B/S 模式下一次性口令身份认证方案的设计与分析①

Design and Analysis of an One – time Password Authentication Scheme under B/S Pattern

王庆生 邱鹏飞 (太原理工大学计算机与软件学院 山西 太原 030024)

摘 要: 采用 RSA,AES,MD5 加密算法,运用普通口令和图片口令结合的双口令技术及服务器标识语(server identification)等设计出一种适用于 B/S 架构的一次性口令身份认证系统方案。该方案实现了双向认证,具有效率高,安全可靠,认证原理灵活等特点。

关键词: 身份认证 一次性口令 加密算法 图片口令 服务器标识

1 引言

身份认证中最常用的技术是口令认证技术,而目前普遍采用的口令认证技术是静态口令认证技术。静态口令技术的特点是简单、易用并且也具备一定的安全性,但随着网络应用的复杂化、攻击手段的多样化,静态口令技术的安全缺陷也越来越明显,已经不再适用于安全性要求较高的网络应用系统。针对静态口令技术的安全缺陷而提出的一次性口令认证技术(OTP one time password):在登录过程中加入不确定因素,使用户每次登录系统时传送的口令都不同,以提高系统的安全性。得到该技术得到了迅速地发展。但是,目前现有的一次性口令认证方案并不是十分完善,各种方案在执行性能和安全性上都存在着缺陷,并且大部分都是基于 C/S 应用而提出的。因此随着 Web 应用层次的不断深入(如:电子商务,网上银行,网上购物的发展和普及应用),设计更加完善的适用于 Web 系统的一次性口令认证方案并实现相应的认证系统,让一次性口令认证技术为更多的网络系统提供更加安全可靠的身份认证。这对 Internet 网络应用的发展有着重大的意义和积极的推动作用。

2 口令安全设置和口令易记性难题

口令的安全性(或保密性)和可记忆性一直是人们最关心的问题,也是需要解决的十五个典型的信息安全问题之一。在一个网络系统中,每个网络服务或系统都要求不同的认证方式,用户需要记忆多个口令,据估算,用户平均至少需要四个口令,特别是系统管理员,需要记住的口令就更多,例如开机口令、系统进入口令、数据库口令、邮件口令、Telnet 口令、FTP 口令、路由器口令、交换机口令等。按照安全原则,口令设置既要求复杂,而且口令长度要足够长,但是口令复杂则记不住,因此,用户选择口令只好用简单的、重复使用的口令,以便于保管,这样一来攻击者只要猜测到某个用户的口令,就极有可能引发系列口令泄露事件。

3 B/S 架构下一次性口令设计难点分析

3.1 设计难点分析

基于 B/S 的一次性口令系统比起传统的基于 C/S 的一次性口令系统技术有如下难点:

(1) 实现上:Http 协议是一种无连接、无状态协议,它是通过用户请求/服务器响应得方式工作的,基本上客户的请求都是经过服务器计算处理完成然后发送给用户的,客户端的浏览器,基本上只是做一个交互界面

① 山西省基金编号:2008011028 – 1

© 中国科学院软件研究所 http://www.c-s-a.org.cn

的显示及提供一些表单的提交,无法在本地完成一些复杂计算。而一次性口令的很多计算任务是需要在本地完成的如:对相关数据的加密,解密等,因此在实现方式上比较困难。

(2) 验证服务器上:HTTP 协议具有简单、快速、灵活、方便等特点。基于 Web 的应用系统对客户端的要求同样灵活、方便,只需客户端具有一个浏览器即可,而无需设计设计专门的客户端,大大减少了开发成本、维护成本等。在每次使用中,只需要打开浏览器,输入正确的域名地址,便可从服务器端下载可与用户交互的页面。同样,基于 WEB 的应用系统每次的登陆界面都必须到服务器下载,而不是一直保存在客户本地的。因此基于 C/S 的一次性口令相互认证方法在 WEB 中是不适用的,同时基于这一点也给网络钓鱼攻击者带来了可趁之机,一旦 WEB 应用系统的用户在输入个人信息时遭受网络钓鱼攻击,则用户输入的原始资料将全部被攻击者获取。

3.2 设计难点解决

在实现上我们可以通过 Applet 与 Servlet 通信及 JavaScript 相结合的方式解决,即:简单的计算我们可以通过 JavaScript 嵌入到 HTML 页面中解决,复杂的我们通过 Applet 与 Servlet 采用 HTTP 对象流的通讯策略既能解决本地计算问题,又能完成客户端与服务器复杂数据的传递问题。

在验证服务器上,最好的方法莫过于基于证书的数字签名认证,然而这种办法虽然安全,但必须以完整的 CA(证书授权中心)体系为基础,而且国内目前尚未法律认可的公正的第三方,而且这些方式技术复杂、成本高,所以无法很有效的普及这种认证方式。同时采用这种方法也失去了一次性口令的特点之一,即:无需第三方公正。

然而我们可以借鉴数字签名的一些特点,通过在服务器端输入一些用户熟悉的、无需记忆的密语我们称之为服务器标识(ServerId)如:

你最喜欢的宠物:Dog,你最难忘的宠物名字:Lily

你最难忘的日子:2008.07.01,这一天我顺利完成了学业,并取得了硕士学位

你最喜欢的一首诗:XXX

你最喜欢的看的一本书:《东周列国志》

你最想去的地方:国内:西双版纳,香格里拉。国外:澳大利亚

注册本系统的时间:2007.03.03

……

这些密语在理解之后就不需要记忆了,因为你已经记住了。只有和生命无关,或者单调的、孤立的的学问,需要你去“记忆”,然而上述的密语是一种和你生命活动紧密关联的知识学问,已经灌注到你的生命的内在层面中,所以就无需记忆。同时在每次登录过程中我们将这些密语经过加密然后传输传输是,所以攻击者无法获取服务器标识。

同时在论文提出的方案中,用户将拥有两个口令:普通口令 + 图片口令。经典认知科学的有关实验表明,人对图画的记忆几乎是没有限制的,不仅记忆的信息量极大,而且记忆的准确性也极高。因此,可以尝试将常规图像转化为口令的方式,即:图片口令。具体的算法描述见具体方案。

4 B/S 架构下的一次性口令身份认证方案描述

本方案的设计中用到了 RSA 加密算法,AES 加密算法,MD5 加密算法. 对这些算法的安全性,性能特点在此不作论述。

4.1 方案符号与标识

为了方便论述,以下将用到的各记号意义如下:

- LogonSid 服务器标示,登陆过程用来让用户验证服务器。
- ModSid 服务器标示,用于用户修改资料验证
- ModLNum:修改时所需的用户三个幸运数的乘积
- Uid:用户 ID
- Pw:用户密码(普通口令)
- Pic:图片(图片口令)
- Kp:私秘密钥,Kc:公共密钥
- E:加密,注册与修改过程加密算法采用 RSA,验证过程采用 AES
- D:解密
- MD5:将字符串进行 md5 信息摘要
- List:服务器端用户名表
- Suid:服务器存储的用户 ID
- Spw:服务器端存储的用户密码

© 中国科学院软件研究所 http://www.c-s-a.org.cn

- K:**共享密钥**, K = MD5 (MD5(Pw)//R)
- R:**认证过程中服务器端产生的随机数**
- //:**连接符,将两个字符串进行连接**
- ⊕:**将两个字符串进行抑或运算并生成新的字符串**
- Authen:**当次登陆的一次性通行口令**

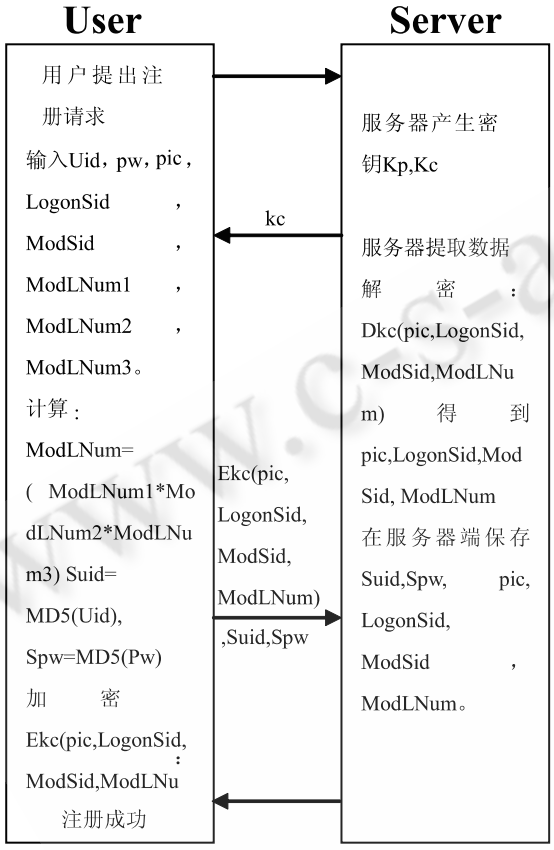

图 1　一次性口令方案注册过程

4.2 注册过程算法描述

注册过程要求用户输入,用户名 Uid,**口令** pw,**认证图片** pic,**登录验证服务器标识** LogonSid,**修改验证服务器标识** ModSid,**三个修改所需的用户自己认为的幸运数字** ModLNum1、ModLNum2、ModLNum3 **如图 1 所示。具体过程如下**:

(1)**用户提出注册请求**。

(2)**服务器用非对称加密产生密钥对** Kp,Kc,**将公钥** Kc **传给客户**。

(3) **用户输入** Uid, pw, pic, LogonSid, ModSid, ModLNum1, ModLNum2, ModLNum3。**在本地计算**, ModLNum = (ModLNum1 * ModLNum2 * ModLNum3), Suid = MD5(Uid), pw = MD5(Pw)、**对认证图片** Pic,**服务器标识**(LogonSid,ModSid)**及** ModLNum **进行加密并将** Ekc(pic,LogonSid, ModSid, ModLNum),Suid,Spw **传给服务器。将** Ekc(pic,LogonSid, ModSid, ModLNum), Suid, Spw **传给服务器**。

(4)**服务器解密** Dkc(pic,LogonSid,ModSid,ModLNum)**得到** pic,LogonSid,ModSid, ModLNum **并在服务器端保存** Suid,Spw, pic,LogonSid, ModSid,ModLNum。

(5)**注册成功**。

4.3 认证过程算法描述

为了进入系统,用户在登录时必须采用挑战/应答方式执行一次身份认证过程。本方案中登陆过程中身份认证过程如图 2 所示。

步骤一:**用户在登陆页面输入** Uid,**点击登录,计算** Uid' = MD5(uid),**并将** Uid' **传给服务器**。

步骤二:**服务器接受** Uid',**判断** Uid' **是否属于** List,**若** Uid' ∈ List,**表明用户输入的** Uid **合法,服务器产生随机数** R,**计算** K = MD5(Spw//R),**将** LogonSid **用** AES **算法加密,即**: LogonSid' = Ek(LogonSid),**将** LogonSid',R,**发送给客户,同时服务器端存储** K;**否则,用户输入的** Uid **不合法,终止与用户的会话。**

步骤三:**客户端接收来自服务器端的验证信息** LogonSid',R。

(1)**用户输入** Pw。

(2)**用相同算法计算** K' = MD5(MD5(Pw)//R)。

(3)**解密得到** LogonSid = Dk'(LogonSid'),**用户确认如果** LogonSid **正确则继续进行验证。否则终止验证登陆过程,登陆修改用户信息页面。**

(4)**输入验证图片,计算图片** MD5 **值,算法如下**:

①**读取图片的宽度** w(**以象素为单位**)**和高度** h,**将图片平均切割成** 9 **个区域**;

②**在每个区域中随机选取一个象素点** Pixel,**共** 9 **个记为**:

pixel $[w_1]$ $[h_1]$, pixel $[w_2]$ $[h_2]$ …… pixel $[w_9]$ $[h_9]$, **同时保存该** 9 **个象素点坐标集合记为**: $w^{[9]}$, $h^{[9]}$。**分别提取这就个象素点的** R, G, B **值记为** R_1、G_1、B_1, R_2、G_2、B_2 …… R_9、B_9、G_9。

③**计算** MD5(Pic) = MD5 (R_1//G_1//B_1//R_2//G_2//B_2 …… R_9//B_9//G_9)。

④**计算** Authen = Md5(Pic) ⊕ K'

© 中国科学院软件研究所 http://www.c-s-a.org.cn

⑤将 Authen 和图片象素采集点 $w^{[9]}$,$h^{[9]}$ 传给服务器

步骤四:服务器端接收来自客户端的数据和 Authen,w[9],h[9]

(1)提取图片的坐标集合(w_1,h_1),(w_2,h_2)……(w_9,h_9),提取图片上该 9 坐标象素点的 R,G,B 值。并以同样的算法计算 该 9 个象素点 R,G,B 值的 MD5 值,记为:MD5'(Pic)

(2)取出存在服务器端的密钥 K,计算 Authen' = Md5'(Pic)⊕K 若 Authen = Authen' 则验证通过,否则验证失败。

4.4 修改过程算法描述

用户需要修改个人认证信息,资料。则同样要通过服务器认证,登陆后方可以修改,因此同样需要一个登陆验证过程,其修改登陆验证过程与上述验证过程相同,不同的地方有如下几点:

(1)在双口令的适用顺序上刚好与登录过程相反,加密密钥 K 的计算上用图片的 9 个区域的 9 个象素点的 R,G,B 的值得连接的 MD5 值作为密钥。即改成与服务器完成对图片的切割产生 9 对图片的坐标,并将坐标集传给服务器,同时计算密钥 K。K 的计算方法如下:K = MD5(Pic) = MD5(R_1//G_1//B_1//R_2//G_2//B_2……R_9//B_9//G_9)

(2)在验证服务器用语变成了 ModSid。

(3)在计算验证通行语 Authen 时除了需要 MD5(pic),用户通用口令外,还需要需加入用 户的三个幸运数字,同时随机数 R 由客户端生成。具体计算方法如下:

①客户端:

Authen = MD(MD5(pw)//R//ModLNum)⊕MD5(Pic)

②服务器端:

Authen = MD(Spw//R//ModLNum)⊕MD5(Pic)

验证通过后,用户便进入修改页面,修改采用加密算法同样为 RSA 加密算法,其过程与注册过程类似。

5 图片处理算法分析

从理论上讲如果在图片上随机选取一个象素点,

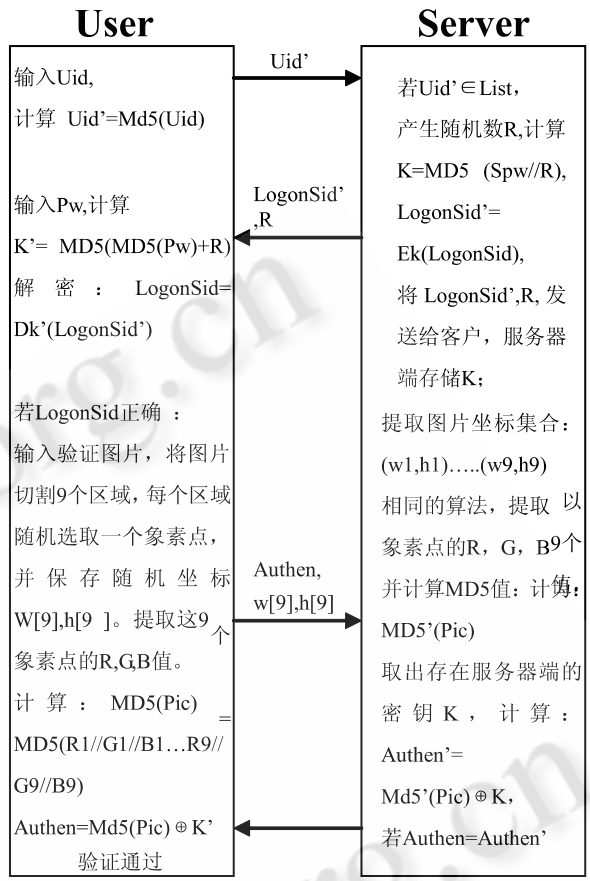

图 2　一次性口令方案认证过程

计算其 R,G,B 值的 MD5 值,那么不同图片在此点 R,G,B 值相同的概率为 1/256*256*256,因此我们选取一个点其实已经足够,但是一张图片连续象素点之间 RGB 值变化不大,而且相同的可能性是十分高的,这就代表不了用户图片的特征,也就失去了选取图片作为口令的一个重大大意义之一。然而如果随机选取图片上一个区域计算其 MD5 值,若该区域太小同样会存在上述问题;若区域面积太大则计算量增加,影响系统性能。因此本论文采取将图片切割成 9 块,在每块中随机采取一个象素点,将其 R,G,B 值连接并计算 MD5 值,有效解决了连续象素点 RGB 值相同问题,提高图片口令的唯一性,安全性,同时解决了因象素点过多而影响性能的问题。另外,一幅 1024*768 的图片,平均切割成 9 个区域,每次在 9 个区域随机选取 9 个点,其可能的选取组合空间约为 87381。因此保证了每次在网络中传输的图片口令的随机性、一次性。

© 中国科学院软件研究所 http://www.c-s-a.org.cn

6 方案对各种攻击的防御及性能分析

(1)身份信息的保护。用户的身份标识 Uid 发送给服务器时,使用 MD5 加密算法处理,同时在网络服务器端存储的用户名是经过 Md5 加密的,因此攻击者不可能从截获的数据中解析出用户标识 Uid。此外,用户口令 pwd 只在客户端出现,且在服务器端存储的也是 Md5 值。因此攻击者不可能通过社交工程从服务器端获得有效的用户名和密码。

(2)重放攻击。由于每次认证服务器端产生的随机数 R,和客户端随机产生的图片 9 个区域的 9 个象素点各不相同,而且不重复,保证了每次传输的认证数据 Authen 唯一性,一次性。因此攻击者重放已经截获的信息是无法通过服务器认证的。

(3)认证过程是安全的。假设入侵者可以截获合法认证过程中的任何通信数据进行分析,他从截获的 ID中是无法获取 Uid 的,当然他直接发送截获的 ID'也能通过服务器的初步认证,但由于口令 Pw 和认证图片 Pic 根本不通过网络传输,因而他根本不可能从截获的数据中解析出 Pw 认证图片 Pic,准确算出校验符 Authen 的值,从而无法通过下一步的认证,故方案的整个认证过程是安全的。由于用户标识 Uid 加密传输,口令 Pw,密钥 K,认证图片 Pic 不通过网络传输以及随机数 R 和图片象素点集的不重复性,因而字典攻击、口令字猜测等常用的攻击手段对此是无能为力的。

(4)假冒攻击。方案的认证过程可以抵抗来自客户端的假冒攻击,这是因为攻击者不知道用户标识 uid、口令 Pw,认证图片 Pic 从而不能算出最终一次性通行口令 Authen 的值,而无法通过服务器的认证。同样的,该认证过程也能有效地防御来自服务器端的假冒攻击(包括钓鱼攻击。),不会使合法用户的合理要求得不到满足。当用户向服务器提出认证请求时,要求服务器发送的挑战报文中包含经过 AES 加密算法加密过的服务器标识 logonSid(修改过程为 modSid)和计算密钥 K 所需的随机数 R ,整个过程密钥 K 不经过网络传输,且每次认证过程密钥 K 是随机产生的一次性密钥,用户可以用服务器发送过来的随机数 R,用户口令 Pw 计算出 K,从而解密出服务器标识语,确认后才继续进行验证登陆。如果发现服务器标识语不对,用户可马上通过正常渠道(如用户手动正确输入域名地址,而不是通过可能带有欺诈性电子邮件中的假冒的容易混淆的链接的或者其它陌生网络链接等)登陆系统的修改模块,进行行相应的验证登陆后修改相关验证资料,从而有效防止来自服务器的欺骗。

由于本系统采用普通口令和图片口令相结合的双口令技术,因此,即使用户在登陆的前半部分,即用户输入了通用口令 Pw,发现来自服务器的标识语 logonSid 不正确,或者根本没有,则此时用户可能中了钓鱼攻击,攻击者也就可能从用户手中成功骗取了 Uid,Pw,然而但用户发现服务器标识语不正确,或者出现莫名错误的时候,此时用户便停止了继续登陆的过程,因此攻击者无法获取用户图片,不能计算出认证所需的一次性口令 Authen 值,从而不能够登陆系统。然而攻击者可以利用 Uid,Pw 尝试登陆系统,从而进进一步得知用户在服务器存储的服务器标识 logonSid,因此当出现服务器标识语有问题时,应该尽早修改相关登陆资料。

在效率方面,本方案在认证过程中客户端共进行 MD5 散列计算 4 次,服务器端共进行 MD5 散列计算 2 次,共 6 次。相比较基于 Lamport 方式的一次性口令认证方案 S/KEY 口令序列认证系统的 N 次散列计算来说效率的提高是明显的,而且 6 次散列计算比较与同类型的基于挑战/应答方式的一次性口令系统其效率也是高的,同时本方案 6 次散列计算客户端就占据 4 次,因此有效减轻了服务器端的压力。在随机数生成方面,服务器端只产生一个随机数 R,而将产生 9 对随机数的任务的工作移到客户端执行,进而大大减少了服务器的额外开销,提高了服务器的运行效率。在信息传送次数方面:客户端详服务器传送 2 次,服务器端向客户端传送 1 次。

7 结束语

本文提出以挑战/应答方案为基础,基于 MD5 安全单向散列函数,AES,RSA 加密算法,服务器标识等设计了一种能有效适用于 B/S 架构的一次性身份认证方案。该方案能够对通信双方实行相互认证,减少了服务器的开销,克服了传统的挑战/应答方案的弱点,有效地保护了用户身份信息,能防止重放攻击,网络钓鱼攻击等常用攻击手段的攻击,通过增加图片口令,增强了认证过程的安全性的同时解决了口令记忆问题。但本方案还存在用户信息修改过程,(下转第 37 页)

© 中国科学院软件研究所 http://www.c-s-a.org.cn

(上接第 27 页)

计算过程相对复杂等不足。所以本方案将来还需进一步完善和提高。

参考文献

1 LamportL. Password authentication with insecure communication. Communications of the ACM, 1981, 24(11):770 -772.

2 杨俊,景疆. 浅谈生物认证技术 - 指纹识别. 计算机时代,2004,17(2):3 -4.

3 冯登国,网络安全原理与技术. 北京:科学出版社,2003,39 -91.

4 张宏,陈志刚. 一种新型一次性口令身份认证方案的设计与分析. 计算机工程,2004,30(17):112 -114.

5 Hung - Yn Chien and Jinn - Ke Jan, Robust and Simple Authentication Protocol. The Computer Journal, 2003,46(2):193 -201.

© 中国科学院软件研究所 http://www.c-s-a.org.cn