所有的模块都可在计算机或设备的,执行操作系统,系统程序和应用程序的(多个)处理器上执行(例如,服务器160和客户端设备170),同时,也可利用多任务,多线程,分布式(例如,云)处理,或其他

这样的技术来实施该模块。这样的计算机或装置的实例包括,但不限于,个人计算机(例如,台式计算机或笔记本计算机),服务器,自动取款机(ATM),销售终端,家用电器,以及移动计算设备,如智能电话,平板电脑或个人数字助理(PDA)。此外,服务器160是适当的任何类型的服务器,如Windows服务器,Linux服务器,Unix服务器诛如此类。

[0080] 图1D是一个示例性的,按照本发明的一个实例由硬件实现的,防盗认证/加密系统100的示意图。在图1D的实例中,认证/加密模块110由中央处理器118和存储器112实施。

[0081] 中央处理器118访问执行操作系统代码114,和存储在存储器112的其他程序代码116。实现认证/加密模块110的功能中的程序代码116被存储在存储器112或外部存储装置(未示出)上,由中央处理器118访问和执行。

[0082] 存储器112可通过以下来实现:例如,随机存取存储器(RAM),高速缓冲存储器,可移动的/不可移动的存储介质,易失性/非易失性存储介质,诸如不可移除的非易失性硬磁盘驱动器,可移除的非易失性的软盘驱动器,光盘驱动器(如CD-ROM,DVD-ROM,或任何其他光存储介质),USB闪存驱动器,和一个存储卡。

[0083] 现在描述认证/加密模块110的功能。认证/加密模块110,通过(由用户接口模块120)向用户显示多个有随机插入用户预选销的符章,对用户登录请求提供验证码/质询认证。如上所述,用户验证码中的每个销(由预选的符号表示)可能被包括在呈现给用户的符章中的一个。至少有一个用户的销会出现在符章中。用户选择包括销的符章构成验证码,使得每个被选择的符章中的销位置对应于验证码中销的位置。因为每个所选符章中包括随机生成的,没被用户作为验证码预选的一些符号,和被用户作为验证码预选的销中的一个符号,所以即使有人观察用户输入这些符章,他也不能确定用户的实际验证码是什么。

[0084] 下面描述两个认证/加密模块功能的说明性实例,并将在本文中称为图形访问表格输入法(Graphical Access Tabular Entry)_4(“GATE_4”)和图形访问表格输入法(Graphical Access Tabular Entry)_5(“GATE_5”)。如图2A所示,GATE_4实例使用4个维度的符号,每维包括36个符号。如图2B所示,GATE_5实例使用5维符号,每维包括26个符号。图2A和图2B是可用于该类型的符号的例子,并且应该理解的是,即使运用任何其他类型的符号,仍属于本发明的范围内。

[0085] 图2A和2B中显示是在现代计算机操作系统中一般已有的符号,并且不要求任何特殊的过程来创建或上传/保存到包括本发明的现有系统中。它们是大多数计算机操作系统中标准的Unicode系统的一个子集。每个符号都有一个Unicode代号,熟悉的字符:A,B,Z,0,1,9,@,+,<%都是Unicode系统中的一部分。例如:

[0086] Unicode的\u0062是字符:b

[0087] Unicode的\u2206是字符:Δ

[0088] Unicode的\u0040是字符:@

[0089] 在图2A和2B的实例中所示的符号。是因为它们广泛,独特,易记而被列入。

[0090] 图3显示了按照本发明的一个实例实现的用认证/加密模块110使用户能够创建验证码的示例性流程图。该过程开始于步骤310,其中用户被要求输入所需的用户名。在步骤320,认证/加密模块110确定用户名在存储器112中是否已存在。如果用户名存在,则过程继续到步骤330,否则该过程继续到步骤340。

[0091] 步骤330问用户是否要更新与用户名相关的现有验证码。如果用户指示“否”,过程

跳回到步骤310,如果用户指示“是”,处理过程进行到步骤340。

[0092] 在步骤340,可供用户选择其验证码的每个销的符号被显示给用户。然后,在步骤350,用户被要求选择一个显示的符号作为组成其验证码的一个销。在步骤360,该过程确定用户是否已请求了保存当前所选的(多个)销作为验证码。这可以通过,例如当用户准备好保存验证码时显示给用户一个可以选择的按钮,来实现。如果用户没有表示要保存验证码,则回到步骤350,用户选择作为验证码中另一个销的另一个符号。如果用户在步骤360指示该验证码应保存,则过程到步骤370。

[0093] 在步骤370中,确定所选择的验证码是否符合预定的长度要求(即,销的最少预定数目)。如果所选择的验证码符合规定,则该验证码在步骤380被保存,如果输入的验证码不符合,则返回到步骤350,其中用户被提示选择更多的作为销的符号。

[0094] 图4显示了用认证/加密模块110验证用户的,根据本发明实现的一个示例性流程图。这个过程开始于步骤410,用户被呈现的登录屏幕提示输入用户名。该过程然后进行到步骤420,其中,认证/加密模块110确认由用户输入的用户名是否存在。如果存在,则输入步骤430,如不存在,则回到步骤410,其中用户被提示输入另一个用户名。

[0095] 在步骤430,认证/加密模块产生基于用户验证码中销的数目的预定数量的符章。如实例图4中所述,产生16个符章并最好用一个4×4符章表显示给用户,如下将更详细描述。该符章组由根据图5中所示的符章生成规则生成。在实例的表中有16个符章,但在现实中,它可由任何数目的大于用户销数的符章组成,并可由3×4,2×5,或任何其他组合的符章表显示,4×4只是显示他们的首选方式。

[0096] 然后,该过程进行到步骤440,用户按照销出现在验证码中的顺序,如图6所示选择包含他验证码中销的符章。在步骤450,认证/加密模块110确认用户是否按照图6中符章选择规则所示,选择了符章。如果遵守了符章选择规则,就进行到步骤460,其中用户被认证并允许访问。如果没遵守符章选择规则,则进行到步骤470,其中用户不被认证并被拒绝访问。

[0097] 如上所述,图5是按照本发明的一个实例,被认证/加密模块110使用的,显示符章生成规则的例表。16个符章中至少出现一个用户验证码中的销。有时用户所有的销都会在符章中出现,大多数时间1到N(N是用户验证码的长度)用户预先选择的销将在16符章中出现。

[0098] 如图5所示,生成的是16个符章,而不是36(在GATE_4实例里),也不是26(在GATE_5实例里)。因此,在GATE_4实例中,仅16/36＝44％的时间里,用户验证码中的一个销可能会出现在16个符章中。在GATE_5实例中,仅16/26＝62％的时间里,用户验证码中的一个销会出现在16个符章中。有可能用户验证码里所有的销都会出现在符章中,而且可以保证至少有一个用户销将会出现在符章中。大部分时间,一些用户验证码中的销会缺席,另一些会出现在符章中。在其他实例中,可以改变规则,使至少2或3个用户验证码里的销会出现在符章表中。

[0099] 正是这种不确定性,使本发明有效。如果登录过程被偷看或拦截,黑客唯一能确定的事情是验证码的长度,因为,如果用户输入多于或少于验证码长度的符章数,登录将失败。唯一能导致,但并不保证,能成功登录的是输入与用户验证码中销的数量相同的符章。

[0100] 然而,即使黑客得知验证码是多长,黑客也不能确定每个销的符号。这是因为,即使销不出现在16符章中,用户仍可以成功登录。这是因为,如符章选择规则图6所示,用户可

以选择一个随机符章中的销代替所呈现的符章中不存在的销。

[0101] 此外,即使用户验证码里所有的销都出现在16符章中,黑客仍无法确定哪个符章里的哪个符号是预先选定的销,因为在GATE_4实例有4个符号,在GATE_5实例有5个符号。这种不确定性使得本发明的系统和方法能防盗与反拦截。

[0102] 如图6中符章选择规则所示。选择一个有效符章的规则可以概括如下:

[0103] • 用户必须从符章表中,对应于用户验证码中的N个销,选择N个符章。因此,如果用户验证码有4个销,用户就选择4个符章。同样地,如果用户验证码中有6个销,用户必须选择6个符章。

[0104] • 如果用户的一个销出现在16个符章的一个中,用户必须选择那个符章输入销码。

[0105] • 如果用户验证码中的一个销在16个符章中都不存在,用户必须选择16个符章中的任何一个(以下,称为“通配符章”)代替该销输入。

[0106] 如上所述,图7是列出由认证/加密模块110使用的,按照本发明的一个实例性的符章验证规则。这些规则是用于验证用户在登录时选择的符章。例如,该规则确定由用户输入的符章数是否等于用户验证码的长度。如果不是这样,用户登录将失败。如果是这样,则该规则要求检查用户验证码中的每个销,看它是否出现在16个符章中。如果用户验证码中的一个销不存在于所述符章中,用户必须选择一个随机符章。如果用户验证码中的一个销出现在符章中,用户必须选择该符章。

[0107] 图8A是按照本发明的一个实例,如屏幕截图3中的步骤310所示,用于输入用户名的一个空的登记画面。图中显示在用户输入用户名之前,GATE_4实施方案的登录画面是怎样的例子。

[0108] 图8B是根据本发明的一个实例,由系统100呈现来创建用户名注册过程的样本屏幕截图。它显示GATE_4在实施用户注册过程(创建用户名)中屏幕,如图3所示,可能是怎样的一个例图:

[0109] • 用户输入一个新的用户名:“admin”(G4 206),然后点击“检查可用性”按钮(G4 208)。系统检查,看用户名“admin”是否已在其存储器中(图3步骤320)。如果是这样,它会显示询问对话框(未显示):“用户名已经存在,你是否要更新现有的验证码?”如果用户不希望更新旧的验证码,该过程将关闭对话框,并等待用户输入另一个用户名。如果用户名“admin”不存在,或者如果它存在,但用户要更新现有的验证码,系统将启用G4 212,G4 222,G4 232和G4 242,它们各自在预定的维数中具有36个符号,如图2A所示。例如,G4 212包括所有第1维的36个符号,而且那些符号将显示在符章中“[1]”(左上)的位置,如图G4 210(符号是从1至36)。G4 222包括36个符号,从“Ⓐ”,到“?”,他们将显示在任何符章中“[2]”的位置(G4 220:右上)。G4 232具有从“○”到“℉”的36个符号,这将显示在符章中“[3]”(G4 230:左下)的位置,G4 242示出了从“+”至“₪”的36个符号,他们将会显示在符章中“[4]”(G4 240:右下)的位置。在这一过程中,该系统将启用在G4 212,G4 222,G4 232和G4 242上的按钮,使用户可以点击他们中的任何一个。作为比较,在图8A中,这些按钮没被启用并显得苍白,因为用户还未输入用户名。如果没有一个用户名,用户将不会被允许创建验证码。

[0110] • 用户点击在G4 222符号中的“Ⓘ”以选择第一销,它在G4 250(图3步骤350)中被显示。

[0111] • 用户点击在G4 232符号中的“♥”以选择第二销,它在G4 252(图3步骤350)中被显示。

[0112] • 用户点击在G4 212符号中的“2”以选择第三销,它在G4 254(图3步骤350)中被显示。

[0113] • 用户点击在G4 242符号中的“✉”以选择第四销,它在G4 256(图3步骤350)中被显示。

[0114] • 在这个例子中用户选择了由4个销组成的验证码,所以验证码长度为4。

[0115] • 在这个例子中G4 258和G4 260保持空缺。

[0116] • 然后用户通过单击“保存”按钮,完成用户名创建(注册)流程-G4 270(图3 370步)。该系统将验证码“Ⓘ♥ 2✉”与用户名“admin”保存到存储器中(图3步骤380)。

[0117] 如上讨论,多维选项并不限于实例图8B中所示的符号。多维选项可以包括任何其他符号。上述的示例中用户有4个销,但也可使用任何数量的销,而仍属本发明的范围之内。如果销的数量太少,验证码将太脆弱。如果销的数量过多,则用户可能记不住验证码。因此,优选的长度是在4到6个销之间。

[0118] 图9A是一幅该用户输入任何信息之前,由系统100呈现的根据本发明一个实例登录样本的屏幕截图。图9A是作为与图9B比较用的。

[0119] 图9B是由系统100执行的显示本发明登录过程的一幅样本屏幕截图。它显示了GATE_4在实施过程中,用户登录时,屏幕如图4所示,可能出现的如下一个例子:

[0120] • 用户输入用户名:“admin”(G4 306)(图4步骤410)。

[0121] • 用户点击“输入”键(G4 309)。系统检查,看用户名“admin”是否已经在其存储器中(图4步骤420),如果不存在,它会显示一条信息(未显示),“用户名不存在,请输入一个有效的用户名“。如果它存在,则系统会显示一个4×4表格(G4 320)。为了更好地描述该表,各行由上向下按:A,B,C,D的次序标志,各列从左至右按:1,2,3,4的次序标志,此表中的符章是根据图5所述的规则生成。

[0122] • 因为我们从图8B中知道,与用户名“admin”相应的验证码是:“Ⓘ ♥ 2✉”,用户需要从16个符章中找到第一销“Ⓘ”开始。

[0123] 因为Ⓘ符号属于第二维,在这个例子中,出现在任何符章的右上方,所以用户只需扫视每个符章的右上角,以查看Ⓘ是否存在。在这个例子中,它在该符章表中的D2位置。这个屏幕截图摄于演示模式,在演示模式中本程序为让用户更好理解这一过程,标示符章中匹配的符号。实时,这并不需要标示。在该图中,在D2中的Ⓘ被标示出。因为它是在D2符章中出现,用户必须如图6所述的规则点击输入这个符章。

[0124] • 用户点击D2之后,符章被复制到验证码的第一位置(G4 350)(图4步骤440)。

[0125] • 用户验证码的第二个销是:“♥”,而这种符号属于第三维。在这个例子中,第三维符号出现在任何符章的左下侧。因此,用户只需察看这16个符章中每个符章的左下侧。在

此例中,它是在符章D3里。因此,用户应该点击输入D3(图4步骤440)。它在本截图中被标示出。

[0126] • 用户点击D3(图4步骤440)之后,该符章被复制到验证码(G4 353)的第二位置。

[0127] • 验证码的第三销是“2”,而且它属于第1维。因此,在该示例中,用户只需察看这16个符章中每个符章的左上侧。在这个例子中,它不存在。根据符章选择规则(图6),用户可以并必须选择用于该销位置的通配符符章(任何符章)。在本实例中,用户随机地点击了符章C3。该符章将被复制到第3销位置-G4 356。

[0128] • 第四个也是最后一个销为“✉”,而这种符号属于第四维。在本实例中第4维符号出现在任何符章的右下侧。因此,用户只需察看这16个符章中每个符章的右下侧。在这个例子中,它是在符章B3中,所以,用户需要点击B3(图4步骤440)。在本图中,它被标示出。用户点击B3后,该符章被复制到验证码(G4 359)的第四位。

[0129] • 由于验证码中只有4个销,G4 362和G4365空缺,他们应该保持空白。如果用户在这一个或两个位置输入符章,系统将拒绝用户访问,因为用户输入了多于原来4个销的符章(图4步骤450)。

[0130] • 输入所有4个符章之后,用户会点击“登录”(G4 370)按钮让系统知道用户完成了符章选择过程,系统将检查所输入的符号是否附合图7所述的规则(图4步骤450)。

[0131] • 在这个例子中,用户输入的符章是有效的,而系统将显示一个“登录成功”信息(G4 380)并允许用户访问(图4步骤460)。

[0132] 图9C是由系统100执行的显示本发明登录过程中失败的一幅样本屏幕截图。它显示了GATE_4在实施过程中,用户登录时,屏幕如图4所示,可能出现的如下一个失败的例子:

[0133] • 用户输入用户名:“admin”(G4 307)(图4步骤410)。

[0134] • 用户点击“输入”键(G4 310)。系统检查,看用户名“admin”是否已经在其存储器中(图4步骤420),如果不存在,它会显示一条信息(未显示),“用户名不存在,请输入一个有效的用户名“。如果它存在,系统会显示一个4×4表格(G4 321)。为了更好地描述该表,各行由上向下按:A,B,C,D的次序标志,各列从左至右按:1,2,3,4的次序标志,此表中的符章是根据图5所述的规则生成。

[0135] • 因为我们从图8B中知道。与用户名“admin”相应的验证码是:“Ⓘ♥ 2✉”,用户需要从16个符章中找到第一销“Ⓘ”开始。因为符号Ⓘ属于第二维,在这个例子中,出现在任何符章的右上侧,所以用户只需扫视每个符章的右上部分,以查Ⓘ是否存在。在这个例子中,它在该符章A1中。用户必须根据图6所描述的规则点击这个符章。

[0136] • 用户点击A1后,该符章被复制到验证码的第一位置(G4 351)(图4步骤440)。

[0137] • 用户验证码的第二销是:“♥”,而这种符号属于第三维。在这个例子中,第三维符号出现在任何符章的左下侧。因此,用户只需察看这16个符章中每个符章的左下侧。在这个例子中,它是在符章D1中。因此,用户应该点击D1(图4步骤440)。然而,在这个例子中,用户没有点击这个符章,相反,点击了B4。这是错误的符章,因此系统会记录这错误并拒绝用户访问。

[0138] • 用户点击B4(图4步骤440)后,该符章被复制到验证码(G4 354)的第二位置。

[0139] • 验证码的第三销是“2”,而且它属于第1维。因此,在该示例中,用户只需察看这16个符章中每个符章的左上侧。在这个例子中,它不存在。根据符章选择规则(图6)中,用户可以并必须选择用于该销位置的通配符符章(任何符章)。在本实例中,用户随机地点击了符章C4。该符章将被复制到第3销位置-G4 357。

[0140] • 第四个也是最后一个销为“✉”,而这种符号属于第四维。在本实例中第4维符号出现在任何符章的右下侧。因此,用户只需察看这16个符章中每个符章的右下侧。在这个例子中,它在符章A2中,所以用户需要点击A2(图4步骤440)。在用户点击A2后,该符章将被复制到验证码(G4 360)的第四位。

[0141] • 由于验证码只有4个销,G4 363和366G4空缺,他们应该保持空白。如果用户在一个或两个位置输入符章,系统将拒绝用户访问,因为用户输入了多于原来4个销的符章(图4步骤450)。

[0142] • 输入所有4个符章之后,用户会点击“登录”(G4 371)按钮让系统知道用户完成了符章选择过程,系统将检查所输入的符号是否附合图7所述的规则(图4步骤450)。

[0143] • 在这个例子中,用户输入的符号是无效的,系统将显示一个“登录失败”的信息(G4 381),并拒绝用户访问(图4步骤470)。

[0144] 图9D是由系统100执行的显示本发明登录过程中另一幅失败的样本屏幕截图。它显示了GATE_4在实施过程中,用户登录时,屏幕如图4所示,可能出现的如下另一个失败的例子:

[0145] • 用户输入用户名:“admin”(G4 308)(图4步骤410)。

[0146] • 用户点击“输入”键(G4 311)。系统检查,看用户名“admin”是否已经在其存储器中(图4步骤420),如果不存在,它会显示一条信息(未显示),“用户名不存在,请输入一个有效的用户名“。如果它存在,则系统会显示一个4×4表格(G4 322)。为了更好地描述该表,各行由上向下按:A,B,C,D的次序标志,各列从左至右按:1,2,3,4的次序标志,此表中的符章是根据图5所述的规则生成的。

[0147] • 因为我们从图8B中知道。与用户名“admin”相应的验证码是:“Ⓘ♥ 2✉”,用户需要从16个符章中找到第一销“Ⓘ”开始。因为符号Ⓘ属于第二维,在这个例子中,出现在任何符章的右上侧,所以用户只需扫视每个符章的右上部分,以查看Ⓘ是否存在。在这个例子中,它在该符章B2中。用户必须根据图6所描述的规则点击这个符章。

[0148] • 用户点击B2后,该符章被复制到验证码的第一位置(G4 352)(图4步骤440)。

[0149] • 用户验证码的第二个销是:“♥”,而这种符号属于第三维。在这个例子中,第三维符号出现在任何符章的左下侧。因此,用户只需察看这16个符章中每个符章的左下侧。在这个例子中,它不存在。根据符章选择规则(图6)中,用户可以并必须选择用于该销位置的通配符符章(任何符章)。在本实例中,用户随机地点击了符章A4。该符章将被复制到第二销位置-G4 355。

[0150] • 验证码的第三销是“2”,并且它属于第1维。因此,在该示例中,用户只需察看这16个符章中每个符章的左上位置。在这个例子中,它在符章B3中。在这个例子中,用户点击了B3,该符章被复制到第3销位置-G4358。

[0151] • 第四个也是最后一个销为“✉”,而这种符号属于第四维。在本实例中第4维符号出现在任何符章的右下侧。因此,用户只需察看这16个符章中每个符章的右下侧。在这个例子中,它是在符章D1中,所以,用户需要点击D1(图4步骤440)。用户点击D1后,该符章将被复制到验证码的第四位置(G4 361)。

[0152] • 由于验证码只有4个销,G4 364和367G4应空缺。在本实例中,用户输入了一个额外的符章D2,因此,系统将拒绝用户访问,因为用户输入了比原来4个销更多的符章。

[0153] • 用户输入所有5个符章后,用户点击“登录”(G4 372)按钮让系统知道用户完成了符章选择过程,系统将检查所输入的符号是否附合图7所述的规则(图4步骤450)。

[0154] • 在这个例子中,用户输入了过多的符章,系统显示“登录失败”的信息(G4 382),并拒绝用户的访问(图4步骤470)。

[0155] 图10A以文字型式显示了GATE_4用户登录过程实例的样本屏幕截图。它反映了程序启动时一个空的屏幕。这个图像被用来和图10B,10C及下面的10D作比较。该屏幕是为了对显示幕后发生的事情作出解释。它在实际用户登录时不显示。

[0156] 图10B以文字型式显示了GATE_4用户登录过程实例的样本屏幕截图。它显示了用户登录过程图9B背后发生了什么。并显示了图4一个范例的处理流程。这只是一个演示,在实际用户登录时不显示。它是用来在视觉上形像说明图7中所示的符章验证规则。

[0157] 图中有3列:“客户端”(左侧),“网络连接”(中)和“服务器端”(右侧)。这个过程从用户在客户端输入他的用户名开始,然后信息通过网络连接到达服务器端。服务器生成16个符章,并将其传送到网络上,然后网络将符章传送到客户端。

[0158] 用户按照图6中所示的符章选择规则选择符章。所选的符章被传送到网络,然后传送到服务器端进行验证,允许或拒绝访问的结果通过网络传送给客户端。处理流程由图10B中箭头表示。下面详细解释。

[0159] • 用户输入用户名:“admin”(G4 406)。

[0160] • 用户点击“输入”键(G4 409)。

[0161] • 用户名“admin”(G4 406)显示在客户端(G4 411),并通过(G4421)连接到网络(G4 413),然后将其再传送给(G4 422)服务器端(G4415)。

[0162] • 在服务器端,系统会检查它的内存,看用户名“admin”是否存在,如果没有,系统会显示一条信息:“用户名不存在,请输入有效的用户名”(未显示)。此例与图8B中相同,在存储器中的验证码是:“Ⓘ♥ 2✉”,系统在存储器中发现它(G4 417)。

[0163] • 系统按照图5中所示的符章生成规则生成16个符章(G4 423)。

[0164] • 那16个符章被传送(G4 424)到网络。

[0165] • 网络传送(G4 425)符章到客户端。

[0166] • 那16个符章被显示在用户登录屏幕上,如图9B所示,的4×4表格中(图9B G4 320)。

[0167] • 用户选择(G4 426)4个符章:G4 350,G4 353,G4 356和G4 359。

[0168] • 用户点击“登录”(图9B G4 370)后4个用户选择的符章被传送(G4427)到网络。

[0169] • 用户选择的4个符章然后被传送(G4 428)到服务器侧。

[0170] • 在服务器端,系统逐个检查所有4个符章:C11,C12,C13和C14,在该例子中,它们

是正确的。

[0171] • 上述登录成功(G4 429)的结果被传送(G4 430)到网络。

[0172] • 网络把结果传送(G4 431)给客户端,并显示(G4 432)图9B中G4380所示的信息。

[0173] 图10C以文字型式显示了GATE_4实施方案中,用户登录过程的样本屏幕截图。它显示了用户登录过程图9C背后发生了什么。并显示了图4一个范例的处理流程。这只是一个演示,在实际用户登录时不显示。它是用来在视觉上形像说明图7中所示的符章验证规则。

[0174] 图中有3列:“客户端”(左侧),“网络连接”(中)和“服务器端”(右侧)。这个过程从用户在客户端输入他的用户名开始,然后信息通过网络连接到服务器端。服务器生成16个符章,并将其传送到网络上,然后将符章传送到客户端。

[0175] 用户按照图6中所示的符章选择规则选择符章。所选择的符章被传送到网络,然后传送到服务器端进行验证,允许或拒绝访问的结果通过网络传送给客户端。处理流程由图10C中箭头表示。下面详细解释。

[0176] • 用户输入用户名:“admin”(G4 506)。

[0177] • 用户点击“输入”键(G4 509)。

[0178] • 用户名“admin”(G4 506)显示在客户端(G4 511),并通过(G4 521)连接送到网络(G4 513),然后将其再传送给(G4 522)服务器端(G4 515)。

[0179] • 在服务器端,系统会检查它的内存,看用户名“admin”是否存在,如果没有,系统会显示一条信息:“用户名不存在,请输入有效的用户名”(未显示)。此例与图8B中相同,在存储器中的验证码是:“Ⓘ♥ 2✉”,系统在存储器中发现它(G4 517)。

[0180] • 系统按照图5中所示的符章生成规则生成16个符章(G4 523)。

[0181] • 那16个符章被传送(G4 524)到网络。

[0182] • 网络传送(G4 525)符章到客户端。

[0183] • 那16个符章被显示在用户登录屏幕上,如图9C所示,的4×4表格中(G4 321在图9C)。

[0184] • 用户选择(G4 526)4个符章:G4 351,G4 354,G4 357和G4 360。

[0185] • 用户点击“登录”(图9C G4 371)后4个用户选择的符章被传送(G4 527)到网络。

[0186] • 4个用户选择的符章然后被传送(G4 528)到服务器端。

[0187] • 在服务器端,系统逐个检查所有4个符章:C21,C22,C23和C24,本实例的第二个符号是不正确的(因为第二销“♥”在D1符章中存在,但用户选择了B4符章,这是错误的。因此,结果是失败的登录)。

[0188] • 上述登录失败(G4 529)的结果被传送(G4 530)到网络。

[0189] • 网络传送(G4 531)结果给客户端,并显示(G4 532)图9C G4 381所示的信息。

[0190] 图10D以文字型式显示了GATE_4实施方案中,用户登录过程的样本屏幕截图。它显示了用户登录过程图9D背后发生了什么。并显示了图4一个范例的处理流程。这只是一个演示,在实际用户登录时不显示。它是用来在视觉上形像说明图7中所示的符章验证规则。

[0191] 图中有3列:“客户端”(左侧),“网络连接”(中)和“服务器端”(右侧)。这个过程从用户在客户端输入他的用户名开始,然后信息通过网络连接到服务器端。服务器生成16个符章,并将其传送到网络上,然后将符章传送到客户端。

[0192] 用户按照图6中所示的符章选择规则选择符章。所选择的符章被传送到网络,然后传送到服务器端进行验证,允许或拒绝访问的结果通过网络传送给客户端。处理流程由图10D中箭头表示。下面详细解释。

[0193] • 用户输入用户名:“admin”(G4 606)。

[0194] • 用户点击“输入”键(G4 609)。

[0195] • 用户名“admin”(G4 606)显示在客户端(G4 611),并通过(G4 621)连接到网络(G4 613),然后将其再传送给(G4 622)服务器端(G4 615)。

[0196] • 在服务器端,系统会检查它的内存,看用户名“admin”是否存在,如果没有,系统会显示一条信息:“用户名不存在,请输入有效的用户名”(未显示)。此例与图8B中相同,在存储器中的验证码是:“Ⓘ♥ 2✉”,系统在存储器中发现它(G4 617)。

[0197] • 系统按照图5中所示的符章生成规则生成16个符章(G4 623)。

[0198] • 那16个符章被传送(G4 624)到网络。

[0199] • 网络传送(G4 625)符章到客户端。

[0200] • 那16个符章在用户登录屏幕上,如图9D所示,被显示在4×4表格中(G4 322在图9D)。

[0201] • 用户选择(G4 626)5个符章:G4 352,G4 355,G4 358,G4 361和G4 364。

[0202] • 用户点击“登录”(图9D G4 372)后5个用户选择符章被传送(G4 627)到网络。

[0203] • 用户选择的5个符章然后被传送(G4 628)到服务器侧。

[0204] • 在服务器端,系统逐个检查所有5个符章:C31,C32,C33,C34和C35,本实例中的第5个符章是不正确的(因为验证码只有4个销,但用户输入了第五符章,这是错误的。因此,结果是失败的登录)。

[0205] • 上面失败的登录(G4 629)结果被传送(G4 630)到网络。

[0206] • 网络传送(G4 631)结果给客户端,并显示(G4 632)图9D G4 382所示的信息。

[0207] 图11A是用户名创建(注册)过程中GATE_5实例的样本屏幕截图。它反映了程序启动时一个空的屏幕。这个图像被作为和以下的图11B比较的基础。

[0208] 图11B是用户名创建(注册)过程中GATE_5实例的样本屏幕截图。它显示了每一符章中每一维符号的位置。它还反映了用户如何创建一个新的用户名,以及用户如何选择并保存销以形成与用户名相应的验证码。过程如下:

[0209] • 用户输入一个新的用户名:“admin”(G5 206),然后点击“查看”按钮(G5 208)。系统查看用户名“admin”是否已经在它的内存中,如果是的话,它会显示询问对话框(未显示):“用户名已经存在,你是否要更新现有的验证码?”如果用户不希望更新旧的验证码,该过程将关闭对话框,并等待用户输入其他用户名。如果用户名“admin”不存在,或者如果它存在,但用户要更新现有的验证码,系统将启用G5 212,G5 222,G5 232,G5 242和G5 248的按钮,每一维都有预定的26个符号,如图2B所述。

[0210] 例如,G5 212包括所有第1维的26个符号,这些符号将显示在符章中“[1]”(左上)的位置,如图G5 210。这26个符号是从“Ⓐ”到“ⓩ“。G5 222显示了从“α”到“ϟ”的26个符号,它们来自第二维,他们会显示在任何符章“[2]”(G5 220:右上)的位置。G5 232示出了26个数字,从1至26,它们来自第3维,将在符章中显示在“[3]”(G5 230:在中间)的位置,G5

242显示了从“○”至“≠”的26个符号,它们来自第四维度,他们会显示在任何符章“[4]”(G5 240:左下)的位置。G5 248显示了从“+”到“₪”的26个符号,他们来自第五维,他们将显示在任何符章“[5]”(G5 246:右下)的位置。

[0211] 此时,该系统将允许启用上述的G5 212,G5 222,G5 232,G5 242和G5 248按钮,使用户可以点击任何一个。作为比较,在图11A中这些按钮没被启用并显得苍白,因为用户还没有输入任何用户名。如果没有一个用户名,它不会允许用户选择任何销。

[0212] • 用户在G5 248的符号中选择并点击第一销“$”,这在G5 250中显示出来。

[0213] • 用户在G5 242的符号中选择并点击第二销“=”,它在G5 252中显示出来。

[0214] • 用户在G5 212的符号中选择并点击第三销“Ⓜ”,它在G5 254中显示出来。

[0215] • 用户在G5 212的符号中选择并点击第四销“Ⓒ”,它在G5 256中显示出来。

[0216] • 用户在G5 232的符号中选择并点击第五销“2”,它在G5 258中显示出来。

[0217] • 用户在G5 242的符号中选择并点击第六销“☺”,它在G5 260中显示出来。

[0218] • 在这个例子中用户在他的验证码里选择了6个销,所以他的验证码的的长度为6。

[0219] • 然后用户单击“保存”按钮(G5 270)完成用户名的创建(注册)过程。该系统将把验证码“$=ⓂⒸ2☺”与用户名“admin”保存到内存中。

[0220] 图12A是用户登录过程GATE_5实例的样本屏幕截图。它反映了程序启动时一个空的屏幕。这个图像被作为和以下的图12B,12C及12D比较的基础。

[0221] 图12B是GATE_5实例的用户登录过程样本屏幕截图。它显示了用户选择验证码的,用16个符章的4×4表格。它反映了符章选择过程是如何工作的,并在用户选择的符章中标出了有户销的符号。它还显示了成功登录可能是什么样的。本实例过程遵循图11B的例子。所以使用同样的验证码。

[0222] 处理过程如下:

[0223] • 用户输入一个新的用户名:“admin”(G5 306)。

[0224] • 用户点击“输入”键(G5 309)。系统检查,看用户名“admin”是否已经在它的内存,如果不存在的话,它会显示出“用户名不存在,请输入有效的用户名”的信息(未显示)。如果它存在,则系统会显示一个4×4的表格(G5 320),以为了更好地描述该表,各行由上向下按:A,B,C,D的次序标志,各列从左至右按:1,2,3,4的次序标志。此表中的符号是根据图5所描述的规则生成的。

[0225] • 由于我们从图11B知道,与用户名“admin”相应的验证码是:“$= ⓂⒸ2☺”,因此用户需要通过从有16个符章的表格中找到第一销“$”开始。

[0226] 因为符号$属于第五维,它只会出现在任何符章的右下角,所以用户只需扫视每个符章的右下部分,以查看$是否存在。在此例中,它在符章B4中,画面拍摄于演示模式,演示模式程序用方框标示出匹配的符号供用户更好地理解这个过程。实用时,它不会被标出。在此实例中,在B4中的$被标出,因为它在B4符章中出现,用户必须根据图6所描述的规则点击这个符章。

[0227] • 用户点击B4后,该符章被复制到验证码(G5 350)的第一个位置。

[0228] • 用户验证码的第二销是:“=”,并且这一符号属第4维,第4维符号只出现在任何符章的左下侧,所以用户仅需看16个符章中每个符章的左下侧,在此例中,它在符章D2中,所以用户应点击D2。它在屏幕截图中被用方框标出。

[0229] • 用户点击D2后,该符章被复制到验证码的第二位置(G5 353)。

[0230] • 验证码的第三销是:“Ⓜ”,它属于第一维,所以用户只需察看这16个符章中每个符章的左上角位置,在此例中,它在D4符章中,所以用户应该点击D4。它在屏幕截图中被用方框标出。

[0231] • 用户点击D4后,该符章被复制到验证码的第三个位置(G5 356)。

[0232] • 第四销是:“Ⓒ”,这符号属于第1维,1维符号只出现在任何符章的左上侧,所以用户仅需察看这16个符章中每个符章的左上侧,在此例中,它不存在,根据符章选择规则,用户可以而且必须在销的位置选择通配符符章(任何符章),所以用户点击一个随机符章A4,该符章被复制到第4销位G5 359。

[0233] • 验证码第五销是:“2”,它属于第三维,所以用户只需察看这16个符章中每个符章的中心,在此例中,它在D2符章中,所以用户应该点击D2。它在屏幕截图中被用方框标出。用户点击后,该符章将被复制到验证码的第5位置(G5 362)。

[0234] • 第六即最后销是:“☺”,这一符号属第4维,第4维符号只出现在任何符章的左下侧,所以用户仅需察看16个符章中每个符章的左侧,在此例中,它在符章A4中,所以用户需要点击A4。它也在演示程序中被用方框标出。用户点击后,该符章将被复制到验证码的第6即最后一个位置(G5365)。

[0235] • 用户输入所有的6个符章后,他将点击“登录”(G5 370)按钮让系统知道他已经完成了符章选择过程,并且系统会检查输入的符章是否按照图7规定所描述是有效的。

[0236] • 在这个例子中,用户输入的符章是有效的,系统显示“登录成功”信息并允许用户访问(G5 380)。

[0237] • 在这个例子中,符章G5 353和G5 362是一样的,符章G5 359和365G5也是一样的,这只是巧合,这样的情况很可能会经常发生。这很可能会迷惑试图猜测验证码的人。

[0238] 图12C是GATE_5用户登录过程的实例样本屏幕截图。它显示了用户从16个符章中选择验证码的4×4表格。它反映了符章选择过程是如何工作的。这也显示了有3销错误的失败登录可能是什么样子的。本实例过程遵循图11B的例子,所以同样的验证码将被使用。处理过程如下:

[0239] • 用户输入用户名:“admin”(G5 307)。

[0240] • 用户点击“输入”键(G5 310)。系统检查,看用户名“admin”是否已经在它的内存,如果不存在的话,它会显示出“用户名不存在,请输入有效的用户名”的信息(未显示)。如果它存在,系统会显示一个4×4表格(G5 321),为更好地描述表格,各行由上向下按:A,B,C,D的次序标志,各列从左至右按:1,2,3,4的次序标志,此表中的符章是根据图5所述的规则生成。

[0241] • 由于我们从图11B知道,与用户名“admin”相应的验证码是:“$= ⓂⒸ2☺”,因此用户需要从16个符章的表格中找到第一销:“$”开始。

[0242] 因为符号$属于第五维,它只会出现在任何符章的右下位置,所以用户只需扫视每

个符章的右下部分,以查看$是否存在。在此例中,它是在符章A2中,用户必须根据图6所描述的规则点击这个符章。

[0243] • 用户点击A2后,该符章将被复制到验证码的第一个位置(G5 351)。

[0244] • 用户验证码的第二销是:“=”,这符号属于第4维,第4维符号只出现在任何符章的左下侧,所以用户仅需察看16个符章中每个符章的左下侧,在此例中,它不存在,根据符章选择规则,用户可以而且必须在销的位置选择通配符符章(任何符章),所以用户点击了一个随机符章A3。

[0245] • 用户点击A3后,该符章将被复制到验证码的第二位置(G5 354)。

[0246] • 验证码的第三销是:“Ⓜ”,它属于第一维,所以用户只需察看这16个符章中每个符章的左上角位置,在此例中,它是在符章C1中,用户需要点击C1。在此例中用户点击了C1。

[0247] • 用户点击C1后,该符章将被复制到验证码的第三个位置(G5 357)。

[0248] • 第四销是:“©”,而这种符号属于第一维,用户只需察看这16个符章中每个符章的左上角位置,在此例中,它是在符章D2中,用户需要点击D2。在此例中用户没点击D2,而点击了C2。这是错误的,系统将拒绝用户访问。

[0249] • 用户点击C2后,该符章被复制到验证码的第四位(G5 360)。

[0250] • 验证码第五销为“2”,它属于第三维,所以用户只需察看这16个符章中每个符章的的中心,在此例中,它是在符章C3中,而根据图6中的选择规则用户必须选择该符章,但在此例中用户选择了符章B2,这是错误的,系统会检查并注意到。

[0251] • 用户点击了错误的符章B2后,它被复制到验证码的第5位置(G5363)。

[0252] • 第六即最后销是:“☺”,并且这符号属第4维,第4维符号只出现在任何符章的左下侧,所以用户仅需察看16个符章中每个符章的左下侧,在此例中,它在符章D1中,用户需点击D1。在此例中用户没有点击D1,而点击了符章C4,这是错误的,系统会注意到这一点。

[0253] • 用户点击了错误的符章C4后,它被复制到验证码的第6位置(G5366)。

[0254] • 用户输入所有的6个符章后,点击“登录”(G5 371)按钮让系统知道他已经完成了符章选择过程,并且系统会检查输入的符章是否按照图7规定所描述是有效的。

[0255] • 在这个例子中,用户输入的符章无效,并且系统显示“登录失败”的信息,并拒绝用户访问(G5 381)。

[0256] 图12D是GATE_5用户登录过程实例的样本屏幕截图。它显示了用户从16个符章中选择验证码的4×4表格。它反映了符章选择过程是如何工作的。这也显示了缺少一个销的失败登录可能是什么样子的。本实例过程遵循图11B的例子,所以同样的验证码将被使用。处理过程如下:

[0257] • 用户输入用户名:“admin”(G5 308)。

[0258] • 用户点击“输入”键(G5 311)。系统检查,看用户名“admin”是否已经在它的内存,如果不存在的话,它会显示出“用户名不存在,请输入有效的用户名”的信息(未显示)。如果它存在,则系统会显示一个4×4的表格(G5 322),为了更好地描述该表,各行由上向下按:A,B,C,D的次序标志,各列从左至右按:1,2,3,4的次序标志,此表中的符章是根据图5所述的规则生成的。

[0259] • 由于我们从图11B知道,与用户名“admin”相应的验证码是:“$= ⓂⒸ2☺”,因此用户需要从16个符章的表格中找到第一销:“$”开始。

[0260] 由于符号$属于第五维,它只会出现在任何符章的右下位置,用户只需扫视每个符章的右下部分,看$是否存在。在此例中,它不存在,用户可以而且必须在此销的位置选择通配符符章(任何符章),所以用户点击一个随机符章D3。

[0261] • 用户点击D3后,该符章将被复制到验证码的第一个位置(G5 352)。

[0262] • 用户验证码的第二销是:“=”,这符号属于第四维,第4维符号只出现在任何符章的左下侧,所以用户仅需察看16个符章中每个符章的左下侧,在此例中,它在符章C1中,用户必须点击这个符章。

[0263] • 用户点击C1后,该符章将被复制到验证码的第二位置(G5 355)。

[0264] • 验证码的第三销是:“Ⓜ”,它属于第1维,所以用户只需察看这16个符章中每个符章的的左上位置,在此例中,它在符章A1中,所以用户需要点击A1。在此例中用户点击了A1。

[0265] • 用户点击A1后,该符章被复制到验证码的第三个位置(G5 358)。

[0266] • 第四销是:“Ⓒ”,这符号属于第1维,用户只需察看这16个符章中每个符章的左上位置,在此例中,它在符章D3中,所以用户需要点击D3。在此例中用户点击了D3。

[0267] • 用户点击D3后,该符章将被复制到验证码的第四位置(G5 361)。

[0268] • 验证码的第五销为“2”,它属于第三维,用户只需察看这16个符章中每个符章的中心,在此例中,它在符章C4中,而根据图6中的选择规则用户必须选择该符章,在示例中用户选择了符章C4。

[0269] • 用户点击符章C4后,它被复制到验证码的第5位置(G5 364)。

[0270] • 第六即最后销是:“☺”,并且这一符号属第4维,第4维符号只出现在任何符章的左下侧,所以用户仅需察看16个符章中每个符章的左下侧,在此例中,它在符章C2中,所以用户需要点击C2。但是,在此例中用户没有点击C2,而是留下最后一个位置空白,只输入了5个销。这是错误的。

[0271] • 用户输入上述5个符章后,点击“登录”(G5 372)按钮让系统知道他已经完成了符章选择过程,并且系统会检查输入的符章是否按照图7规定所描述是有效的。

[0272] • 在此例中,用户输入的符章是无效的,因为原来的验证码有6个销,但在此例中,用户只输入了5个符章,所以用户访问请求被拒绝,该系统显示一个“登录失败”的信息(G5 382)。

[0273] 图13A以文字型式显示了GATE_5实例的用户登录过程的样本屏幕截图。它反映了程序启动时一个空的屏幕。这个图像被用作和图13B,13C和下面的13D进行比较的基础。该屏幕是为了对显示幕后发生的事情作出解释。它在实际用户登录时不显示。

[0274] 图13B以文字型式显示了GATE_5实例中用户登录过程的样本屏幕截图。它显示了用户登录过程图12B背后发生了什么。并且显示了处理流程图4的一个示范实例。这只是一个演示,在实际用户登录时不显示。它是用来在视觉上形像说明图7中所示的符章验证规则。

[0275] 图中有3列:“客户端”(左侧),“网络连接”(中)和“服务器端”(右侧)。这个过程从

用户在客户端输入他的用户名开始,然后信息通过网络连接到达服务器端。服务器生成16个符章,并将其传送到网络上,然后网络将符章传送到客户端。

[0276] 用户按照图6中所示的符章选择规则选择符章。所选的符章被传送到网络,然后传送到服务器端进行验证,允许或拒绝访问的结果通过网络传送给客户端。处理流程由图13B中箭头表示。下面详细解释:

[0277] • 用户输入用户名“admin”(G5 406)。

[0278] • 用户点击“输入”(G5 409)。

[0279] • 用户名“admin”(G5 406)显示在客户端(G5 411),并通过(G5 421)连接到达网络(G5 413),然后将其再传送给(G5 422)服务器端(G5 415)。

[0280] • 在服务器端,系统会检查它的内存,看用户名“admin”是否存在,如果没有,系统会显示一条信息:“用户名不存在,请输入有效的用户名”(未示出)。此例与图11B中相同,所以在存储器中的验证码是:“$=Ⓜ ©2☺”,系统在存储器中发现了它(G5 417)。

[0281] • 该系统会根据图5生成16个符章(G5 423)。

[0282] • 那16个符章被传送到网络(G5 424)。

[0283] • 网络传送符章到客户端(G5 425)。

[0284] • 那16个符章在用户登录屏上,如图12B所示,被显示在4×4表格中(G5 320)。

[0285] • 用户选择6个符章(G5 426):G5 350,G5 353,G5 356,G5 359,G5 362和G5 365。

[0286] • 用户点击“登录”(G5 370)后,如图12B所示,6个用户选择的符章被传送到网络(G5 427)。

[0287] • 那6个用户选择的符章然后被传送到服务器侧(G5 428)。

[0288] • 在服务器端,系统会逐一检查所有6个符章:K11,K12,K13,K14,K15和K16,在此例中,它们是正确的。

[0289] • 登录成功(G5 429)的上述结果被传送到网络(G5 430)。

[0290] • 网络传送(G5 431)结果给客户端,并显示(G5 432)图12B的G5 380中所示的信息。。

[0291] 图13C以文字型式显示了GATE_5实施方案中,用户登录过程的样本屏幕截图。它显示了用户登录过程图12C背后发生了什么。并显示了图4一个范例的处理流程。这只是一个演示,在实际用户登录时不显示。它是用来在视觉上形像说明图7中所示的符章验证规则。

[0292] 图中有3列:“客户端”(左侧),“网络连接”(中)和“服务器端”(右侧)。这个过程从用户在客户端输入他的用户名开始,然后信息通过网络连接到达服务器端。服务器生成16个符章,并将其传送到网络上,然后网络将符章传送到客户端。

[0293] 用户按照图6中所示的符章选择规则选择符章。所选择的符章被传送到网络,然后传送到服务器端进行验证,允许或拒绝访问的结果通过网络传送给客户端。处理流程由图13C中箭头表示。下面详细解释。

[0294] • 用户输入用户名:“admin”(G5 506)。

[0295] • 用户点击“输入”键(G5 509)。

[0296] • 用户名“admin”(G5 506)显示在客户端(G5 511),并连接(G5 521)到网络(G5 513),然后将其再传送(G5 522)到服务器端(G5 515)。

[0297] • 在服务器端,系统会检查它的内存,看用户名“admin”是否存在,如果没有,系统

会显示一条信息:“用户名不存在,请输入有效的用户名”(未显示)。此例与图11B中相同,在存储器中的验证码是:“$=Ⓜ© 2☺”,系统在存储器中发现了它(G5 517)。

[0298] • 系统按图5中所示的符章生成规则生成16个符章(G5 523)。

[0299] • 那16个符章被传送(G5 524)到网络。

[0300] • 网络传送(G5 525)符章到客户端。

[0301] • 那16个符章在用户登录屏幕上,如图12C所示,被显示在4×4表格中(图12C G5 321)。

[0302] • 用户选择6个符章(G5 526):G5 351,G5 354,G5 357,G5 360,G5 363和G5 366。

[0303] • 用户点击“登录”(图12C G5 371)之后,那6个用户选择的符章被传送(G5 527)到网络。

[0304] • 那6个用户选择的符章被传送(G5 528)到服务器端。

[0305] • 在服务器端,系统会逐一检查所有6个符章:K21,K22,K23,K24,K25和K26。在这个例子中,最后3个所选符章是不正确的(需要选择D2符章,C3符章和D1符章分别作为第4,第5和第6符章,但是用户选择了C2符章,B2符章和C4符章,这是错误的。因此,结果是失败的登录)。

[0306] • 登录失败(G5 529)的上述结果被传送(G5 530)到网络。

[0307] • 网络传送(G5 531)结果给客户端,并显示(G5 532)图12C G5 381中所示的信息。

[0308] 图13D以文字型式显示了GATE_5实施方案中,用户登录过程的样本屏幕截图。它显示了用户登录过程示例图12D背后发生了什么。并且显示了图4的处理流程。这只是一个演示,在实际用户登录时不显示。它是用来在视觉上形像说明图7中所示的符章验证规则。

[0309] 图中有3列:“客户端”(左侧),“网络连接”(中)和“服务器端”(右侧)。这个过程从用户在客户端输入他的用户名开始,然后信息通过网络连接到达服务器端。服务器生成16个符章,并将其传送到网络上,然后网络将符章传送到客户端。

[0310] 用户根据图6中所示的符章选择规则选择符章,选择的符章被传送到网络,然后传送到服务器端进行验证。允许或拒绝访问的结果通过网络传送到客户端。处理流程由图13D中箭头表示。处理过程如下:

[0311] • 用户输入用户名“admin”(G5 606)。

[0312] • 用户点击“输入”(G5 609)。

[0313] • 用户名“admin”(G5 606)显示在客户端(G5 611),并(G5 621)连接到网络(G5 613),然后将其再传送(G5 622)到服务器端(G5 615)。

[0314] • 在服务器端,系统会检查它的内存,看用户名“admin”是否存在,如果没有,系统会显示一条信息:“用户名不存在,请输入有效的用户名”(未显示)。此例与图11B中相同,在存储器中的验证码是:“$=Ⓜ©2☺”,系统在存储器中发现了它(G5 617)。

[0315] • 系统按照图5的符章产生的规则生成16个符章(G5 623)。

[0316] • 那16个符章被传送(G5 624)到网络。

[0317] • 网络传送(G5 625)符章到客户端。

[0318] • 那16个符章在用户登录屏幕上,如图12D所示,被显示在4×4表格中(G5 322)。

[0319] • 用户选择5个符章(G5 626):G5 352,G5 355,G5 358,G5 361和G5 364。请注意G5 367,它说“[☺]缺少输入”,这意味着,该系统期望以“☺ “符号作为最后一个销,因此应该有一个第六符章,然而输入的第六个符章空缺,这是一个错误,系统将拒绝用户访问。

[0320] • 用户点击图12D中的“登录”(G5 372)后,5个用户选择的符章被传送(G5 627)到网络。

[0321] • 然后那5个用户选择的符章被传送(G5 628)到服务器侧。

[0322] • 在服务器端,系统会逐一检查所有5个符章:K31,K32,K33,K34和K35,在此例中所有的5个符章都是正确的,但第6个符章空缺(用户没有点击C2,而是留下最后一个位置空白,只输入了5个销),因此K36有一个[X]标志,它代表了一个差错。这是错误的。因此,结果是一个失败的登录。

[0323] • 上面一个登录失败(G5 629)的结果被传送(G5 630)到网络。

[0324] • 网络把结果传送(G5 631)给客户端,并显示(G5 632)图12D G5 382所示的信息。

[0325] 该方法可以进一步扩展到使用以下功能,使猜测验证码更困难:分配一定数目的“隐藏”销,这样,当那些隐藏销出现在选择表中时,它们不被选择。用户必须选择不具有这些销的符章取而代之,避免具有这些销的符章。

[0326] 因此,例如,如果用户的验证码是“123($#)456”,验证码的长度为8,2个销被隐藏在“(”和“)”之间。对销1,2,3,4,5,6遵循上述规则。对于“$”和“#”,遵循“隐藏的销的规则”,它们是:如果他们没有出现在任何16个符章中,用户可以而且必须在他们的销的位置选择通配符代替,但如果这16个符章中的一个出现了一个用户预选的销,用户必须避免这个销并选择其它15个符章中的一个。

[0327] 为了使一个销成为隐藏销,在验证码创建(注册)屏幕的每个销的位置下方可显示一个复选框,当用户选用销下方的复选框时,该销成为一个隐藏的销,在符章验证过程中,使用上述隐藏销规则来验证用户登录。

[0328] 本发明也可以被用于任何通讯过程中,附上有16个符章的选择表[用图5所描述的符章生成规则和发送者的验证码生成],并与一些作为钥匙的符章一起使用。

[0329] 如果钥匙对这个所附的表有效,则该信息是真实的信息。如果钥匙对所附的表无效,那信息是假信息。通过保留真实信息并删除假信息,就会获得最终的[原始]正确的信息。接收者使用相同的验证码来解密该信息,所以该过程可如以下实例进行:

[0330] • 信息_1:我会晚上7点回家[+有16个符章的符章表+无效钥匙]->扔掉信息

[0331] 信息_2:我会在下午3点回家[+有16个符章的符章表+有效钥匙]==>我会在下午3点回家

[0332] 信息_3:我们会放弃3号的攻击[+有16个符章的符章表+无效钥匙]->扔掉信息

[0333] 信息_4:我们会在3号中午攻击[+有16个符章的符章表+有效钥匙]==>我们会在3号中午攻击

[0334] 因此,正确的最终信息是:我会在下午3点回家。我们会在3号中午攻击。

[0335] • 信息_1:我[+有16个符章的符章表+有效钥匙]==>我

[0336] 信息_2:会[+有16个符章的符章表+有效钥匙]==>会

[0337] 信息_3:不会[+有16个符章的符章表+无效钥匙]->扔掉信息

[0338] 信息_4:去[+有16个符章的符章表+有效钥匙]＝＝>去

[0339] 因此,正确的最终信息是:我会去。

[0340] • 信息_1:U[+有16个符章的符章表+无效钥匙]->扔掉信息

[0341] 信息_2:N[+有16个符章的符章表+无效钥匙]->扔掉信息

[0342] 信息_3:T[+有16个符章的符章表+有效钥匙]＝＝>T

[0343] 信息_4:R+有16个符章的符章表+有效钥匙]＝＝>R

[0344] 信息_5:U[+有16个符章的符章表+有效钥匙]＝＝>U

[0345] 信息_6:E[+有16个符章的符章表+有效钥匙]＝＝>E

[0346] 因此,正确的最终信息是:TRUE。

[0347] 本发明还可以防止网络钓鱼,这往往是一种恶意的,获取敏感信息,如用户名,验证码和信用卡信息,假装成在电子通信中可信任的实体的一种尝试手段。因为,在本发明中,在用户和服务器之间传送的所有的信息都是以符章的形式进行的,并且每个符章都有多个符号,没有明确的用户销会被暴露。

[0348] 图14是使用GATE_4实例进行信息加密处理的样本屏幕截图。它显示文字信息是如何用发送者的验证码加密的,并显示了加密过的信息可能看起来是什么样的。处理过程如下:

[0349] • 信息发送者在G4 700输入一个文字信息“secret”。

[0350] • 发送者在G4 702输入验证码“123”并点击“加密”按钮[G4 703]。

[0351] • 原始信息“secret”与一些随机填充字符混合,变成了以下结果信息“sLeQWNcrMfYeMtHQr”,如图G4 704。

[0352] • 接收者在接收端将使用相同的验证码“123”[G4 706]对信息进行解密。

[0353] 图15A是使用GATE_4实例进行信息解密处理的样本屏幕截图。它显示了加密过的原始信息是如何可以用接收端的验证码被成功解密的例子。处理过程如下:

[0354] • 结果信息[G4 704]里的每个字符都有着一个4×4符章表,表中的每个符章都具有4个符号,该信息中的每个字符也都附带一个“钥匙”,钥匙是由符章组成的,钥匙中的符章数在2至6之间。

[0355] • 接收器使用相同的验证码“123”[G4 706]对信息进行解密,解密后的字符信息被显示在G4 704中。结果解密的信息是“secret”[G4 708]。

[0356] • 图15A显示出了每个字符是什么样子的。屏幕截图中用户点击第一个字符“s”[G4 710]后,它作为例子[G4 710]被显示,G4 712显示当前显示的字符为“S”。附带到该字符的4×4符章表被显示于表格G4 714中。

[0357] • 附带到该字符“s”的钥匙也被示为符章:G4 720,G4 722和G4 724。

[0358] • 信息中的填充字符:L,Q,W,N,M,F,Y,M,H,q和r是故意附带符章表和无效钥匙的,因此它们将在接收器侧被确认无效。

[0359] • 在所示的例子中,用户可以点击在G4 704中的每个字符,以显示其内容和钥匙符章,然后点击“检查”键[G4 726],以查看该字符是否有效。在截图中,它显示字符“s”是有效的,检查成功[G4 730]。

[0360] 图15B是使用GATE_4实例进行信息解密处理的样本屏幕截图。它显示了图15A中所示过程的幕后会发生什么。以及如何将信息在接收器侧进行验证。处理过程如下:

[0361] • 信息中字符“S”[G4 750]被用发送验证码“123”[G4 752]进行加密并和有16个符章[G4 754]的4×4表,及一些由符章[G4 720,G4 722和G4 724]组成的钥匙附在一起。这个信息被发送到网络[G4 756],然后再发送到接收端[G4 758]。

[0362] • 在接收端,相同的验证码“123”[G4 760]被用来解码信息。钥匙符章通过[G4 762]验证过程G4 764,G4 766,检查每个钥匙符章。从C51,C52和C53,可以看到他们都是有效的,因此,最后是达到[G4 768]该信息是有效的[G4 770]结论,如图15A[G4 730]所示。

[0363] 图16A是使用GATE_4实例进行信息解密处理的样本屏幕截图。它显示了加密过的填充信息是如何被解密,并由接收器验证码确认为无效信息的一个例子。处理过程如下:

[0364] • 图16A显示了不是原始信息一部分的,每个填充符是什么样子的一个例子。它显示了用户点击信息[G4 704]中的第二个字符“L”[G4 711]后的内容。G4 713显示了字符为“L”。附于该字符的由16个符章组成的4×4表格在G4 715中被示出。“L”附加的4个钥匙符章被示为G4 721,G4 723,G4 725和G4 727。

[0365] • 由于字符“L”是一个填充符而不是原始信息的一部分,发送者故意用一个4×4符章表和一个将不能被验证的钥匙附带着。可以清楚地看到发件人验证码[G4 702]和接收器验证码[G4 706]相同,并且有3个销“1”,“2”和“3”,从而有效的钥匙应有3个符章不多也不少。此例中,钥匙里有4个符章,因此它是无效的,字符“L”应被忽略而不会是最后被解密的信息的一部分。

[0366] • 在截图中,当用户点击“检查”按钮G4 726时,显示说明该验证过程失败[G4 731]。

[0367] 图16B是使用GATE_4实例进行信息解密处理的样本屏幕截图。它显示了图16A中所示过程的幕后会发生什么。以及如何在接收器侧验证填料信息无效。处理过程如下:

[0368] • 信息中字符“L”[G4 751]被用发送验证码“123”[G4 752]进行加密并和有16个符章[G4 755]的4×4表,及一些由符章[G4 721,G4 723,G4 725和G4 727]组成的钥匙附在一起。上述信息被发送到网络[G4 757]然后发送到接收机[G4 759]。

[0369] • 在接收端,相同的验证码“123”[G4 760]被用来解码信息。钥匙符章通过[G4 763]验证过程G4 765,G4 767,检查每一个符章。从C61,C62,C63和C64,可以看到,最后的2个钥匙符章无效。

[0370] • 第三验证码“3”出现在第三个符章[G4 790]中,它应该被选中。然而,第8个符章[G4 792]被选中,并出现在第3个钥匙符章G4 725位置.这是不正确的。

[0371] • 发送验证码等于接收验证码,并且有3个销,但附加的钥匙有4个符章。最后的符章[G4 727]也是无效的。

[0372] • 最后得出结论[G4 769],该信息是无效的[G4 771],如图16A所示[G4731]。

[0373] 图17是使用GATE_4实例进行信息解密处理的样本屏幕截图。它显示了加密后的原始信息如果用不同与发送端钥匙的接收端钥匙,是如何不能被成功解密的例子。处理过程如下:

[0374] • 用户在G4 700输入文字信息“secret”,然后输入验证码“123”[G4 702]并点击“加密”按钮[G4 703]。

[0375] • 信息被加密,发送和接收,并在G4 705出现了:“sLeQWNcrMfYeMtHQr”[G4 705]。

[0376] • 接收器用验证码“567”[G4 707]解密发送者用验证码“123””[G4 702]加密过的

信息。

[0377] • 解密后的信息是:“ecrQ”,在G4 705中被用方框标出。

[0378] • 结果的信息显示为“ecrQ”[G4 709]。

[0379] • 结果的信息与发送者原信息“secret”[G4 700]不同,因为该接收器使用了不同与发送者的验证码来解密该信息。

[0380] 图18是使用GATE_5实例进行信息加密处理的样本屏幕截图。它显示的文字信息是如何用发送者的验证码加密的,并显示了加密过的信息可能看起来是什么样的。处理过程如下:

[0381] • 信息发件人在G5 700输入一个文字信息“FYEO”。

[0382] • 发件人在G5 702输入验证码“123”并单击“加密”按钮[G5 703]。

[0383] • 原始信息“FYEO”与一些随机填充字符混合,变成了以下结果信息“FlPRojcYnEbAO”,如图[G5 704]。

[0384] • 接收器在接收端将使用相同的验证码“123”[G5 706]对信息进行解密。

[0385] 图19A是使用GATE_5实例进行信息解密处理的样本屏幕截图。它显示了加密过的原始信息是如何可以用接收端的验证码被成功解密的例子。处理过程如下:

[0386] • 所接收的信息[G5 704]中的每个字符都有着一个4×4符章表,表中的每个符章都具有5个符号,该信息中的每个字符也都附带一个“钥匙”,钥匙是由符章组成的,钥匙中的符章数在2至6之间。

[0387] • 接收器使用相同的验证码“123”[G5 706]对信息进行解码,解密后的信息被显示在G5 704中。结果解密的信息是“FYEO”[G5 708]。

[0388] • 图19A显示出了每个字符是什么样子的。在截图中,第一个字符“F”[G5 710]被示为一个例子。G5 712显示当前的字符是“F”,G5 714显示该字符附带的4×4符章表。

[0389] • 附带到该字符“F”的钥匙符章也被示为G5 720,G5 722和G5 724。

[0390] • 信息中的填充字符:L,P,R,邻,J,C,N,B和A是故意附带符章表和无效钥匙的,因此它们将在接收器侧被确认无效。

[0391] • 在这个例子中,用户可以点击在G5 704的每个字符,以显示其内容和钥匙符章,然后点击“检查”键[G5 726],以查看该字符是否有效。在截图中,它显示字符“F”是有效的,检查成功[G5 730]。

[0392] 图19B是使用GATE_5实例进行信息解密处理的样本屏幕截图。它显示了图19A中所示过程的幕后会发生什么。以及如何将信息在接收器侧进行验证。处理过程如下:

[0393] • 信息中字符“F”[G5 750]与发送者的验证码“123”[G5 752]进行加密并和有16个符章[G5 754]的4×4表,及一些由符章[G5 720,G5 722和G5 724]组成的钥匙附在一起。这个信息被发送到网络[G5 756]然后发送到接收端[G5 758]。

[0394] • 在接收端,相同的验证码“123”[G5 760]被用来解码信息。钥匙符章通过[G5 762]验证过程G5 764,G5 766,检查每个密钥符章。从K51,K52和K53可以看到它们都是有效的,因此,最后是达到[G5 768]该信息是有效的[G5 770]结论,如图19A[G5 730]所示。

[0395] 图20A是使用GATE_5实例进行信息解密处理的样本屏幕截图。它显示了加密过的填充信息是如何被解密,并由接收器验证码确认为无效信息的一个例子。处理过程如下:

[0396] • 图20A显示了不是原始信息一部分的,每个填充符是什么样子的一个例子。它显

示了信息[G5 704]中第三个字符“P”[G5 711]的内容。G5 713显示字符为“P””。附于该字符的由16个符章组成的4×4表格在G5 715中被示出。

[0397] “P”附加的3个钥匙符章被示为G5 721,G5 723和G5 725。

[0398] • 由于字符“P”是一个填充字符,而不是原来的信息的一部分,发送者故意用一个4×4符章表和一个将不能被验证的钥匙附带着。可以清楚地看到发件人验证码[G5 702]和接收机验证码[G5 706]是相同的,并具有3个销为“1”,“2”和“3”。验证码的第一销“1”出现在4×4表[G5 716]的最后一个符章中,该符章应被选为第一钥匙符章。然而,第二符章[G5 718]在表中被选中,并显示在第一钥匙符章位置G5 721.这是错误的,并会使此信息作废。

[0399] • 在截图中,当用户点击“检查”按钮G5 726后,显示此字符“P”的验证过程失败[G5 731]。

[0400] 图20B是使用GATE_5实例进行信息解密处理的样本屏幕截图。它显示了图20A中所示过程的幕后会发生什么。以及填料信息在接收器端如何确认为无效。处理过程如下:

[0401] • 信息中字符“P”[G5 751]被用发送验证码“123”[G5 752]进行加密并和有16个符章[G5 755]的4×4表,及一些由符章[G5 721,G5 723和G5 725]组成的钥匙附在一起。这个信息被发送到网络[G5 757],然后再发送到接收端[G5 759]。

[0402] • 在接收端,相同的验证码“123”[G5 760]被用来解码信息。钥匙符章通过[G5 763]验证过程G5 765,G5 767,检查每一个符章。从K61,K62和K63可以看到第一符章无效。

[0403] • 第1验证码“1”出现在最后一个符章[G5 790],它应该被选中。然而,第二个符章[G5 792]被选择,并在第一钥匙符章位置[G5 721]被显示出来。因此无效。

[0404] • 一个最后的结论是达到[G5 769]该信息是无效的[G5 771],如图所示。20A[G5 731]。

[0405] 图21是使用GATE_5实例进行信息解密处理的样本屏幕截图。它显示了加密后的原始信息如果用不同与发送端钥匙的接收端钥匙是如何不能被成功解密的例子。处理过程如下:

[0406] • 用户在G5 700输入文字信息“FYEO”,然后输入验证码“123”[G5 702]并点击“加密”按钮[G5 703]。

[0407] • 对信息进行加密,发送和接收,并在G5 705出现了:“F1PRojcYnEbAO”[G5 705]。

[0408] • 接收器用验证码“680”[G5 707]解密发送者用验证码“123”[G5 702]加密过的信息。

[0409] • 解密后的信息是:“NE”,在G5 705中被用方框标出。

[0410] • 结果的信息显示为“NE”[G5 709]。

[0411] • 结果的信息与发送者原信息“FYEO”[G5 700]不同,因为该接收器中使用了不同与发送器的验证码来解密该信息。

[0412] 优选使用下面的步骤,对验证码中的每个销,生成一组对4×4表中的16符章有效的钥匙符章:

[0413] • 对所有的16个符章:(a)如果销在符章中发现,选择此符章。及(b)如果销不在任何符章中,挑选该表16个符章中随机的一个符章。

[0414] 优选使用下面的步骤对每个无效的信息,产生一组,对具有16个符章的4×4表,无效的钥匙符章:

[0415] <1>设置布尔“Done_Fixing”为false。

[0416] <2>检查所有的16个符章,对验证码中的每个销执行步骤<3>和<4>

[0417] <3><A>如果在符章中发现销:

[0418] (1)如果Done_Fixing等于false,故意挑选除这个之外的其他任何错误的符章,并设置Done_Fixing为true。

[0419] (2)如果Done_Fixing等于true,挑选该符章。

[0420] <B>如果销不在任何符章中,挑选该表16个符章中随机的一个符章。

[0421] <4>把以上生成的钥匙符章储存到一个向量中。

[0422] <5>生成一个随机数N:范围在-1到1之间。

[0423] <A>如果N=-1,删除向量中的最后一个钥匙符章。

[0424] <B>如果N=0,什么也不做。

[0425] <C>如果N=1并且用户销长<6,从表中16个符章里挑选一个随机符章添加到向量中。

[0426] <6>在向量中的符章将是最后的钥匙符章。

[0427] 上述实例和优点仅是示范性的,而不应当被解释为限制本发明的。本发明的描述是说明性的,而不是限制权利要求范围的。许多替换,修改和变化对那些熟练的技术人员将是显而易见的。在不脱离本发明的精神和范围的例子中可以进行各种变化,如下面的权利要求所定义。

[0428] 例如,虽然本发明结合了GATE_4和GATE_5的实例,其中4维和5维符号被分别使用,任意维数(包括仅一维)可被使用而仍在本发明的范围之内。在一般例子中,只要每个符章中有一个以上的符号,可以使用任何数目的和任何维数的符号。此外,上述的GATE_4和GATE_5实例,以及相关的屏幕截图,是说明性的,而不是限制本发明范围的。

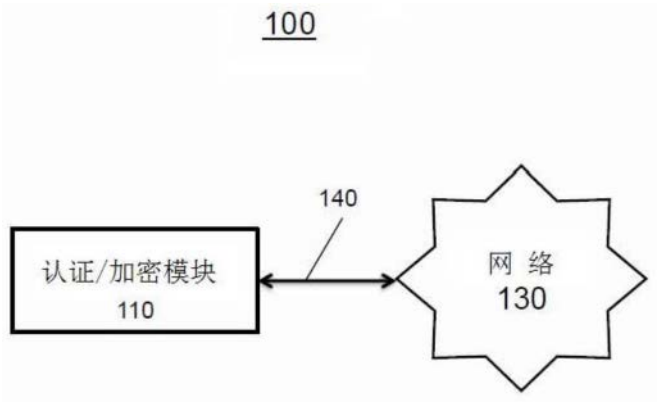
100
140
认证/加密模块
110
网 络
130

图1A

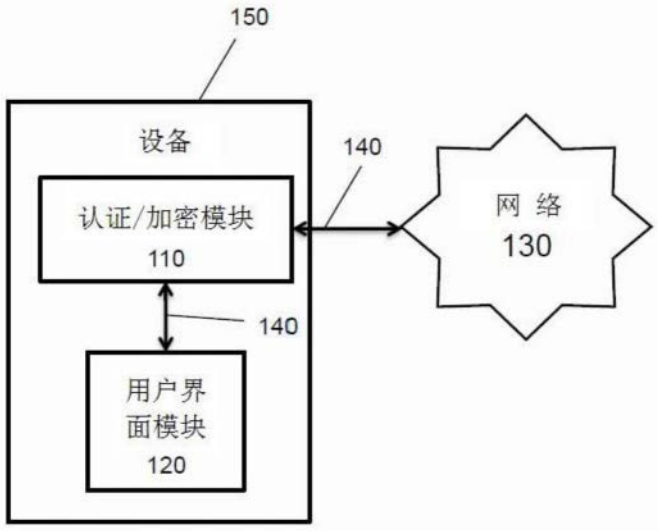
150
设备
140
认证/加密模块
110
网 络
130
140
用户界
面模块
120

图1B

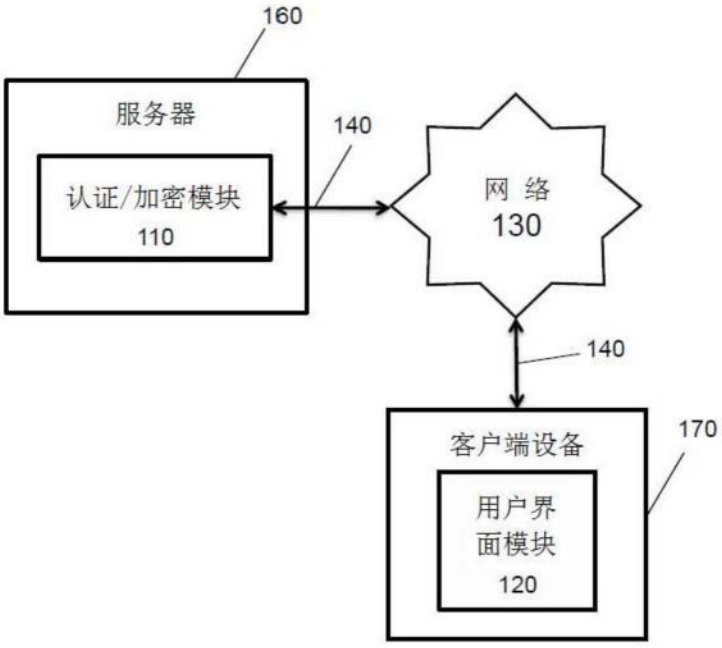
160
服务器
认证/加密模块
110
140
网 络
130
140
客户端设备
用户界
面模块
120
170

图1C

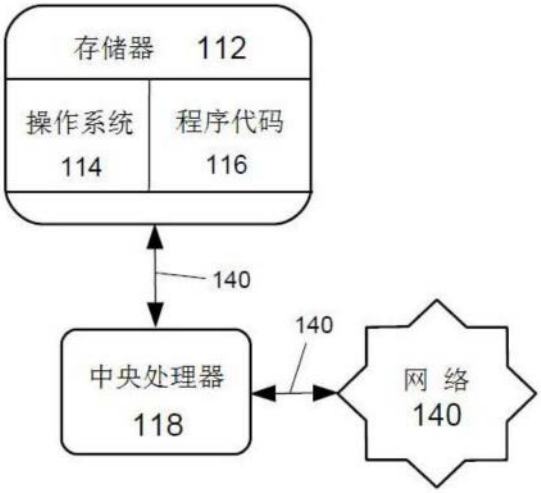
存储器 112
操作系统
114
程序代码
116
140
中央处理器
118
140
网 络
140

图1D

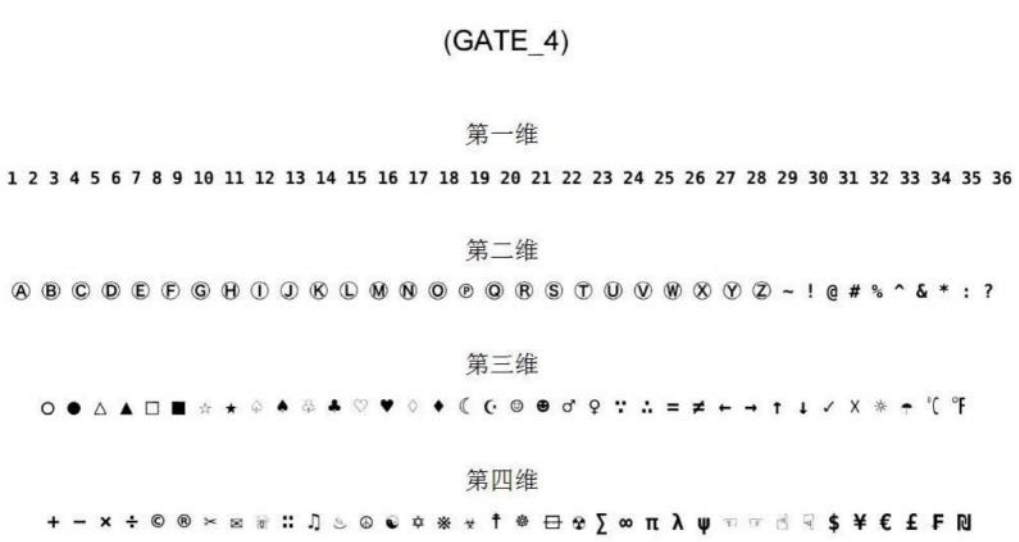
(GATE_4)
第一维
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36
第二维
Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ Ⓚ Ⓛ Ⓜ Ⓝ Ⓞ Ⓟ Ⓠ Ⓡ Ⓢ Ⓣ Ⓤ Ⓥ Ⓦ Ⓧ Ⓨ Ⓩ ~ ! @ # % ^ & * : ?
第三维
○ ● △ ▲ □ ■ ☆ ★ ♤ ♠ ♧ ♣ ♡ ♥ ♢ ♦ ☾ ☉ ☺ ☻ ♂ ♀ ∵ ∴ = ≠ ← → ↑ ↓ ✓ X ✳ ☂ ℃ ℉
第四维
+ − × ÷ © ® ✂ ✉ ☏ :: ♫ ♨ ☮ ☯ ✡ ❋ ☣ ✝ ☸ ⊟ ☢ ∑ ∞ π λ ψ ☜ ☞ ☝ ☟ $ ¥ € £ ₣ ₪

图2A

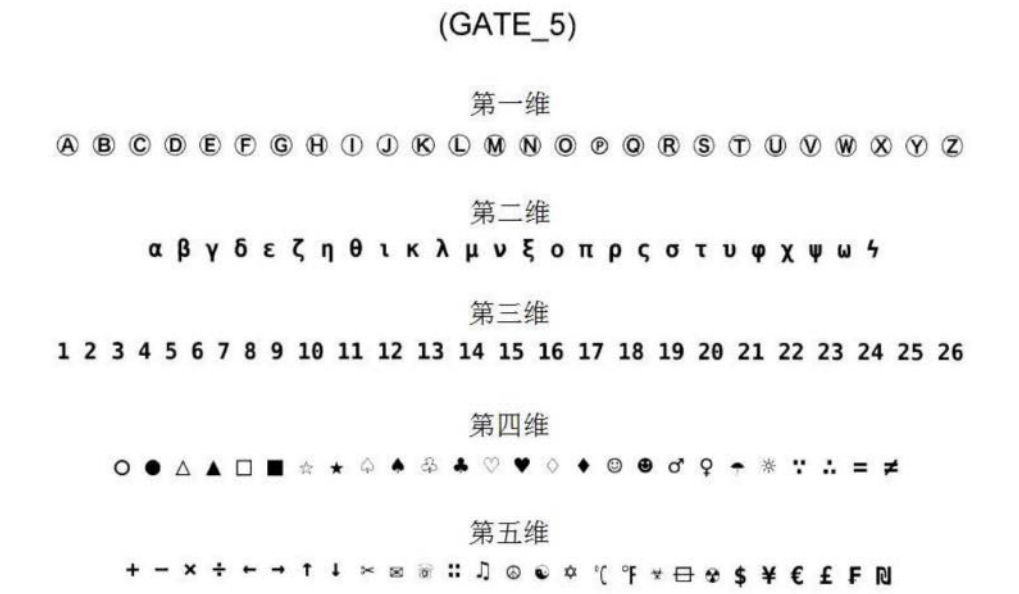
(GATE_5)
第一维
Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ Ⓚ Ⓛ Ⓜ Ⓝ Ⓞ Ⓟ Ⓠ Ⓡ Ⓢ Ⓣ Ⓤ Ⓥ Ⓦ Ⓧ Ⓨ Ⓩ
第二维
α β γ δ ε ζ η θ ι κ λ μ ν ξ ο π ρ ς σ τ υ φ χ ψ ω ϟ
第三维
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
第四维
○ ● △ ▲ □ ■ ☆ ★ ♤ ♠ ♧ ♣ ♡ ♥ ♢ ♦ ☺ ☻ ♂ ♀ ☂ ✳ ∵ ∴ = ≠
第五维
+ − × ÷ ← → ↑ ↓ ✂ ✉ ☏ :: ♫ ☮ ☯ ✡ ℃ ℉ ☣ ⊟ ☢ $ ¥ € £ ₣ ₪

图2B

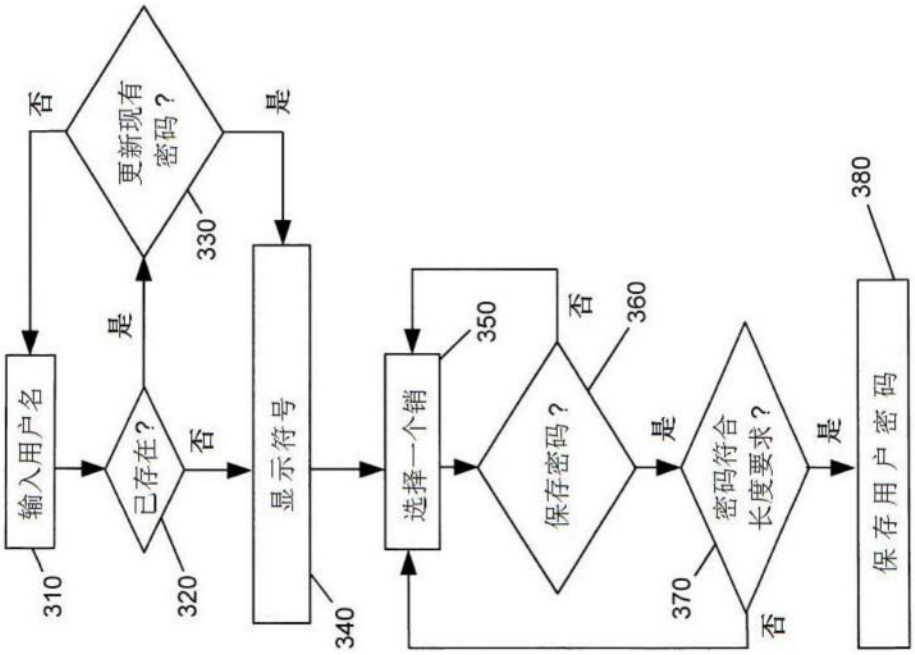
输入用户名
已存在?
更新现有密码?
显示符号
选择一个销
保存密码?
密码符合长度要求?
保存用户密码
否
是
310
320
330
340
350
360
370
380

图3

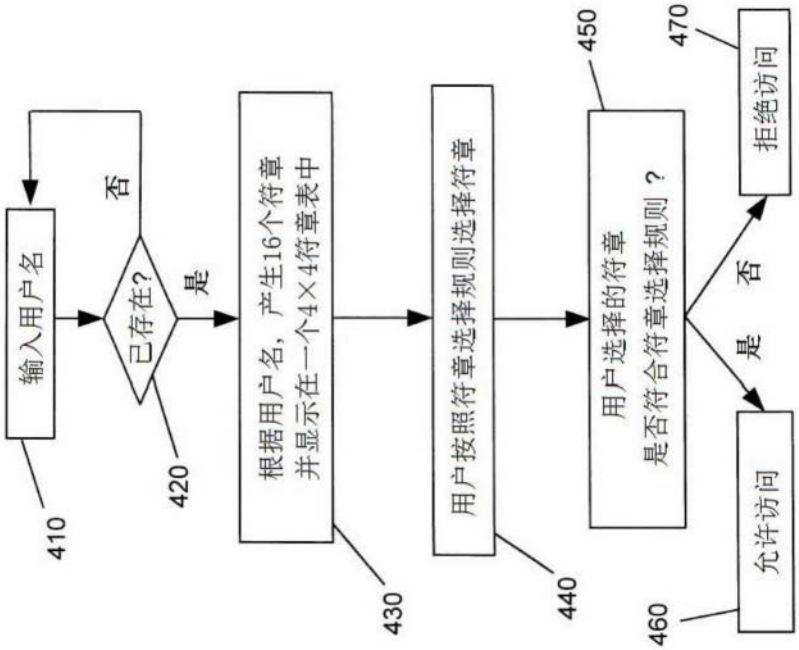
输入用户名
已存在?
根据用户名，产生16个符章并显示在一个4×4符章表中
用户按照符章选择规则选择符章
用户选择的符章是否符合符章选择规则?
拒绝访问
允许访问
否
是
410
420
430
440
450
460
470

图4

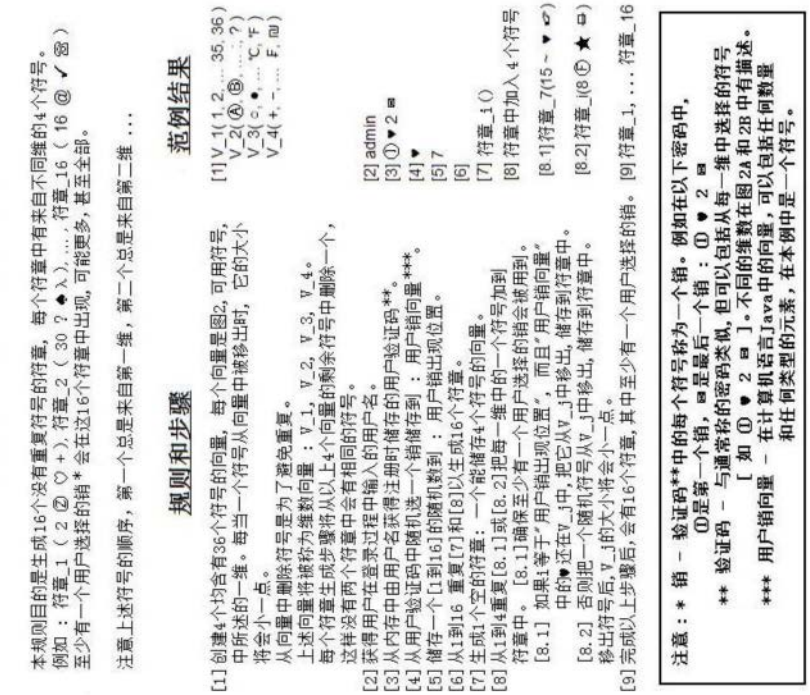
[用 GATE_4 作实例说明 (参见图 2A)]
本规则目的是生成16个没有重复符号的符号，每个符号中有来自不同维的4个符号。
例如：符号_1 (2 ⓩ ♡ +)，符号_2 (30 ? ♠ λ)，... ，符号_16 (16 @ ✓ ☏)
至少有一个用户选择的销 * 会在这16个符号中出现，可能更多，甚至全部。
注意上述符号的顺序，第一个总是来自第一维，第二个总是来自第二维 ...
规则和步骤
范例结果
[1] 创建4个均含有36个符号的向量，每个向量是图2，可用符号，中所述的一维。每当一个符号从向量中被移出时，它的大小将会小一点。
从向量中删除符号是为了避免重复。
上述向量将被称为维数向量 ：V_1，V_2，V_3，V_4。
每个符号生成步骤将从上4个向量的剩余符号中删除一个，这样没有两个符号会有相同的符号。
[2] 获得用户在登录过程中输入的用户名。
[3] 从内存中由用户名获得注册时储存的用户验证码**。
[4] 从用户验证码中随机选一个销储存到 ：用户销向量***。
[5] 储存一个[1到16]的随机数到 ：用户销出现位置。
[6] 从1到16 重复[7]和[8]以生成16个符号。
[7] 生成1个空的符号： 一个能储存4个符号的向量。
[8] 从1到4重复[8.1]或[8.2]把每一维中的一个符号加到符号向量中。 [8.1]确保至少有一个用户选择的销会被用到。
[8.1] 如果i等于"用户销出现位置"，而且"用户销向量"中的♥还在V_j中，把它从V_j中移出，储存到符号向量中。
[8.2] 否则把一个随机符号从V_j中移出，储存到符号向量中。
移出符号后，V_j的大小将会小一点。
[9] 完成以上步骤后，会有16个符号，其中至少有一个用户选择的销。
[1] V_1(1, 2, ... 35, 36)
V_2(Ⓐ Ⓑ, ... ?)
V_3(○, ●, ... ☜, ☞)
V_4(+, -, ... ₤, ₪)
[2] admin
[3] ① ♥ 2 ◙
[4] ♥
[5] 7
[6]
[7] 符号_1 ()
[8] 符号中加入4个符号
[8.1]符号_7(15 ~ ♥ ↶)
[8.2] 符号_1(8 Ⓔ ★ ◘)
注意：* 销 – 验证码**中的每个符号称为一个销。例如在以下密码中，
①是第一个销，◙是最后一个销 ：① ♥ 2 ◙
** 验证码 – 与通常你的密码类似，但可以包括从每一维中选择的符号
[如 ① ♥ 2 ◙]。不同的维数在图2A和2B中有描述。
*** 用户销向量 – 在计算机语言Java中的向量，可以包括任何数量和任何类型的元素，在本例中是一个符号。

图5

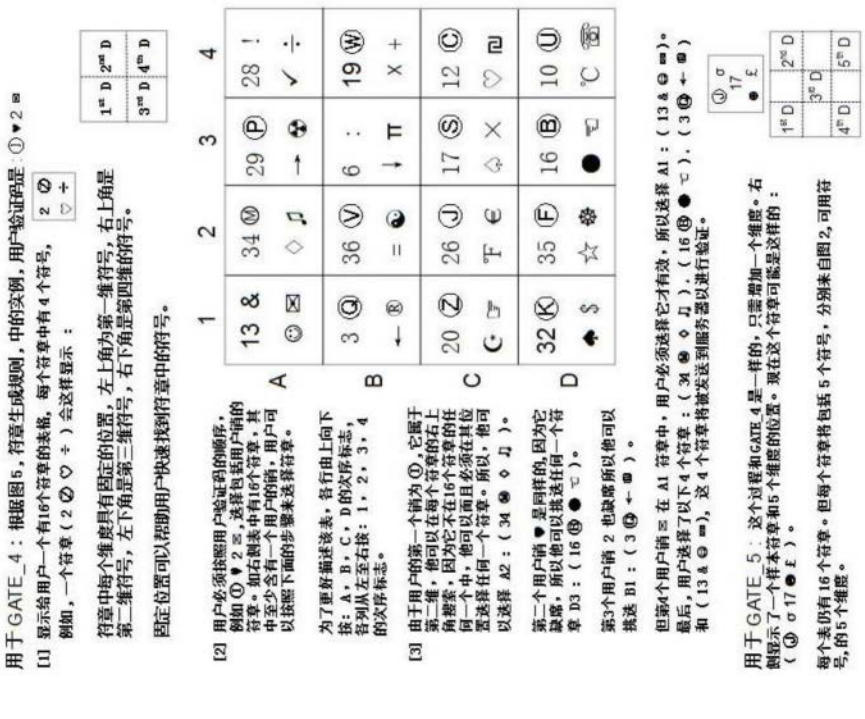
用于GATE_4：根据图5，符号生成规则，中的实例，用户验证码是：① ♥ 2 ◙
[1] 显示给用户一个有16个符号的表格，每个符号中有4个符号，
例如，一个符号 (2 Ø ♡ ÷) 会这样显示：
符号中每个维度具有固定的位置，左上角为第一维符号，右上角是第二维符号，左下角是第三维符号，右下角是第四维的符号。
固定位置可以帮助用户快速找到符号中的符号。
[2] 用户必须按照用户验证码的顺序，例如① ♥ 2 ◙，选择包括用户销的符号。如右图表中有16个符号，其中至少含有一个用户的销，用户可以按照下面的步骤来选择符号。
为了更好描述该表，各行由上向下按：A，B，C，D的次序标志，各列从左至右按：1，2，3，4的次序标志。
[3] 由于用户的第一个销为①，它属于第二维，他可以在每个符号的右上角搜索，因为它不在16个符号的任何一个中，他可以而且必须在其位置选择任何一个符号。所以，他可以选择 A2：(34 ☻ ◇ ♫)。
第二个用户销 ♥ 是同样的，因为它缺席，所以他可以挑选任何一个符号 D3：(16 Ⓟ ● ☜)。
第3个用户销 2 也缺席所以他可以挑选 B1：(3 Ⓠ ← ®)。
但第4个用户销 ◙ 在 A1 符号中，用户必须选择它才有效，所以选择 A1：(13 & ☺ ✉)。
最后，用户选择了以下4个符号：(34 ☻ ◇ ♫)，(16 Ⓟ ● ☜)，(3 Ⓠ ← ®)
和 (13 & ☺ ✉)，这4个符号将被发送到服务器以进行验证。
用于GATE_5：这个过程和GATE_4是一样的，只需增加一个维度。右侧显示了一个样本符号和5个维度的位置。现在这个符号可能是这样的：(Ⓐ σ 17 ● £)。
每个表仍有16个符号。但每个符号将包括5个符号，分别来自图2，可用符号，的5个维度。
A
B
C
D
1 2 3 4
13 & 34 Ⓜ 29 Ⓟ 28 !
☺ ✉ ◇ ♫ → ☢ ✓ ÷
3 Ⓠ 36 Ⓥ 6 : 19 Ⓦ
← ® = ☯ ↓ π X +
20 Ⓩ 26 Ⓙ 17 Ⓢ 12 Ⓒ
☞ ₣ ∈ ♤ × ♡ ₪
32 Ⓚ 35 Ⓕ 16 Ⓑ 10 Ⓤ
♠ $ ☆ ❀ ● ☜ ℃ ☏
1st D 2nd D
3rd D 4th D
Ⓐ σ 17 ● £
1st D 2nd D
3rd D
4th D 5th D

图6

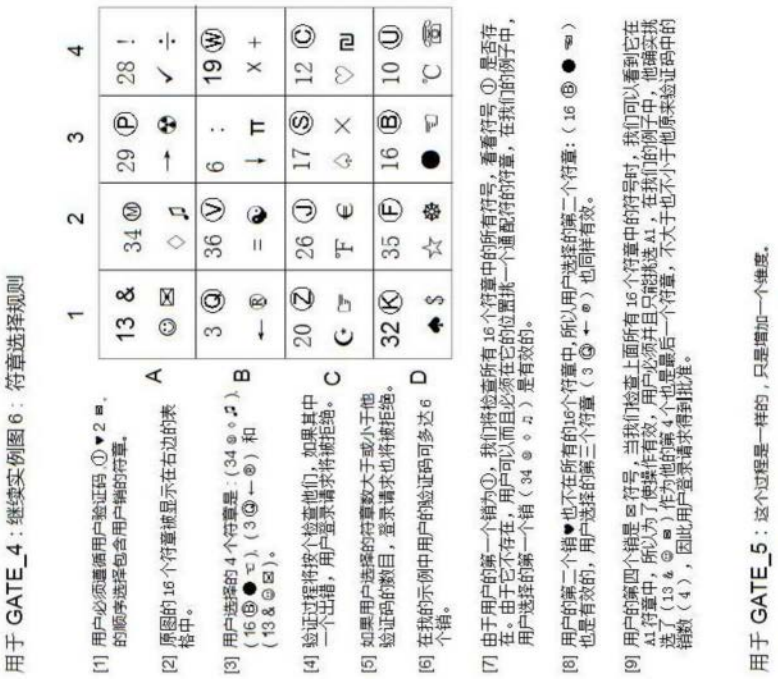
用于 GATE_4：继续实例图 6： 符章选择规则
1
2
3
4
13 &
34
29
28 !
A
3
36
6 :
19
B
20
26
17
12
C
32
35
16
10
D
用于 GATE_5：这个过程是一样的，只是增加一个维度。

图7

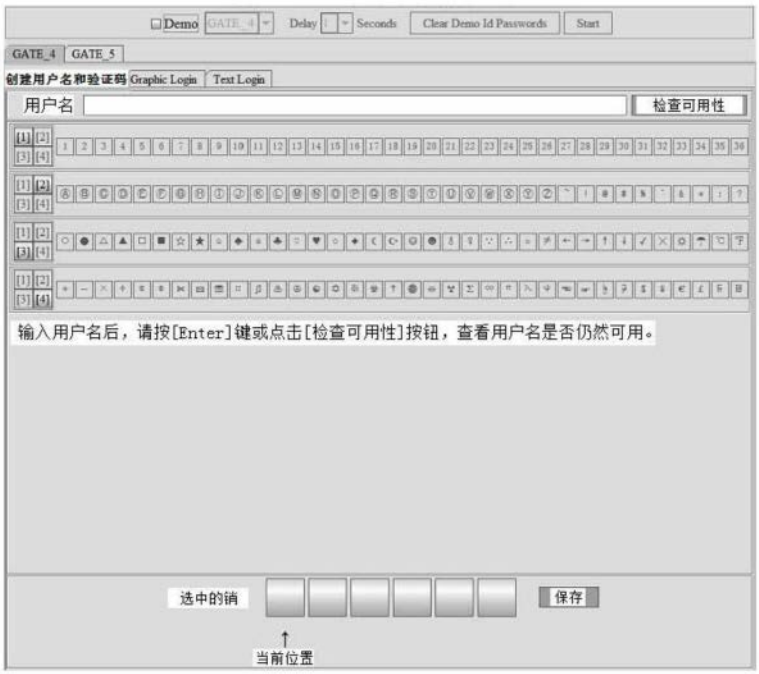
Demo
Delay
Seconds
Clear Demo Id Passwords
Start
GATE_4
GATE_5
创建用户名和验证码
Graphic Login
Text Login
用户名
检查可用性
输入用户名后，请按[Enter]键或点击[检查可用性]按钮，查看用户名是否仍然可用。
选中的销
保存
当前位置

图8A


Demo
GATE_4
Delay
Seconds
Clear Demo Id Passwords
Start
GATE_4
GATE_5
创建用户名和验证码
Graphic Login
Text Login
用户名
admin
G4 206
检查可用性
G4 208
G4 210
G4 212
G4 220
G4 222
G4 230
G4 232
G4 240
G4 242
从上述任何符号中挑选至少4个销。有4类符号：[1]数字，[2]字符，[3]标记，[4]其他符号。
你可以从同一类或不同类别中选择。你可以多次选择同样的销：
这些销将是你登录时的验证码，挑选你容易记住，而别人却很难猜到的记号。
G4 250
G4 252
G4 254
G4 256
G4 258
G4 260
选中的销
2
保存
G4 270
当前位置

图8B


Demo
Delay
Seconds
Clear Demo Id Passwords
Start
GATE_4
GATE_5
Create User Id & Pin
图形登录
Text Login
用户名
输入
[A] 在中间的表内有4×4=16个符章按钮。每个符章按钮有4个符号。它们布置成4组：
[1]数字
[2]字符
[3]标记
[4]其他符号
将鼠标移到上面符章样本上后，可看到所有各组中的36个符号。上述各组的位置在任何符章中是固定的，这样数字总是在左上角显示，字符始终在符章的右上角显示，等等。这能帮助你从中间的符章表内快速查找并输入销→
例如，如果你的验证码是：
那么下列符章含有你的销，因此，将是有效的。
[B] 规则如下：
[1]输入你的用户名。
[2]点击[输入]键或按钮。
[3]按照你的销的顺序。
[4]←从中间的表内选择包括你的销的符章。
[5]如果你的销不在表内的任何符章中，你可以，而且必须，为该销选择任何一个符章。
[6]在你选择全部符章后点击[登录]按钮。
每组中各有36个符号，但表内只有 4×4 = 16 个符章，所以一些符号将会缺席。这是一项安全功能，是故意这么设计的，使你的验证码更难被猜到。
符章
登录
当前位置

图9A


Demo
GATE_4
Delay 1 Seconds
Clear Demo Id Passwords
Start
GATE_4
GATE_5
Create User Id & Pin
图形登录
Text Login
G4 309
用户名
admin G4 306
输入
登录结果
登录成功
OK G4 380
[A] 在中间的表内有4×4=16个符章按钮。每个符章按钮有4个符号。它们布置成4组：
[1]数字
[2]字符
[3]标记
[4]其他符号
将鼠标移到上面符章样本上后，可看到所有各组中的36个符号。上述各组的位置在任何符章中是固定的，这样数字总是在左上角显示，字符始终在符章的右上角显示，等等。这能帮助你从中间的符章表内快速查找并输入销→
例如，如果你的验证码是：
那么下列符章含有你的销，因此，将是有效的。
[B] 规则如下：
[1]输入你的用户名。
[2]点击[输入]键或按钮。
[3]按照你的销的顺序。
[4]←从中间的表内选择包括你的销的符章。
[5]如果你的销不在表内的任何符章中，你可以，而且必须，为该销选择任何一个符章。
[6]在你选择全部符章后点击[登录]按钮。
每组中各有36个符号，但表内只有4×4=16个符章，所以一些符号将会缺席。这是一项安全功能，是故意这么设计的，使你的验证码更难被猜到。
1 2 3 4
A B C D
G4 320
符章
G4 350
G4 353
G4 356
G4 359
G4 362
G4 365
登录
G4 370
当前位置

图9B


Demo
GATE_5
Delay 1 Seconds
Clear Demo Id Passwords
Start
GATE_4
GATE_5
Create User Id & Pin
图形登录
Text Login
G4 310
用户名
admin G4 307
输入
登录结果
登录失败
OK G4 381
[A] 在中间的表内有4×4=16个符章按钮。每个符章按钮有4个符号。它们布置成4组：
[1]数字
[2]字符
[3]标记
[4]其他符号
将鼠标移到上面符章样本上后，可看到所有各组中的36个符号。上述各组的位置在任何符章中是固定的，这样数字总是在左上角显示，字符始终在符章的右上角显示，等等。这能帮助你从中间的符章表内快速查找并输入销→
例如，如果你的验证码是：
那么下列符章含有你的销，因此，将是有效的。
[B] 规则如下：
[1]输入你的用户名。
[2]点击[输入]键或按钮。
[3]按照你的销的顺序。
[4]←从中间的表内选择包括你的销的符章。
[5]如果你的销不在表内的任何符章中，你可以，而且必须，为该销选择任何一个符章。
[6]在你选择全部符章后点击[登录]按钮。
每组中各有36个符号，但表内只有4×4=16个符章，所以一些符号将会缺席。这是一项安全功能，是故意这么设计的，使你的验证码更难被猜到。
1 2 3 4
A B C D
G4 321
符章
G4 351
G4 354
G4 357
G4 360
G4 363
G4 366
登录
G4 371
当前位置

图9C


Demo
Delay
Seconds
Clear Demo Id Passwords
Start
GATE_4
GATE_5
Create User Id & Pin
图形登录
Text Login
用户名
admin
G4 308
输入
G4 311
登录结果
登录失败
OK
G4 382
[A] 在中间的表内有4×4=16个符章按钮。每个符章按钮有4个符号。它们布置成4组：
[1]数字
[2]字符
[3]标记
[4]其他符号
将鼠标移到上面符章样本上后，可看到所有各组中的36个符号。上述各组的位置在任何符章中是固定的，这样数字总是在左上角显示，字符始终在符章的右上角显示，等等。这能帮助你从中间的符章表内快速查找并输入销→
例如，如果你的验证码是：
1♥2
那么下列符章含有你的销，因此，将是有效的。
[B] 规则如下：
[1]输入你的用户名。
[2]点击[输入]键或按钮。
[3]按照你的销的顺序。
[4]←从中间的表内选择包括你的销的符章。
[5]如果你的销不在表内的任何符章中，你可以，而且必须，为该销选择任何一个符章。
[6]在你选择全部符章后点击[登录]按钮。
每组中各有36个符号，但表内只有 4×4 = 16 个符章，所以一些符号将会缺席。这是一项安全功能，是故意这么设计的，使你的验证码更难被猜到。
1
2
3
4
A
B
C
D
G4 322
符章
G4 352
G4 355
G4 358
G4 361
G4 364
G4 367
当前位置
登录
G4 372

图9D

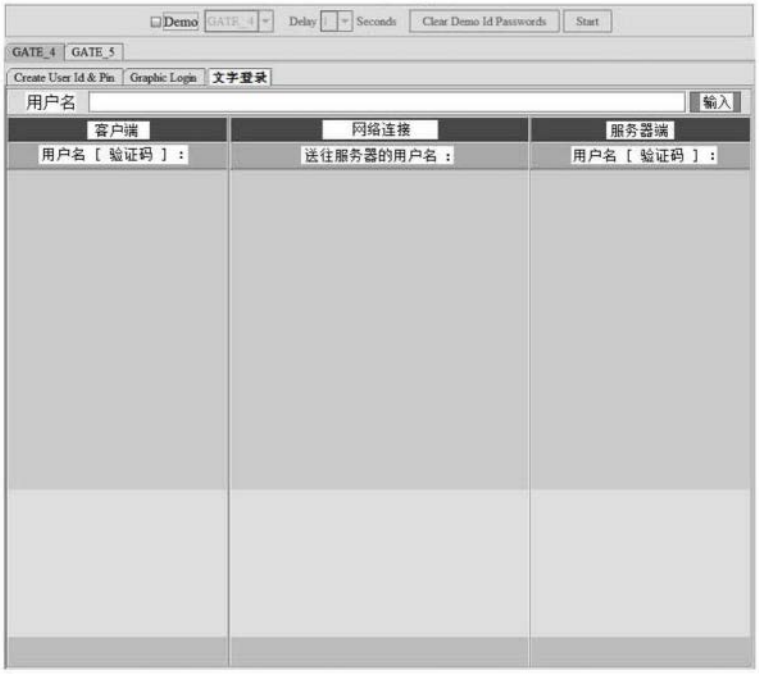
Demo
Delay
Seconds
Clear Demo Id Passwords
Start
GATE_4
GATE_5
Create User Id & Pin
Graphic Login
文字登录
用户名
输入
客户端
网络连接
服务器端
用户名 [验证码] :
送往服务器的用户名 :
用户名 [验证码] :

图10A


Demo
GATE_4
Delay
Seconds
Clear Demo Id Passwords
Start
GATE_4
GATE_5
Create User Id & Pin
Graphic Login
文字登录
用户名
admin
输入
客户端
网络连接
服务器端
用户名：admin
送往服务器的用户名：admin
用户名[验证码]:admin
从网络得到以下符章
网络把信息从服务器送到客户端
由文件中的验证码生成
选择包含用户销的符章
把被选中的符章从客户端送到服务器
从网络得到被选中的符章
允许访问
把登录结果传递给客户端
检查客户端符章

图10B


Demo
GATE_5
Delay
Seconds
Clear Demo Id Passwords
Start
GATE_4
GATE_5
Create User Id & Pin
Graphic Login
文字登录
用户名
admin
输入
客户端
网络连接
服务器端
用户名：admin
送往服务器的用户名：admin
用户名[验证码]:admin
从网络得到以下符章
网络把信息从服务器送到客户端
由文件中的验证码生成
选择包含用户销的符章
把被选中的符章从客户端送到服务器
从网络得到被选中的符章
拒绝访问
把登录结果传递给客户端
检查客户端符章

图10C


Demo
Delay
Seconds
Clear Demo Id Passwords
Start
GATE_4
GATE_5
Create User Id & Pin
Graphic Login
文字登录
用户名
admin
输入
客户端
网络连接
服务器端
用户名 : admin
送往服务器的用户名 : admin
用户名[验证码]:admin
从网络得到以下符章
网络把信息从服务器送到客户端
由文件中的验证码生成
选择包含用户销的符章
把被选中的符章从客户端送到服务器
从网络得到被选中的符章
拒绝访问
把登录结果传递给客户端
检查客户端符章

图10D

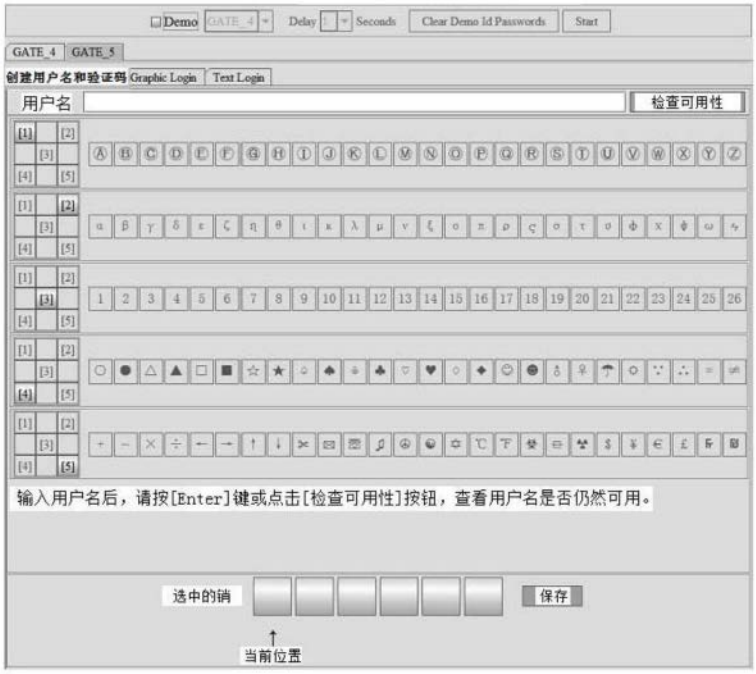
Demo
Delay
Seconds
Clear Demo Id Passwords
Start
GATE_4
GATE_5
创建用户名和验证码
Graphic Login
Text Login
用户名
检查可用性
输入用户名后，请按[Enter]键或点击[检查可用性]按钮，查看用户名是否仍然可用。
选中的销
保存
当前位置

图11A


Demo
GATE_5
Delay
Seconds
Clear Demo Id Passwords
Start
GATE_4
GATE_5
创建用户名和验证码
Graphic Login
Text Login
用户名
admin
G5 206
G5 208
检查可用性
G5 210
G5 212
G5 220
G5 222
G5 230
G5 232
G5 240
G5 242
G5 246
G5 248
从上述任何符号中挑选至少4个销。有5类符号：[1]英文字母，[2]希腊字符，[3]数字，[4]符号，[5]其他。
你可以从同一类或不同类别中选择。你可以多次选择同样的销：
这些销将是你登录时的验证码，挑选你容易记住，而别人却很难猜到的记号。
选中的销
G5 250
G5 252
G5 254
G5 256
G5 258
G5 260
保存
G5 270
当前位置

图11B


Demo
Delay
Seconds
Clear Demo Id Passwords
Start
GATE_4
GATE_5
Create User Id & Pin
图形登录
Text Login
用户名
输入
[A] 在中间的表内有4×4=16个符章按钮。每个符章按钮有5个符号。它们布置成5组：
[1]英文字母
[2]希腊字符
[3]数字
[4]符号
[5]其他
将鼠标移到上面符章样本上后，可看到所有各组中的26个符号。上述各组的位置在任何符章中是固定的，这样英文字母总是在左上角显示，数字始终在符章的中间显示，等等。这能帮助你从中间的符章表内快速查找并输入销→
例如，如果你的验证码是：
那么下列符章含有你的销，因此，将是有效的。
[B] 规则如下：
[1]输入你的用户名。
[2]点击[输入]键或按钮。
[3]按照你的销的顺序。
[4]←从中间的表内选择包括你的销的符章。
[5]如果你的销不在表内的任何符章中，你可以，而且必须，为该销选择任何一个符章。
[6]在你选择全部符章后点击[登录]按钮。
每组中各有26个符号，但表内只有 4×4=16 个符章，所以一些符号将会缺席。这是一项安全功能，是故意这么设计的，使你的验证码更难被猜到。
符章
登录
当前位置

图12A

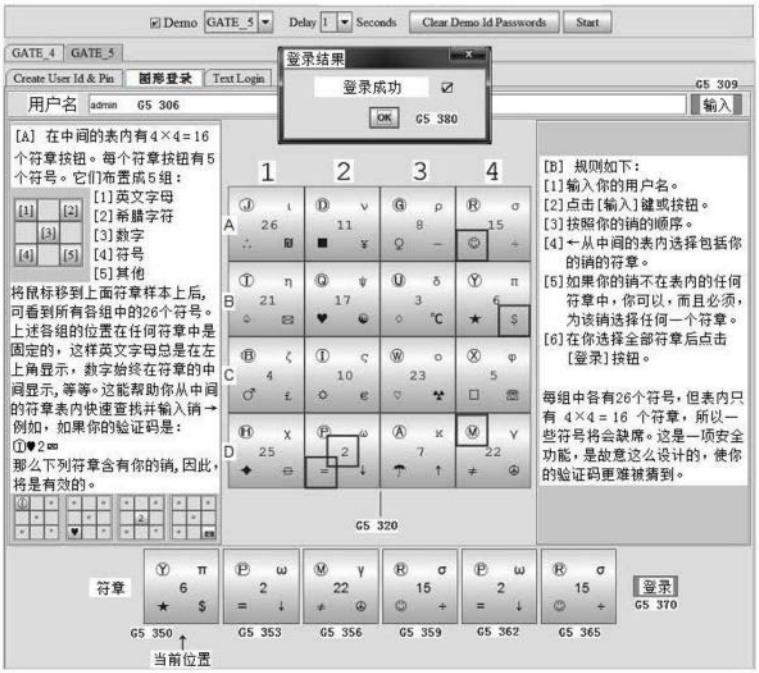
Demo
GATE_5
Delay 1 Seconds
Clear Demo Id Passwords
Start
GATE_4
GATE_5
Create User Id & Pin
图形登录
Text Login
用户名
admin
G5 306
登录结果
登录成功
OK
G5 380
G5 309
输入
[A] 在中间的表内有4×4=16个符章按钮。每个符章按钮有5个符号。它们布置成5组：
[1]英文字母
[2]希腊字符
[3]数字
[4]符号
[5]其他
将鼠标移到上面符章样本上后，可看到所有各组中的26个符号。上述各组的位置在任何符章中是固定的，这样英文字母总是在左上角显示，数字始终在符章的中间显示，等等。这能帮助你从中间的符章表内快速查找并输入销→
例如，如果你的验证码是：
①♥2∞
那么下列符章含有你的销，因此，将是有效的。
[B] 规则如下：
[1]输入你的用户名。
[2]点击[输入]键或按钮。
[3]按照你的销的顺序。
[4]←从中间的表内选择包括你的销的符章。
[5]如果你的销不在表内的任何符章中，你可以，而且必须，为该销选择任何一个符章。
[6]在你选择全部符章后点击[登录]按钮。
每组中各有26个符号，但表内只有 4×4=16 个符章，所以一些符号将会缺席。这是一项安全功能，是故意这么设计的，使你的验证码更难被猜到。
G5 320
符章
登录
G5 370
G5 350
G5 353
G5 356
G5 359
G5 362
G5 365
当前位置

图12B


Demo
GATE_5
Delay
Seconds
Clear Demo Id Passwords
Start
GATE_4
GATE_5
Create User Id & Pin
图形登录
Text Login
用户名
admin
G5 307
登录结果
登录失败
OK
G5 381
G5 310
输入
[A] 在中间的表内有4×4=16个符章按钮。每个符章按钮有5个符号。它们布置成5组：
[1]英文字母
[2]希腊字符
[3]数字
[4]符号
[5]其他
将鼠标移到上面符章样本上后，可看到所有各组中的26个符号。上述各组的位置在任何符章中是固定的，这样英文字母总是在左上角显示，数字始终在符章的中间显示，等等。这能帮助你从中间的符章表内快速查找并输入销→
例如，如果你的验证码是：
①♥2∞
那么下列符章含有你的销，因此，将是有效的。
[B] 规则如下：
[1]输入你的用户名。
[2]点击[输入]键或按钮。
[3]按照你的销的顺序。
[4]←从中间的表内选择包括你的销的符章。
[5]如果你的销不在表内的任何符章中，你可以，而且必须，为该销选择任何一个符章。
[6]在你选择全部符章后点击[登录]按钮。
每组中各有26个符号，但表内只有 4×4=16 个符章，所以一些符号将会缺席。这是一项安全功能，是故意这么设计的，使你的验证码更难被猜到。
G5 321
符章
登录
G5 371
G5 351
G5 354
G5 357
G5 360
G5 363
G5 366
当前位置

图12C


Demo
GATE_5
Delay
Seconds
Clear Demo Id Passwords
Start
GATE_4
GATE_5
Create User Id & Pin
图形登录
Text Login
用户名
admin
G5 308
G5 311
输入
登录结果
登录失败
OK
G5 382
[A] 在中间的表内有4×4=16个符章按钮。每个符章按钮有5个符号。它们布置成5组：
[1]英文字母
[2]希腊字符
[3]数字
[4]符号
[5]其他
将鼠标移到上面符章样本上后，可看到所有各组中的26个符号。上述各组的位置在任何符章中是固定的，这样英文字母总是在左上角显示，数字始终在符章的中间显示，等等。这能帮助你从中间的符章表内快速查找并输入销→例如，如果你的验证码是：
[B] 规则如下：
[1]输入你的用户名。
[2]点击[输入]键或按钮。
[3]按照你的销的顺序。
[4]←从中间的表内选择包括你的销的符章。
[5]如果你的销不在表内的任何符章中，你可以，而且必须，为该销选择任何一个符章。
[6]在你选择全部符章后点击[登录]按钮。
每组中各有26个符号，但表内只有 4×4=16 个符章，所以一些符号将会缺席。这是一项安全功能，是故意这么设计的，使你的验证码更难被猜到。
那么下列符章含有你的销，因此，将是有效的。
G5 322
符章
G5 352
G5 355
G5 358
G5 361
G5 364
G5 367
登录
G5 372
当前位置

图12D

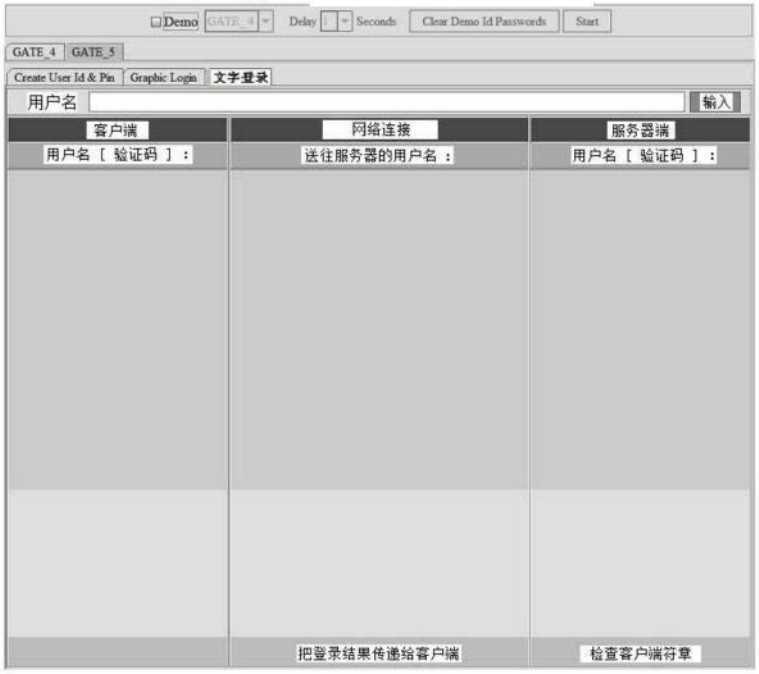
Demo
Delay
Seconds
Clear Demo Id Passwords
Start
GATE_4
GATE_5
Create User Id & Pin
Graphic Login
文字登录
用户名
输入
客户端
网络连接
服务器端
用户名 [验证码] :
送往服务器的用户名 :
用户名 [验证码] :
把登录结果传递给客户端
检查客户端符章

图13A


Demo
GATE_5
Delay
Seconds
Clear Demo Id Passwords
Start
GATE_4
GATE_5
Create User Id & Pin
Graphic Login
文字登录
用户名
admin
输入
客户端
网络连接
服务器端
用户名：admin
送往服务器的用户名：admin
用户名[验证码]:admin
从网络得到以下符章
网络把信息从服务器送到客户端
由文件中的验证码生成
选择包含用户销的符章
把被选中的符章从客户端送到服务器
从网络得到被选中的符章
允许访问
把登录结果传递给客户端
检查客户端符章

图13B


Demo
GATE_5
Delay
Seconds
Clear Demo Id Passwords
Start
GATE_4
GATE_5
Create User Id & Pin
Graphic Login
文字登录
用户名
admin
输入
客户端
网络连接
服务器端
用户名：admin
送往服务器的用户名：admin
用户名[验证码]:admin
从网络得到以下符章
网络把信息从服务器送到客户端
由文件中的验证码生成
选择包含用户销的符章
把被选中的符章从客户端送到服务器
从网络得到被选中的符章
拒绝访问
把登录结果传递给客户端
检查客户端符章

图13C


Demo
GATE_5
Delay
Seconds
Clear Demo Id Passwords
Start
GATE_4
GATE_5
Create User Id & Pin
Graphic Login
文字登录
用户名
admin
G5 606
输入
客户端
网络连接
服务器端
用户名 : admin G5 611
送往服务器的用户名 : admin
用户名[验证码]:admin
从网络得到以下符章
网络把信息从服务器送到客户端
由文件中的验证码生成
选择包含用户销的符章
把被选中的符章从客户端送到服务器
从网络得到被选中的符章
拒绝访问
把登录结果传递给客户端
检查客户端符章

图13D


Demo
GATE_5
Delay
Seconds
Clear Demo Id Passwords
Start
GATE_4
GATE_5
Create User Id & Pin
Graphic Login
Text Login
加密
清晰的信息
secret G4 700
发送器验证码
加密
[A] GATE系统可用于加密，这是使用GATE_4加密信息的一个例子。
信息中的字符与其他填充字符混在一起，以隐藏信息。
每个字符附有一个代表锁的，有16个符章的，表格。每个字符也附有一个由多个符章组成的钥匙。
每个字符的表格和钥匙是由发送者的验证码生成的。
原始信息的有效字符附带有效的钥匙，填充字符附带无效的钥匙。
[B]如下所示，GATE_4被用来解密信息。
从发送器接收所有字符后，检查每个字符，看它附带的钥匙对附加的，有16个符章的，表格是否有效。
认证过程使用接收器验证码。接收器验证码应和发送器验证码相同，才能恢复原始信息。
每个未经认证的字符都不是原始信息的一部分，将被省略。每个被验证的字符都是原始信息的一部分，将被保留，并一起揭示原始信息。
钥匙符章
检查
当前位置
加密过的信息
发出/收到
解密后的信息
接收器验证码
解密

图14

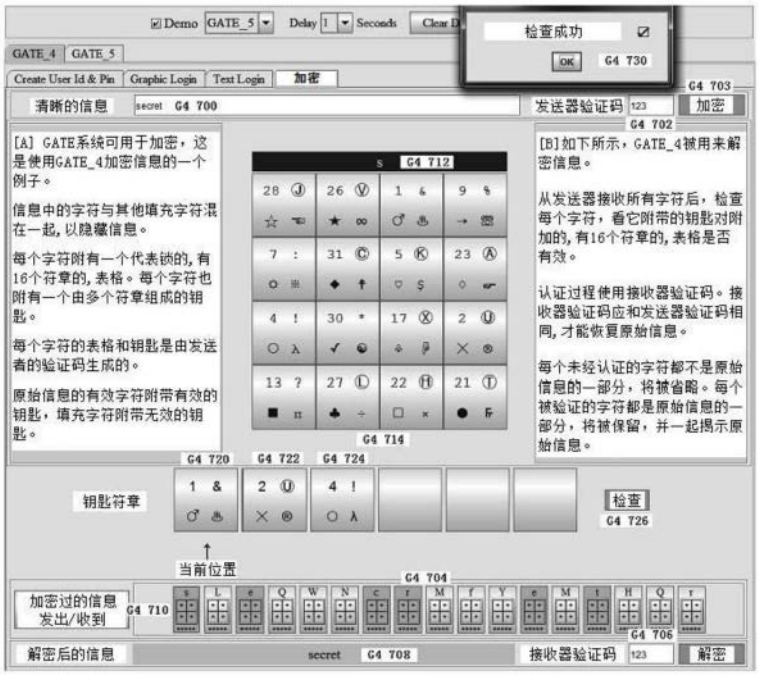
Demo
GATE_5
Delay 1 Seconds
检查成功
OK
G4 730
GATE_4
GATE_5
Create User Id & Pin
Graphic Login
Text Login
加密
G4 703
清晰的信息
secret G4 700
发送器验证码 123
加密
G4 702
[A] GATE系统可用于加密，这是使用GATE_4加密信息的一个例子。
信息中的字符与其他填充字符混在一起，以隐藏信息。
每个字符附有一个代表锁的，有16个符章的，表格。每个字符也附有一个由多个符章组成的钥匙。
每个字符的表格和钥匙是由发送者的验证码生成的。
原始信息的有效字符附带有效的钥匙，填充字符附带无效的钥匙。
G4 712
G4 714
[B]如下所示，GATE_4被用来解密信息。
从发送器接收所有字符后，检查每个字符，看它附带的钥匙对附加的，有16个符章的，表格是否有效。
认证过程使用接收器验证码。接收器验证码应和发送器验证码相同，才能恢复原始信息。
每个未经认证的字符都不是原始信息的一部分，将被省略。每个被验证的字符都是原始信息的一部分，将被保留，并一起揭示原始信息。
G4 720
G4 722
G4 724
钥匙符章
检查
G4 726
当前位置
G4 704
加密过的信息 发出/收到
G4 710
G4 706
解密后的信息
secret G4 708
接收器验证码 123
解密

图15A

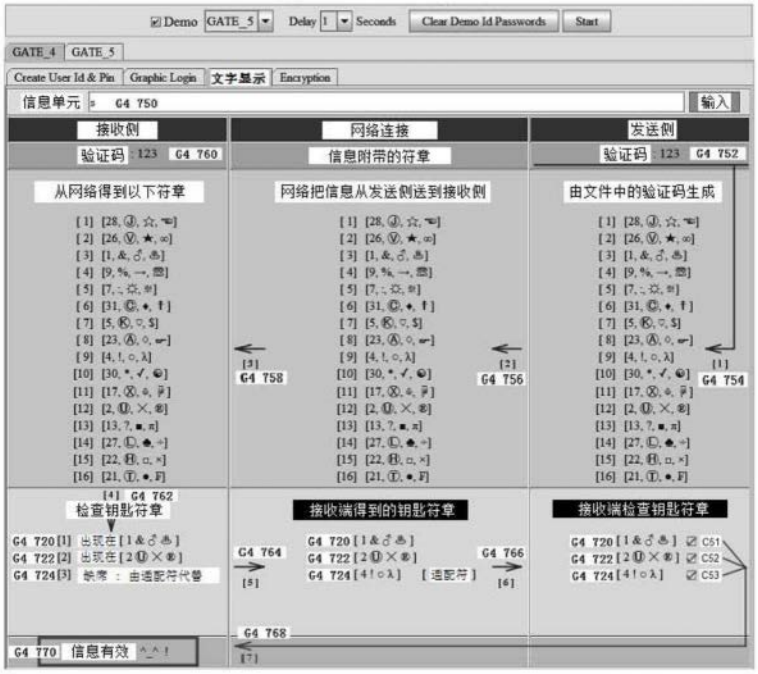
Demo
GATE_5
Delay 1 Seconds
Clear Demo Id Passwords
Start
GATE_4
GATE_5
Create User Id & Pin
Graphic Login
文字显示
Encryption
信息单元 G4 750
输入
接收侧
网络连接
发送侧
验证码 123 G4 760
信息附带的符章
验证码 123 G4 752
从网络得到以下符章
网络把信息从发送侧送到接收侧
由文件中的验证码生成
G4 758
G4 756
G4 754
G4 762
检查钥匙符章
接收端得到的钥匙符章
接收端检查钥匙符章
G4 764
G4 766
G4 768
G4 770 信息有效 ^_^ !

图15B

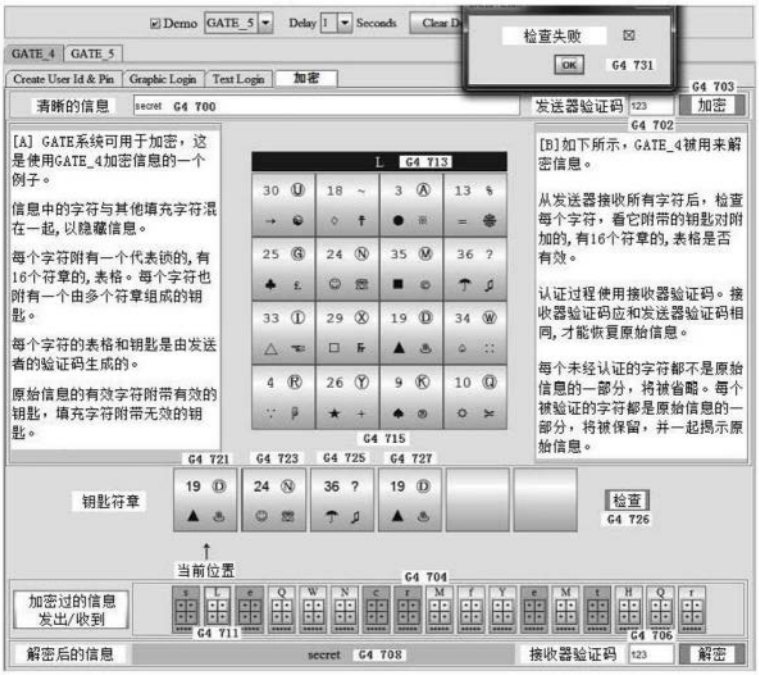
Demo
GATE_5
Delay 1 Seconds
检查失败
OK
G4 731
GATE_4
GATE_5
Create User Id & Pin
Graphic Login
Text Login
加密
清晰的信息
secret G4 700
发送器验证码 123
加密
G4 703
G4 702
[A] GATE系统可用于加密，这是使用GATE_4加密信息的一个例子。
信息中的字符与其他填充字符混在一起，以隐藏信息。
每个字符附有一个代表锁的，有16个符章的，表格。每个字符也附有一个由多个符章组成的钥匙。
每个字符的表格和钥匙是由发送者的验证码生成的。
原始信息的有效字符附带有效的钥匙，填充字符附带无效的钥匙。
L G4 713
G4 715
[B]如下所示，GATE_4被用来解密信息。
从发送器接收所有字符后，检查每个字符，看它附带的钥匙对附加的，有16个符章的，表格是否有效。
认证过程使用接收器验证码。接收器验证码应和发送器验证码相同，才能恢复原始信息。
每个未经认证的字符都不是原始信息的一部分，将被省略。每个被验证的字符都是原始信息的一部分，将被保留，并一起揭示原始信息。
G4 721
G4 723
G4 725
G4 727
钥匙符章
检查
G4 726
当前位置
G4 704
加密过的信息
发出/收到
G4 711
G4 706
解密后的信息
secret G4 708
接收器验证码 123
解密

图16A

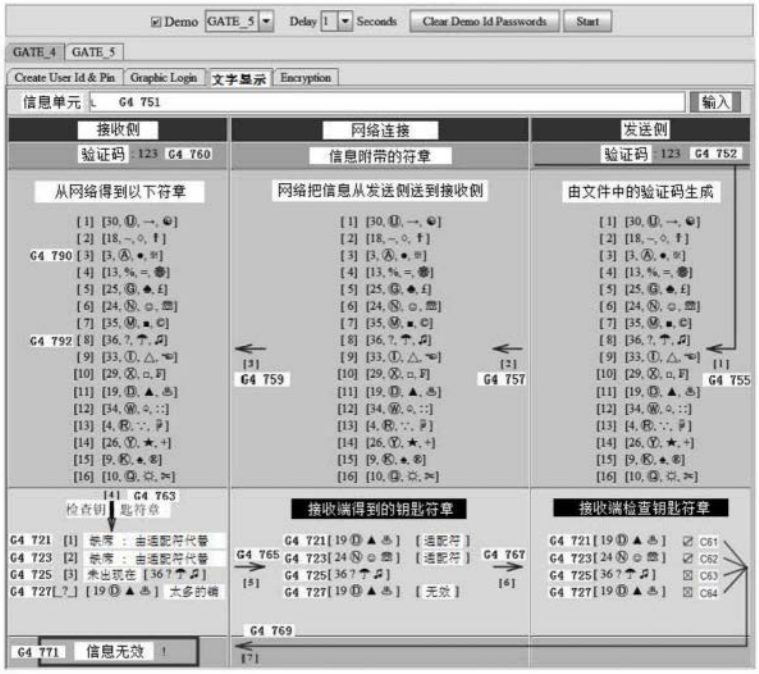
Demo
GATE_5
Delay 1 Seconds
Clear Demo Id Passwords
Start
GATE_4
GATE_5
Create User Id & Pin
Graphic Login
文字显示
Encryption
信息单元 G4 751
输入
接收侧
验证码 123 G4 760
网络连接
信息附带的符章
发送侧
验证码 123 G4 752
从网络得到以下符章
网络把信息从发送侧送到接收侧
由文件中的验证码生成
G4 790
G4 792
G4 759
G4 757
G4 755
检查钥匙符章
接收端得到的钥匙符章
接收端检查钥匙符章
G4 765
G4 767
G4 769
G4 771
信息无效

图16B

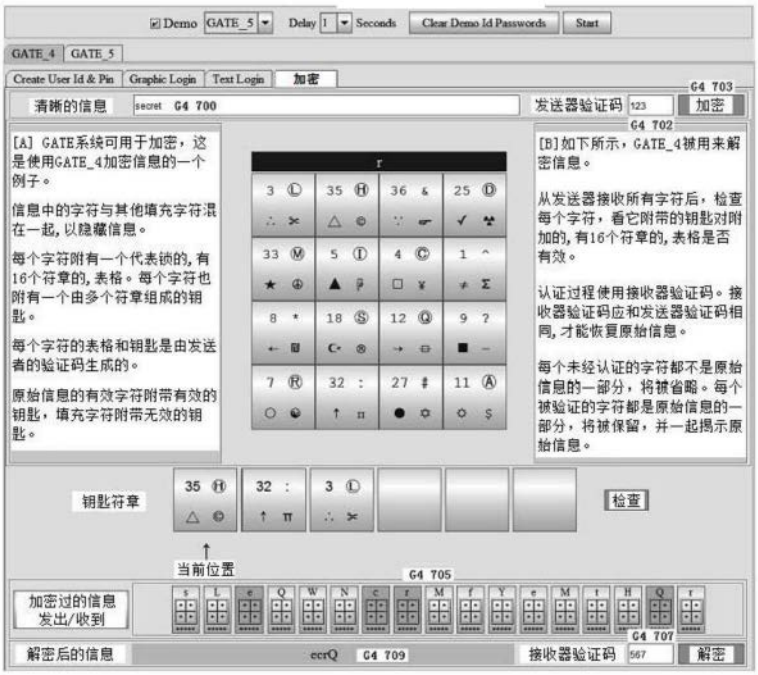
Demo
GATE_5
Delay
1
Seconds
Clear Demo Id Passwords
Start
GATE_4
GATE_5
Create User Id & Pin
Graphic Login
Text Login
加密
清晰的信息
secret
G4 700
发送器验证码
123
加密
G4 703
G4 702
[A] GATE系统可用于加密，这是使用GATE_4加密信息的一个例子。
信息中的字符与其他填充字符混在一起，以隐藏信息。
每个字符附有一个代表锁的，有16个符章的，表格。每个字符也附有一个由多个符章组成的钥匙。
每个字符的表格和钥匙是由发送者的验证码生成的。
原始信息的有效字符附带有效的钥匙，填充字符附带无效的钥匙。
[B]如下所示，GATE_4被用来解密信息。
从发送器接收所有字符后，检查每个字符，看它附带的钥匙对附加的，有16个符章的，表格是否有效。
认证过程使用接收器验证码。接收器验证码应和发送器验证码相同，才能恢复原始信息。
每个未经认证的字符都不是原始信息的一部分，将被省略。每个被验证的字符都是原始信息的一部分，将被保留，并一起揭示原始信息。
钥匙符章
检查
当前位置
G4 705
加密过的信息
发出/收到
G4 707
解密后的信息
ecrQ
G4 709
接收器验证码
567
解密

图17


Demo
GATE_5
Delay
1
Seconds
Clear Demo Id Passwords
Start
GATE_4
GATE_5
Create User Id & Pin
Graphic Login
Text Login
加密
G5 702
G5 703
清晰的信息
FYEO
G5 700
发送器验证码
123
加密
[A] GATE系统可用于加密，这是使用GATE_5加密信息的一个例子。
信息中的字符与其他填充字符混在一起，以隐藏信息。
每个字符附有一个代表锁的，有16个符章的，表格。每个字符也附有一个由多个符章组成的钥匙。
每个字符的表格和钥匙是由发送者的验证码生成的。
原始信息的有效字符附带有效的钥匙，填充字符附带无效的钥匙。
[B]如下所示，GATE_5被用来解密信息。
从发送器接收所有字符后，检查每个字符，看它附带的钥匙对附加的，有16个符章的，表格是否有效。
认证过程使用接收器验证码。接收器验证码应和发送器验证码相同，才能恢复原始信息。
每个未经认证的字符都不是原始信息的一部分，将被省略。每个被验证的字符都是原始信息的一部分，将被保留，并一起揭示原始信息。
钥匙符章
检查
当前位置
G5 704
加密过的信息
发出/收到
G5 706
解密后的信息
接收器验证码
123
解密

图18


Demo
GATE_5
Delay
Seconds
检查成功
OK
G5 730
GATE_4
GATE_5
Create User Id & Pin
Graphic Login
Text Login
加密
清晰的信息
FYEO G5 700
发送器验证码
加密
G5 703
G5 702
[A] GATE系统可用于加密，这是使用GATE_5加密信息的一个例子。
信息中的字符与其他填充字符混在一起，以隐藏信息。
每个字符附有一个代表锁的，有16个符章的，表格。每个字符也附有一个由多个符章组成的钥匙。
每个字符的表格和钥匙是由发送者的验证码生成的。
原始信息的有效字符附带有效的钥匙，填充字符附带无效的钥匙。
F G5 712
G5 714
[B]如下所示，GATE_5被用来解密信息。
从发送器接收所有字符后，检查每个字符，看它附带的钥匙对附加的，有16个符章的，表格是否有效。
认证过程使用接收器验证码。接收器验证码应和发送器验证码相同，才能恢复原始信息。
每个未经认证的字符都不是原始信息的一部分，将被省略。每个被验证的字符都是原始信息的一部分，将被保留，并一起揭示原始信息。
钥匙符章
G5 720
G5 722
G5 724
检查
G5 726
当前位置
G5 704
加密过的信息
发出/收到
G5 710
G5 706
解密后的信息
FYEO G5 708
接收器验证码
解密

图19A


Demo
GATE_5
Delay
Seconds
Clear Demo Id Passwords
Start
GATE_4
GATE_5
Create User Id & Pin
Graphic Login
文字显示
Encryption
信息单元 F G5 750
输入
接收侧
验证码 123 G5 760
网络连接
信息附带的符章
发送侧
验证码 123 G5 752
从网络得到以下符章
网络把信息从发送侧送到接收侧
由文件中的验证码生成
[3] G5 758
[2] G5 756
[1] G5 754
[4] G5 762
检查钥匙符章
接收端得到的钥匙符章
接收端检查钥匙符章
G5 764
G5 766
G5 768
G5 770 信息有效 ^_^ !

图19B

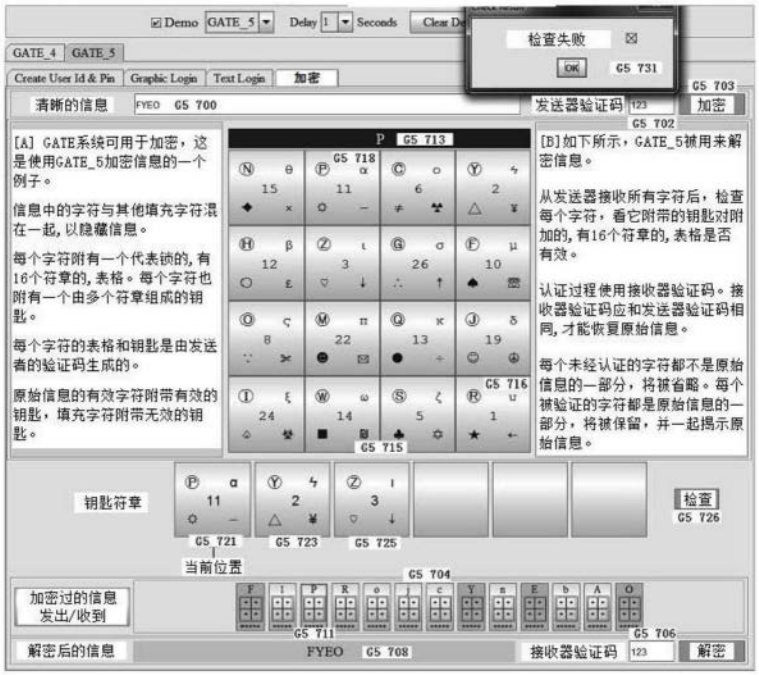
Demo
GATE_5
Delay
Seconds
检查失败
OK
G5 731
GATE_4
GATE_5
Create User Id & Pin
Graphic Login
Text Login
加密
清晰的信息
FYEO G5 700
发送器验证码
加密
G5 703
G5 702
[A] GATE系统可用于加密，这是使用GATE_5加密信息的一个例子。
信息中的字符与其他填充字符混在一起，以隐藏信息。
每个字符附有一个代表锁的，有16个符章的，表格。每个字符也附有一个由多个符章组成的钥匙。
每个字符的表格和钥匙是由发送者的验证码生成的。
原始信息的有效字符附带有效的钥匙，填充字符附带无效的钥匙。
P G5 713
G5 718
G5 716
G5 715
[B]如下所示，GATE_5被用来解密信息。
从发送器接收所有字符后，检查每个字符，看它附带的钥匙对附加的，有16个符章的，表格是否有效。
认证过程使用接收器验证码。接收器验证码应和发送器验证码相同，才能恢复原始信息。
每个未经认证的字符都不是原始信息的一部分，将被省略。每个被验证的字符都是原始信息的一部分，将被保留，并一起揭示原始信息。
钥匙符章
G5 721
G5 723
G5 725
检查
G5 726
当前位置
G5 704
加密过的信息
发出/收到
G5 711
G5 706
解密后的信息
FYEO G5 708
接收器验证码
解密

图20A

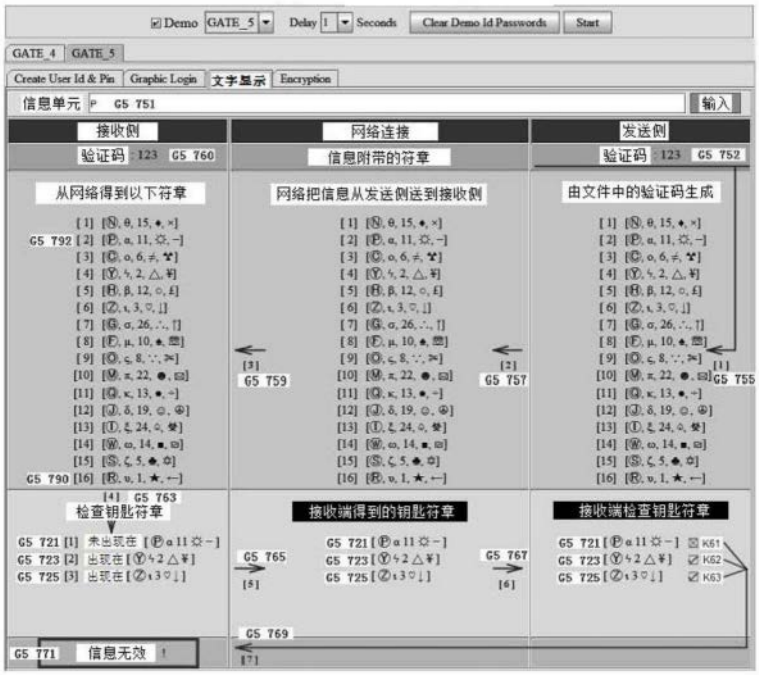
Demo
GATE_5
Delay
Seconds
Clear Demo Id Passwords
Start
GATE_4
GATE_5
Create User Id & Pin
Graphic Login
文字显示
Encryption
信息单元
G5 751
输入
接收侧
网络连接
发送侧
验证码 123 G5 760
信息附带的符章
验证码 123 G5 752
从网络得到以下符章
网络把信息从发送侧送到接收侧
由文件中的验证码生成
G5 792
G5 790
G5 759
G5 757
G5 755
检查钥匙符章
接收端得到的钥匙符章
接收端检查钥匙符章
G5 721 [1] 未出现在
G5 723 [2] 出现在
G5 725 [3] 出现在
G5 765
G5 767
G5 769
G5 771
信息无效
K61
K62
K63

图20B

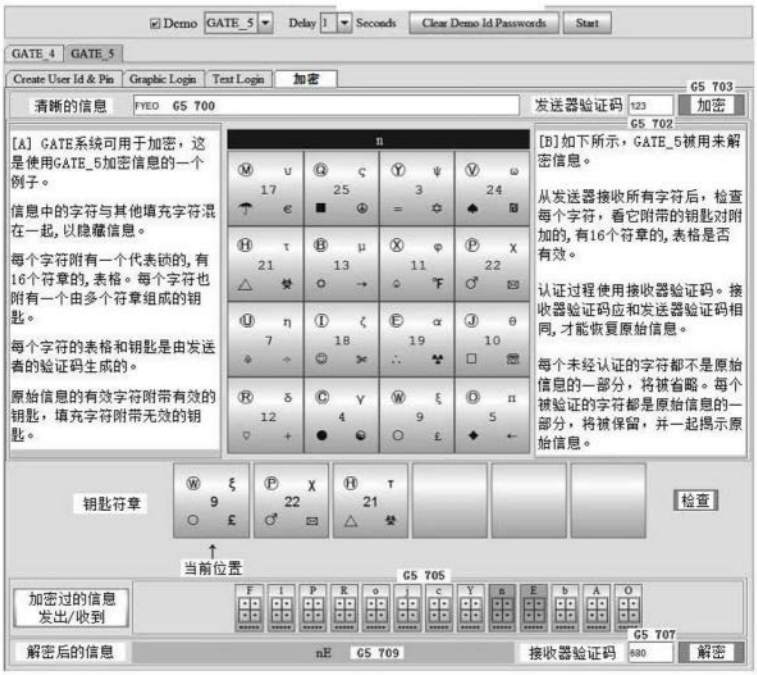
Demo
GATE_5
Delay
1
Seconds
Clear Demo Id Passwords
Start
GATE_4
GATE_5
Create User Id & Pin
Graphic Login
Text Login
加密
清晰的信息
G5 700
发送器验证码
123
加密
G5 703
G5 702
[A] GATE系统可用于加密，这是使用GATE_5加密信息的一个例子。
信息中的字符与其他填充字符混在一起，以隐藏信息。
每个字符附有一个代表锁的，有16个符章的，表格。每个字符也附有一个由多个符章组成的钥匙。
每个字符的表格和钥匙是由发送者的验证码生成的。
原始信息的有效字符附带有效的钥匙，填充字符附带无效的钥匙。
[B]如下所示，GATE_5被用来解密信息。
从发送器接收所有字符后，检查每个字符，看它附带的钥匙对附加的，有16个符章的，表格是否有效。
认证过程使用接收器验证码。接收器验证码应和发送器验证码相同，才能恢复原始信息。
每个未经认证的字符都不是原始信息的一部分，将被省略。每个被验证的字符都是原始信息的一部分，将被保留，并一起揭示原始信息。
钥匙符章
检查
当前位置
G5 705
加密过的信息
发出/收到
G5 707
解密后的信息
G5 709
接收器验证码
680
解密

图21