过单向函数计算n次得到的口令为：

$$P_1 = f(f(f(f(s))))$$

第2个口令用单向函数运算$n-1$次：

$$P_2 = f(f(f(s)))$$

第3个口令对f运算2次，第4个运算1次。总之，$P_{i-1} = f(P_i)$。要注意的地方是，给定任何序列里的口令，我们很容易计算出口令序列里的前一个值，但却不可能计算出后一个值。如，给定P_2很容易计算出P_1，但不可能计算出P_3。

口令服务器首先由P_0进行初始化，即$f(P_1)$。这一值连同登录用户名和整数1被存放在口令文件的相应条目里。整数1表示下一个所需的口令是P_1。当用户第一次登录时，他首先把自己的登录名发送到服务器，服务器回复口令文件里的整数值1。用户机器在本地对所输入的s进行运算得到P_1。随后服务器根据P_1计算出$f(P_1)$，并将结果同口令文件里的(P_0)进行比较。如果符合，登录被允许。这时，整数被增加到2，在口令文件中P_1覆盖了P_0。

下一次登录时，服务器把整数2发送到用户计算机，用户机器计算出P_2。然后服务器计算$f(P_2)$的值并将其与口令文件中存放的值进行比较。如果两者匹配，就允许登录。这时整数n被增加到3，口令文件中由P_2覆盖P_1。这一机制的特性保证了即使入侵者可以窃取P_i也无法从P_i计算出P_{i+1}，而只能计算出P_{i-1}，但P_{i-1}已经使用过，现在失效了。当所有n个口令都被用完时，服务器会重新初始化一个密钥。

4. 挑战—响应认证

另一种口令机制是让每一个用户提供一长串问题并把它们安全地放在服务器中（如可以用加密形式）。问题是用户自选的并且不用写在纸上。下面是用户可能选择的问题：

1) 谁是Marjolein的姐妹？

2) 你的小学在哪一条路上？

3) Woroboff女士教什么课？

在登录时，服务器随机提问并验证答案。要使这种方法有效，就要提供尽可能多的问题和答案。

另一种方法叫做挑战—响应。使用这种方法时，在登录为用户时用户选择某一种运算，例如x^2。当用户登录时，服务器发送给用户一个参数，假设是7，在这种情形下，用户就输入49。这种运算方法可以每周、每天后者从早到晚经常变化。

如果用户的终端设备具有十分强大的运算能力，如个人计算机、个人数字助理或手机，那么就可以使用更强大的挑战响应方法。过程如下：用户事先选择密钥k，并手工放置到服务器中。密钥的备份也被安全地存放在用户的计算机里。在登录时，服务器把随机产生的数r发送到用户端，由用户端计算出$f(r, k)$的值。其中，f是一个公开已知的函数。然后，服务器也做同样的运算看看结果是否一致。这种方法的优点是即使窃听者看到并记录下双方通信的信息，也对他毫无用处。当然，函数f需要足够复杂，以保证k不能被逆推。加密散列函数是不错的选择，r与k的异或值（XOR）作为该函数的一个参数。迄今为止，这样的函数仍然被认为是难以逆推的。

9.4.2 使用实际物体的认证方式

用户认证的第二种方式验证一些用户所拥有的实际物体而不是用户所知道的信息。如金属钥匙就被使用了好几个世纪。现在，人们经常使用磁卡，并把它放入与终端或计算机相连的读卡器中。而且一般情况下，用户不仅要插卡，还要输入口令以保护别人冒用遗失或偷来的磁卡。银行的ATM机（自动取款机）就采用这种方法让客户使用磁卡和口令码（现在大多数国家用4位的PIN代码，这主要是为了减少ATM机安装计算机键盘的费用）通过远程终端（ATM机）登录到银行的主机上。

载有信息的磁卡有两种：磁条卡和芯片卡。磁条卡后面粘附的磁条上可以写入存放140个字节的信息。这些信息可以被终端读出并发送到主机。一般这些信息包括用户口令（如PIN代码）这样终端即便在与银行主机通信断开的情况下也可以校验。通常，用只有银行已知的密钥对口令进行加密。这些卡片每张成本大约在0.1美元到0.5美元之间，价格差异主要取决于卡片前面的全息图像和生产量。在鉴别用户方面，磁条卡有一定的风险。因为读写卡的设备比较便宜并被大量使用着。