一、USBKey 实现身份认证原理

采用冲击/响应（挑战/应答）的认证方法，登录时在服务器端和客户端同时进行计算，客户端计算前要先验证 USER PIN，通过后在硬件中使用 HMAC-MD5 密钥进行计算，服务器端在服务器上使用软件进行计算，比较计算结果。

二、USBKey 的优点

1、兼容性好

USBKey 不仅对打印机、扫描仪等设备具有高度的透明性，特别是多个相同的 USBKey 也可以使用 USB HUB 并联在一起使用，相互之间不会干扰。

2、速度快

对于使用 USBKey 加密后的软件，其运行速度同加密前区别不大，USBKey 能够在很短的时间内处理完毕，保证用户程序的顺畅运行。

3、使用简便

USBKey 在 API 函数调用上从用户角度出发，最大限度简化使用接口。用户能够在很短的时间内掌握 USBKey 的使用方法，节约开发上所投入的时间。

三、高加密强度和身份认证相结合

USBKey 是全新设计的高强度 USBKey，有完整的用户管理。

（1）用户必须在超级用户状态下（SO PIN 验证通过），通过自己设定的不超过 51 字节的种子生成 PID，以后打开和关闭 USBKey 都需要通过 PID 来完成。PID 的生成算法是在 USBKey 内部完成的，而且是不可逆的，也就是说，只有生成者才知道什么样的种子能生成什么样的 PID，别的人即使知道 PID，同时也能够调用这个计算过程，但因为不知道种子是什么，是无法生成和您相同的 PID 的硬件，保证了用户的 USBKey 的独特性。

（2）用户在对 USBKey 中的数据进行读写操作时需要进行 USER PIN 验证，又增加了一层对软件的保护性。

（3）用户可以在配置设备时设为只读，那么 USBKey 中的数据只可以读取，而不能被更改，密钥也不能被修改，从而保证了锁内数据不被篡改。

（4）使用 USBKey 硬件中的 HMAC-MD5 算法进行冲击响应身份认证。HMAC-MD5 密钥存在 USBKey 中，该密钥只用于计算，任何人获取不到密钥的内容，保证密钥的安全性。

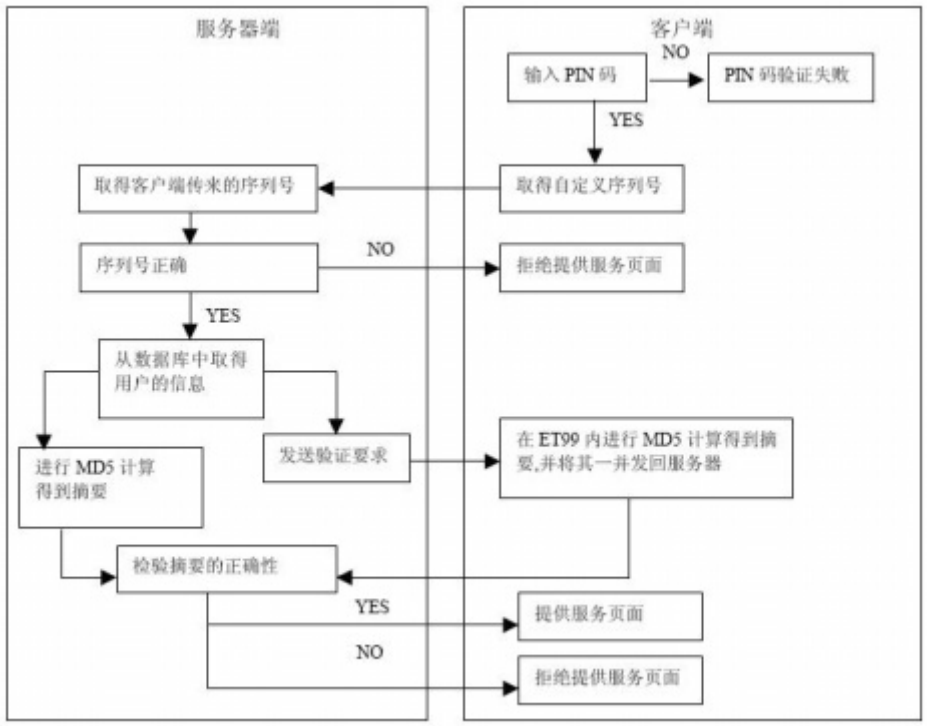
（5）提供了安全方便的外壳加密工具，使加密工作非常简单。

四、系统支持

USBKey 采用无驱设计，使用方便，兼容性好，支持多种操作系统台：全系列 Windows、Linux 和 MAC。包括 32 位和 64 位。

五、使用 USBKey 进行身份认证

可以应用 USBKey 进行冲击响应身份认证，替换掉传统的用户名和密码方式，使登录更加安全。其原理如下图所示：



在整个认证过程中，采用冲击响应的认证方式。当需要在网络上验证用户身份时，先由客户端向服务器发出一个验证请求。服务器接到此请求后生成一个随机数并通过网络传输给客户端（此为冲击）。客户端将收到的随机数提供给，由使用该随机数与存储在中的密钥进行 HMAC-MD5 运算并得到一个结果作为认证证据传给服务器（此为响应）。与此同时，服务器也使用该随机数与存储在服务器数据库中的该客户密钥进行 HMAC-MD5 运算，如果服务器的运算结果与客户端传回的响应结果相同，则认为客户端是一个合法用户。

六、原理详解

冲击/响应的算法原理可以参见（http://www.rfc-editor.org/rfc/rfc2085.txt）。其中公式原型为：

MD5(K XOR opad, MD5(K XOR ipad, text))

K：密钥（即分配给不同用户的密钥）

text：随机数
opad：数值 0x5C
ipad：数值 0x36
XOR：异或运算符

ET99 USBKey 在此基础上进行了改进，结合进硬件的特点，使得存储在 USBKey 中的密钥 K 也是不可被盗取和得知的。ET99 的公式为：

MD5(MD5(K XOR opad), MD5(MD5(K XOR ipad), text))

我们看到 ET99 的计算公式中，将 K XOR opad 和 K XOR ipad 的结果又分别做了 MD5 运算。假如分配给客户独有的密钥 K 为 123456，那么 K 与 opad 异或的结果再进行 MD5 运算后的 16 字节称为 K1，K 与 ipad 异或的结果再进行 MD5 运算后的 16 字节称为 K2，K1＋K2 拼接后的 32 字节 Kin99 作为存入 ET99 USBKey 中密钥区中的密钥（ET99 可以存储 8 个 32 字节密钥）。这样做的目的有两点：（1）硬件保证 32 字节的密钥 Kin99 是无法被导出的。（2）即使通过物理探针分析芯片，得到这 32 字节，通过拆分得到 K1 和 K2，但由于 MD5 是成熟的单向散列算法，通过 K1 和 K2 是反推不出 K 的值，因此绝对保证了分配给用户，代表用户身份的 K 是无法被复制的。

那么从公式中就可以一目了然看到整个冲击/响应客户端和服务器端进行计算的过程。

ET99 中计算：
（1） 使用 K2＋text（随机数）拼接后进行 MD5 运算，产生 16 字节的结果 Rc1。
（2） 使用 K1＋Rc1 拼接后进行 MD5 运算，产生客户端计算的结果 Rc。

服务器端计算：
（1） 从数据库中取得客户的密钥 K。
（2） K 与 ipad 异或后进行 MD5 运算产生 16 字节的 Rs1。
（3） Rs1＋text（随机数）拼接后进行 MD5 运算，产生 16 字节 Rs2。
（4） K 与 opad 异或后进行 MD5 运算产生 16 字节的 Rs3。
（5） Rs3＋Rs2 拼接后进行 MD5 运算产生 16 字节的服务器端计算结果 Rs。

我们看客户端的 ET99 USBKey 中和服务器端的计算过程都是使用相同的算法，相同的过程，因此计算结果 Rc 和 Rs 是一致的。K1 和 K2 也是由 K 产生的。同时有随机数的参与，因此每次的计算结果是变化的，使用一次即作废，保证了身份认证的安全。

七、安全性

在整个认证过程中网络上所传递的只有 3 种数据：用户名，随机数和计算结果，计算结果由随机数的不同而每次各不相同，这些数据被截取到也是没有意义的。同时 ET99USBKey 是客户端的安全产品，具有以下特点：

- 登录用户必须先输入自己的 USER PIN 进行验证后才有权限完成计算。
- USER PIN 有最大重试次数限制，连续输入错误会锁死。从而防止硬件丢失后，被不合法的用户反复重试。

- 存储在 ET99USBKey 中的密钥不能被任何人获取。
- 用户登录时必须具备硬件和保护硬件的 USER PIN 双重因子时才能登录。有硬件，不知道 USER PIN 或者知道 USER PIN，没有硬件，都是没有办法登录的。比传统的用户名和密码方式大大增加的登录用户的安全性。
- 保障了系统开发商的利益。使用硬件登录，不存在用户名密码共享的问题。