基于 MAC 地址的软件动态口令实现方案①

赖叶蕾[1], 叶 晰[1,2], 叶依如[1]

[1](温州医科大学 信息与工程学院, 温州 325035)
[2](中山大学 软件学院, 广州 510275)

摘 要: 随着网络技术的发展, 传统的静态密码身份认证方案已不能给电子商务活动提供足够的保护. 描述了动态口令技术的原理, 分析了硬件和软件实现动态口令的利弊, 设计了一种基于客户端 MAC 地址的动态口令认证协议, 并在此基础上论述了系统方案的实施流程、总体设计和认证过程. 最后进行了安全性分析并提出了相应的提高安全性的措施. 分析表明, 该方案具有适用面广、安全性高、使用方便和系统成本低的特点.

关键词: 动态口令; 身份认证; 软件口令牌; MAC 地址

Implementation of Dynamic Identity Authentication System Based on MAC Address

LAI Ye-Lei[1], YE Xi[1,2], YE Yi-Ru[1]

[1](School of Information and Engineering, Wenzhou Medical University, Wenzhou 325035, China)
[2](School of Software, Sun Yat-Sen University, Guangzhou 510275, China)

Abstract: With the development of Internet technology, the traditional static password based on authentication solutions is no longer an adequate protection scheme to serious enterprise applications. In this article, the principle of dynamic password technology is described and the pros and cons of implementing dynamic password technology by software and hardware are analyzed as well. A dynamic password authentication protocol based on MAC Address of client is designed. According to this protocol, the system architecture, authentication processes and safety measures are described and its security is analyzed as well. The analysis indicates that this system features high security and wide application. It can be conveniently used and implemented in low cost.

Key words: dynamic password; identity authentication; software token; MAC address

随着电子商务活动的盛行, 越来越多的人习惯了网上购物和网上支付等生活方式. 网上购物和网上支付等活动必然涉及到用户的身份认证问题. 一般来说, 确定用户的身份可通过用户知道什么(如传统意义上的静态密码)和用户拥有什么[1](如口令牌、U 盾或者指纹和虹膜等生物特征). 随着网络和黑客技术的发展, 用户名/密码方式认证已经被证明是不安全的[2], 而生物特征认证方式虽然安全程度高, 但由于认证终端的价格不菲(如苹果公司最新推出的首款支持指纹识别支付的 iPhone5S 售价为 5288 元), 所以即使在外国该技术也仅仅处于刚开始推广的阶段. 目前, 国内金融机构等需要严格身份认证的部门普遍采用 IC 卡认证(如中国银行的电子口令牌)和 USB Key 认证(如工商银行的 U 盾). 无论是动态口令技术还是 USB Key 技术, 都有各自的优缺点. USB Key 使用签名技术, 安全性最高, 没有 USB Key 设备接入客户端则无法实现转账等操作, 故使用起来灵活度不够. 而动态口令的使用相对灵活, 认证时只需知道口令即可(即无需认证设备接入), 缺点是存在被不法分子骗取的口令的风险. 而动态口令的实现又可分为用硬件(电子口令牌)或者软件(直接运行在 PC 端的软件口令牌)来实现. 电子口令牌实现的身份认证技术其安全性相对较高, 但是其成本也很高, 不适合中小型电子商务网站的应用, 而软件实现身份认证技术的安全性虽不如硬件口令牌, 但

① 基金项目:浙江省大学生科技创新计划(新苗人才计划)(2013R413034);广东省自然科学基金(10151027501000061)
收稿时间:2013-12-17;收到修改稿时间:2014-01-15

其实现方便, 成本低, 只需在原认证方案上增加一个模块, 无须而外购置设备和发行电子口令牌. 本文就是提出了一种基于客户端机器的 MAC 地址的软件口令牌认证方案. 该软件口令牌运行时读取客户端的 MAC 地址作为加密因素之一, 由于 MAC 地址的全球唯一性, 这也体现了该动态口令的唯一性.

1 动态口令的基本原理和技术模式

动态口令又称为一次性口令, 其特点是用户根据服务方提供的动态口令牌上显示的数字来输入动态口令, 并且每个登录口令只使用一次, 攻击者无法用窃听到的口令来做下一次登录, 同时利用单向散列(HASH)函数的不可逆性, 阻止了攻击者从窃听到的当次口令推出下一次登录口令[3]. 根据动态口令产生时不确定因素的选择方式, 动态口令可分为: 事件同步机制、时间同步机制和挑战/应答机制[4]. 由于基于时间同步的令牌技术实现相对容易, 且用户使用时无需额外输入, 用户满意度较高, 故在本方案中我们采用基于时间同步的令牌技术.

2 动态口令算法设计和实现

2.1 生成动态口令的算法

动态口令的安全性主要由单向散列函数计算上的不可逆性来保证. 常用的单向散列函数包括: MD4, MD5, SHA-1, SHA-256 和 SHA-512 等. 由于 MD4 和 MD5 的安全性已经受到人们的质疑, 而 SHA-256 和 SHA-512 的计算量偏大, 故最终本项目选择了安全性较高且计算量不大的 SHA-1 作为产生动态口令的单向散列函数.

动态口令的产生和验证过程如图 1 所示. 为了方便描述, 对图中采用的符号做如下定义: S 为认证服务器, A 为用户; KA1 为 A 用户密钥 1; KA2 为 A 用户密钥 2; PA 为动态口令.

客户端软件中保存有该用户的密钥信息, 用户密钥 KA1 和 KA2 是在用户注册时产生的两个随机整数, 并保存在认证服务器端的用户注册信息表中. 客户端软件中的时钟计数器每隔 1 秒(类似于 Timer 事件)自动使用密钥同时加密 MAC 地址和当前时间 t(如 2013-12-14 23:20, 精确到分钟), 然后用 Sha-1 算法对加密结果进行单向散列计算(第二层加密), 产生两个 40 位的十六进制摘要. 然后再把这两个摘要分别平均分割为 8 段并转换为十进制, 最后把两组 8 段的十进制数分别进行异或运算并取每段最右边位得到最终的 8 位动态口令(如 “4 6 6 0 8 7 6 5”). 由于被加密因素中的时间精确到分钟, 故最终产生的动态口令也是每隔 60 秒变换一次.

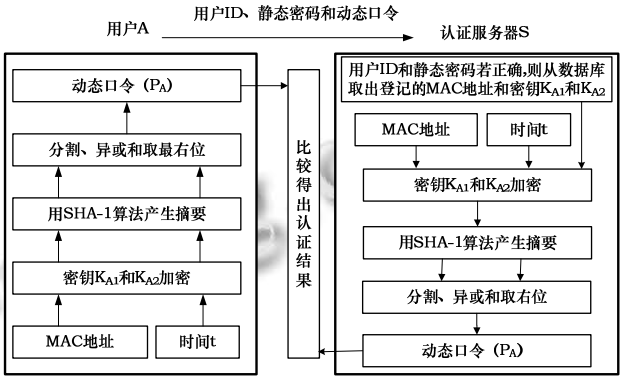

图 1 动态口令的产生和验证流程图

2.2 动态口令的验证

服务器收到用户输入的 UserID、静态密码和动态口令后, 先验证 UserID 和静态密码是否正确, 如正确则从数据库中读取用户登记的 MAC 地址、密钥 KA1 和密钥 KA2. 使用密钥对 MAC 地址和当前时间 t 进行同客户端相同的处理, 最终产生服务器端的动态口令, 并同客户端输入的动态口令进行比较, 一致则验证通过.

2.3 加密过程

本系统方案中使用密钥 KA1 和 KA2 对用户 MAC 地址和当前时间 t 进行第一层加密, 我们为此写了一个加密函数, 其代码(C#语言)如下:

```
static string Encrypt(String StrSource, Byte Key1, Byte Key2)
 {
      foreach (char c in StrSource)   //从待加密字符串
                                        中取出一个字符
    {
       i++;   ASC = (Byte)c;
       if (i % 2 == 1)
          {
            //取奇数位字符和 Key1 进行异或
            BLowData = (byte)(ASC ^ Key1);
            //将运算后数据合成新的字符
```

```
            StrTemp += Convert.ToString(BLowData);
             }
            else
             {
               //取偶数位字符和 Key2 进行异或
               BHighData = (byte)(ASC ^ Key2);
               //将运算后数据合成新的字符
            StrTemp += Convert.ToString(BHighData);
            }
        }
        return StrTemp;
    }
```

第二层加密使用 SHA-1 加密算法，由于 SHA-1 的理论不可逆性决定了攻击者无法根据密文逆推出明文，这样一来也保证了整体方案的安全性.

3　系统设计和实现

3.1　系统实施流程

本身份认证系统的实施步骤为:

1) 用户在网站上注册新会员，并填写个人信息与客户端机器的 MAC 地址.

2) 网站审核通过用户的信息后，服务器软件产生两个随机整数(0 至 255)作为该用户密钥 KA1 和 KA2，并存放于数据库用户信息表中.

3) 网站生成一个口令生成安装文件(该文件包含有用户的密钥信息)，并发放给用户(可采取网站下载或 Email 附件下载等方式). 当用户进行身份验证时，必须先安装并运行该软件口令牌得到该时段(精确到分钟)的动态口令.

登陆时用户先在网站上输入用户名和静态密码，如果正确则进一步提示输入该时刻的动态口令，动态口令正确则登录成功. 登录界面如图 2, 3 所示.

图 2　静态密码验证界面　　图 3　动态口令验证界面

3.2　服务器端软件的设计和实现

服务器端我们使用了 Windows + IIS + SQL Server 2005 + ASP.net 架构，并使用了目前较流行的 C#进行编程. 主要的文件和对应的页面如图 4 和表 1 所示.

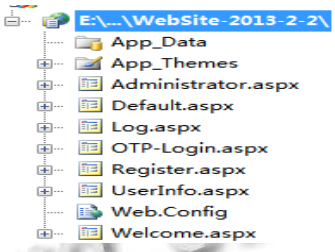

图 4　服务器端软件文件结构图

表 1　文件名和页面对应表

文件名	对应的页面
Administrator.aspx	管理员登录页面
Defualt.aspx	静态密码验证页面
Log.aspx	日志管理页面
OTP-Login.aspx	动态口令验证页面
Register.aspx	用户注册页面
UserInfo.aspx	用户信息管理页面
Welcome.aspx	登录成功欢迎页面

当用户登录系统时，除了输入用户名和静态密码外，还需输入该时刻的动态密码. 如图 5 所示，如果用户名或静态密码错误则提示“错误的用户名或密码”. 如果用户名和静态密码正确，但动态密码输入有误，则提示如图 6 所示的“错误的动态口令”.

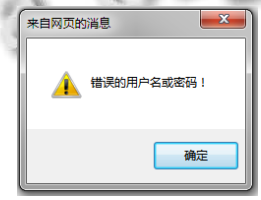

图 5　用户名/静态密码错误提示图

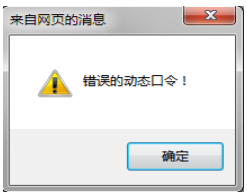

图 6　动态口令错误提示图

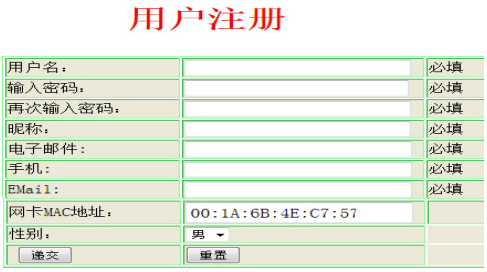

图 7　用户注册界面

用户首次使用网站服务时, 需要先注册, 用户注册界面如图 7 所示. 用户在注册时要输入运行该软件口令牌的计算机的网卡 MAC 地址, 由于每张网卡的MAC地址是全球唯一的, 这也体现了用户加密因素的唯一性.

3.3 PC 端软件口令牌的设计与实现

用户在注册时需要输入本地计算机的网卡 MAC 地址作为被加密因子之一. 一般的来说我们可以运行“ipconfig /all”得到本地网卡的 MAC 地址.

鉴于部分用户可能对命令行操作不太适应或者客户机可能有多个网卡(如遇此种情况, 我们读取第一个 MAC 地址作为注册信息), 故我们特定编写了一个可自动获取本地计算机的网卡 MAC 地址的小程序, 用户在注册时如遇困难可通过 “帮组”按钮进而下载该程序, 其运行界面如图 8 所示. 此外通过点击 “复制”按钮, 用户可以很方便地把 MAC 地址复制到剪贴板中, 这样注册时就无需用户手动输入 MAC 地址.

图 8 获取 MAC 地址的程序运行效果图

用户注册时输入网卡 MAC 地址后, 系统服务器会把该用户的 MAC 地址信息存放在用户信息数据库中, 并按照 2.1 节所描述的算法生成软件口令牌并发放给用户(可通过网页直接下载或电子邮件附件形式发放). 由于软件口令牌运行时会直接读取本地网卡 MAC 地址信息并作为被加密因子之一, 而网卡 MAC 地址是全球唯一的, 故即使黑客窃取了该软件口令牌也无法在非注册机器上得到正确的动态口令. 当然理论上黑客可以伪造本机的 MAC 地址, 以达到生成正确的动态口令的目的, 我们将在第 4 节“安全性分析”进行相关讨论.

用户收到系统发来的安装文件后, 直接点击安装并运行就可以得到当时(精确到分钟)的动态口令. 图 9 为软件口令牌运行效果图.

图 9 软件口令牌运行效果图

4 安全性分析和提高安全性的措施

1)从用户易用性和降低系统复杂度的角度来考虑, 我们也可使用 UserID 和时间作为被加密因子, 这样一来用户注册时无须额外输入 MAC 地址, 提高了用户满意度, 但同时也降低了系统安全性. 因为攻击者可以通过假冒用户进行注册申请, 从而获得服务器发送的用于生成动态口令的可执行文件. 由于该可执行文件中包含了被加密因子(即 UserID), 因此该 UserID 容易被攻击者通过软件跟踪或其他方式予以破译. 而通过读取系统的 MAC 地址这一方式有效阻止了这种攻击方式, 因为在可执行文件中没有被加密因子信息(只包含密钥信息).

2) 黑客在窃取了软件口令牌之后, 可能通过特殊软件修改系统的注册表, 伪造网卡 MAC 的地址, 并以此来达到生成正确动态口令的目的. 通过查找文献, 我们发现通过程序读取本地网卡 MAC 地址的常用方法有如下五种:

① 通过 IPConfig 命令读取 MAC 地址
② 通过 WMI 读取 MAC 地址
③ 通过 Network Interface 读取 MAC 地址
④ 通过 Send ARP 读取 MAC 地址
⑤ 从注册表读取 MAC 地址

而在本项目方案中我们通过 WMI 读取 MAC 地址, 并未从注册表(容易被黑客改写)读取 MAC 地址, 故本方案可以抵御此类攻击.

3)可考虑将保存在服务器端的用户密钥和 MAC 地址等信息用公钥加密算法(如 RSA 等)加密后保存在用户数据库中, 这样就能防止由于黑客对服务器数据库的攻击, 而造成用户 MAC 地址和用户密钥的泄露[5]. 服务器使用用户数据时, 可先用私钥解密再读取明文. 当然这也加重的服务器的负担.

4)如果用户需要, 也可以为软件口令牌设置特定的启动密码, 这样可防止同机操作的其他人员使用该软件.

5 结语

本文阐述了用密钥对客户端 MAC 地址和当前时间进行加密, 然后再用单向散列算法产生摘要, 最终产生动态口令的技术. 该方法在尽量保持硬件实现动态口令优点的同时, 提出了用软件实现动态口令技术的整套方案. 本方案中的被加密因子 MAC 地址由软

件口令牌运行时直接从系统中提取, 由于 MAC 地址本身是全球唯一的, 这也保证了所产生动态口令的唯一性. (当然由于动态口令位数固定, 所以碰撞是在所难免的[6]). 该方案能够满足一般性身份认证系统的需要, 可以直接在现有系统基础上进行开发. 一般只需在现有认证模块之后再添加一个动态口令认证模块即可, 现有信息系统的其他部分不需要任何修改. 此外部署本方案无需任何额外的硬件成本, 故本方案也非常适用于各类中小型商务网站的身份认证系统.

参考文献

1 胡天麟,刘嘉勇,陈芳,隋喆.基于 MD5 的 OTP 认证系统的原理及实现.信息技术,2005,9:140–142.

2 李凤银,刘培玉,鞠宏传.OTP 技术的一种改进方案与应用.计算机系统应用,2003,4:34–36.

3 吴佩萱.基于时间同步机制的动态密码认证系统.长江大学学报(自然版),2005,2(7):256–257.

4 曾伟国,胡汉平,王祖喜,孔涛.基于手机令牌方式的动态身份认证系统.计算机与数字工程,2005,9:21–24.

5 Kim HC, Lee HW, Lee KS, et al. A design of one-time password mechanism using public key infrastructure. Proc. 4th International Conference on Networked Computing and Advanced Information Management, NCM 2008,1. 18–24.

6 叶晰,叶依如.基于 MD5 算法的动态口令技术的软件实现.计算机应用与软件,2009,26(11):281–282.