

简单易行的 S/KEY 认证改进方案

何冰

(广西师范大学 计算机科学与信息工程学院, 广西 桂林 541004)

(hebing@mailboxgxnu.edu.cn)

摘要:分析了传统 S/KEY 一次性口令(OTP)认证方案及现有的改进方案存在的缺陷与不足,提出了一种新的 S/KEY 认证改进方案。该方案以用户口令 PW 哈希值作为验证因子实现了双向认证,通过增加消息的完整性保护防止关键消息被伪造,并具备原有方案简单易行的特性,能有效抵御重放攻击、小数攻击和冒充攻击。

关键词:一次性口令;身份认证;S/KEY 认证

中图分类号: TP393.08 **文献标志码:** A

Simple improvement for S/KEY authorization scheme

HE Bing

(College of Computer Science and Information Technology, Guangxi Normal University, Guilin Guangxi 541004, China)

Abstract: The author analyzed some defects of the traditional S/KEY One-Time Password (OTP) authorization scheme and recent improvements, and proposed a new S/KEY improvement scheme. The new scheme provided mutual authorization with Hash values of user password as an authentication factor. It can effectively prevent key message from being forged by adding message integrity protection. The new scheme is as simple and easy to be implemented as traditional S/KEY scheme. Additionally, it can effectively avoid replay attack, small integer attack and impersonating attack.

Key words: One-Time Password (OTP); identity authorization; S/KEY authorization

0 引言

身份认证是确保系统资源不被非法用户非法访问的重要保证。在不安全的网络环境中,口令认证方式是最简单、最便利的身份认证机制之一。口令认证被广泛应用于各类网络应用系统,如政府组织网站、企业内及校园网内的各类信息管理系统。然而,当前网络环境是不安全的,存在各类攻击,如重放攻击、冒充攻击等。因而,亟须为各类网络应用系统提供一种具有较高安全性的,并易于实施的口令认证机制。

传统口令认证采用静态口令技术,即用户口令以明文形式且固定不变地在网络上传输,在服务器端数据库中存储。随着网络攻击手段的深入化和多样化,静态口令技术已不能满足各类网络应用系统的安全需求。为了解决这些问题, Lamport^[1]提出了采用 Hash 链的一次性口令技术(One-Time Password, OTP)。基于 Lamport 方案, Haller^[2]提出了 S/Key 方案。该方案能有效防止口令重放攻击,服务器端不需保留用户口令,具有简单易实施的特点。但随着进一步研究,人们发现该方案面临冒充攻击和小数攻击的威胁^[3]。文献[4-9]中针对 S/Key 方案的安全问题提出了改进方案,这些改进方案为了提高认证的安全性,或者需要在用户端配置存储卡甚至智能卡,或者要求实现用户端和服务端的时间同步,虽然增强了安全性却不方便用户,同时增加了实施的难度。

本文通过对 S/Key 方案及近年的改进方案进行分析,提出了一种新的 S/Key 的改进方案,实现了用户端与服务端的双向认证,解决了 S/Key 方案面临的安全问题,同时不需要增加用户端的存储负担,保持了 S/Key 方案简单易实施的

1 S/KEY 认证方案及目前改进方案分析

S/KEY 认证方案是一种比较经典的一次性口令认证方案,能有效防止攻击者利用窃听到的旧口令进行重放攻击,但仍存在严重的缺陷,很多研究都针对其安全问题进行了分析和改进。以下给出 S/KEY 认证方案存在的安全问题,并对目前的改进方案进行分析。

1.1 传统 S/KEY 认证方案

传统 S/KEY 认证方案具有以下特点:1)用户每次登录系统所用的口令都不一样,能有效防止攻击者利用窃听到的旧口令进行重放攻击;2)在服务器端没有用户口令明文 PW,而所存储的一次口令是变化的,能有效防止攻击者通过攻击服务器端数据库获得用户口令;3)用户只要记住用户名和口令即可登录,不需要任何存储工具,方便用户使用,也利于方案在网络环境下实施。

但同时传统 S/KEY 认证方案也存在以下安全缺陷。

1)传统 S/KEY 认证方案实际上是一个单向认证协议,只是服务器对用户身份的认证,没有实现用户对服务器身份的认证。攻击者可以冒充服务骗取用户的一次有效口令,再伪装成合法用户登录服务器,因而无法抵御冒充服务器攻击。

2)传统 S/KEY 认证方案中的种子值 seed 和第 i 次认证序列数 $N-i$ 是以明文形式进行传输的,攻击者通过篡改 $N-i$ 为一个很小的数值 m 给用户,能骗取得到用户的口令序列中在 m 与 $N-i$ 之间的有效口令,也就是说无法抵御小数攻击。

1.2 现有 S/KEY 认证系统改进方案

近年来很多文献针对传统 S/KEY 认证方案存在的冒

充服务器攻击、小数攻击提出了相应的解决方案,主要有以下三种改进方式。

1) 引入存储卡或智能卡。文献[4]将基于公钥算法的会话密钥协商与 S/Key 认证方案有机地结合在一起,方案的每次认证口令和会话密钥都不同,增强了对重放攻击的抵御能力。通过用户端与服务器端之间的共享密钥 K_{CS} 实现双向认证,并利用会话密钥协商期间所产生的随机数与种子值 $seed$ 和第 i 次认证序列数 $N-i$ 进行异或运算防止攻击者进行小数攻击。但该方案对破坏协议攻击与中间人攻击仅能通过会话密钥保护其后的通信,并不能有效防止攻击者通过认证;且在注册过程中,用户端需要用 USBkey 保存与 S 的共享密钥 K_{CS} ,若 USBkey 丢失,则面临着严重的安全威胁。文献[5]对带智能卡的认证方案进行了详细分析,并指出了带智能卡的认证方案应满足的 10 项需求目标,提出了一种新的基于智能卡的认证方案。该方案能很好地解决在服务器端密码表或认证表存储不安全的问题,同时在减轻计算量及提供用户修改口令方面做了改进;但当智能卡丢失时,仅能防止用户口令被猜测,不能防止攻击者假冒用户的身份通过认证。

2) 引入混沌理论。文献[6]将混沌理论中混沌序列应用到一次性口令当中,利用数字信号去模仿混沌序列,每次认证通过硬件不同的随机数,能有效抵御重放攻击;同时也由于使用混沌序列的随机数代替了 Hash 链方式,避免了遭受小数攻击的问题,增强了认证的安全性,减少了时间和空间的开销。但需要在用户端设置混沌随机数发生器,增加了实施的难度。文献[7]应用混沌 Hash 函数代替传统的 Hash 函数,用混沌置乱排序算法产生迭代次数代替网上明文传输的迭代次数,可以有效抵御小数攻击,并能解决明文传输认证序列数 $N-i$ 的不安全问题。但由于认证过程用户端和服务器端都要存储同样的置换表,既增加了用户端的存储负担,又面临着置换表存储的安全性问题。

3) 增加时戳作为验证因子。文献[8]中提出了一种基于质询响应的一次性口令认证方案,通信双方通过检测对方对非重复质询作出的响应值是否正确来实现相互认证。同时在质询值中引入随机数和时戳两个随机因素来抵御重放攻击,由于没有采用 Hash 链方式,减少了运算量,不存在小数攻击。此外该方案采用纯软件方式进行认证,不需要用户保存任何认证信息,即不依赖于智能卡、USB 卡等硬件存储设备。但该方案通过时戳是否过期来判断是否存在冒充攻击,因此方案实施要求实现通信双方时间同步,这给方案的实施带来困难。文献[9]中提出一种能抵御重放攻击、内部攻击,同时具备安全可修复性的使用智能卡的一次口令认证方案,但同样采用时戳作为抵御重放攻击的随机因素,需要实现时间同步。

从以上改进方案的分析来看,现有改进方案主要存在两方面的问题。

1) 增加了用户的存储负担。用户不能仅依靠用户 ID 和口令进行认证,必须配置相关存储设备去记录难以记忆的验证因子、置换表或会话密钥。这样,若存储卡或智能卡丢失,存在用户身份被假冒的危险。此外用户日常生活和工作经常需要登录到各类不同的服务器,就必须携带各类服务器的存储卡或智能卡,给用户使用带来不便。这导致方案实施受到一定的限制,不适用于对安全强度有一定要求的网络应用系统。

2) 增加了方案的实施难度和代价。如采用时戳作为验

证因子,由于需要实现时间同步而不利于方案实施;而引入混沌理论方法,用户端必须设置混沌随机数发生器。

2 新 S/KEY 认证方案

2.1 符号与标识

以下是本文中所用到符号的含义说明: $A \rightarrow B: x$ 表示 A 向 B 传送信息 x ; C, S 分别表示用户端、服务器端; ID_C, PW 分别表示用户身份标识、用户的口令; $seed$ 表示 S 为 C 生成的种子值; $N, N-i$ 分别表示口令序列的元素个数、当前序列数(第 i 次认证序列数); R_C 表示服务器端生成的随机数; $h(x), h^N(x)$ 分别表示对 x 进行一次哈希运算及对 x 进行 N 次哈希运算; $\oplus, +$ 分别表示异或运算、联结运算。

2.2 方案目标

方案目标如下:

- 1) 实现用户端与服务器端双方相互认证;
- 2) 保证协议中关键消息的完整性,即防止种子值 $seed$ 和第 i 次认证序列数 $N-i$ 被篡改或伪造;
- 3) 服务器不保留用户的秘密口令 PW ;
- 4) 认证过程用户只要记得口令,而不需要存储任何信息,即不需配置存储卡或智能卡;
- 5) 方案实施简单易行,不增加实施的代价。

2.3 注册过程

首先, C 随机选择口令 PW , S 为每个新 C 生产一个随机数种子 $seed$, 然后由 C 设置一次性口令序列的最大元素个数 N , 计算出初始动态口令 $p_0 = h^N(seed + PW)$ 及 $h(PW)$, 交给 S 保存。 S 为用户 C 的保存的注册数据内容为 $ID_C, P_0, seed, N$ 。而 $h(PW)$ 作为 C 认证 S 的验证因子, 要求 S 对其进行严格保护。注册过程必须在安全环境下, 由 C 和 S 协商完成。

2.4 认证过程

第 i 次认证过程如下(第 i 次认证登录时 S 保存的数据内容为: 用户 ID_C , 上次口令, $P_{i-1}, seed, N-i+1$)。

步骤 1 $C \rightarrow S: ID_C, R_C \oplus h(PW)$, 认证请求。

步骤 2 $S \rightarrow C: N-i, seed, h(N-i+seed+R_C)$ 。

步骤 3 $C \rightarrow S: P_i$ 。

步骤 4 $C \rightarrow S: Success$ 或 $Failure$ 。

步骤 1 中 C 生成随机数 R_C 作为认证 S 的挑战, 与验证因子 $h(PW)$ 进行异或运算后, 将计算结果与身份标识 ID_C 一起发送给 S 提出认证请求。

步骤 2 中 S 收到认证请求后, 用所保留的 $h(PW)$ 与 $R_C \oplus h(PW)$ 进行异或运算提取出挑战值 R_C , 并计算 $h(N-i+seed+R_C)$, 再将计算结果连同 $N-i, seed$ 一起发送给 C 。

步骤 3, C 收到消息后, 首先将收到 $N-i, seed$ 与步骤 1 生成的 R_C 进行哈希运算, 并将运算结果与收到的 $h(N-i+seed+R_C)$ 进行比较: 若匹配, 说明 S 身份是真实的, 计算 $p_i = h^{N-i}(seed+PW)$, 并发送给 S ; 否则可断定存在冒充的服务器, 中止认证过程。

步骤 4, S 收到消息后, 用收到的 p_i 计算 $h(p_i)$, 结果与所保留的上次口令 p_{i-1} 进行比较: 若匹配, 给 C 发送认证 $Success$ 消息, 并用 p_i 替换 p_{i-1} ; 否则, 给 C 发送认证 $Failure$ 消息。

3 新方案安全性分析

3.1 安全性分析

新方案与原方案一样, 每次认证口令都不同, 并且一次有效, 能有效防止重放攻击。

新方案中, 服务器没有用户的口令 PW , 存储的是一次性口

令 $p_i = h^{N-i}(seed + PW)$, 所以即使攻击者获得服务器端的口令文件, 也无法通过该文件得到用户的口令, 假冒合法用户。

新方案实现了双向身份认证。在注册阶段, S 保存验证因子 $h(PW)$, 在步骤 1 中 C 发送验证因子与随机数 R_c 异或运算的结果。因为只有 C 和 S 拥有 $h(PW)$, 所以只有真正的 S 才能正确提取 R_c 。这样用户在步骤 2 中接收到 S 发送的响应消息后, 通过收到序列值 $N-i$ 、种子值 $seed$ 与生成的 R_c 进行哈希运算, 并与收到的 $h(N-i+seed+R_c)$ 进行比较; 若匹配则说明对方拥有验证因子, 能正确提取 R_c , 是真实的服务器; 否则用户可以终止认证过程。所以能有效地抵御冒充服务攻击。

新方案的步骤 2 中不仅发送了明文的序列值 $N-i$ 及种子值 $seed$, 还包括了防止 $N-i$ 、 $seed$ 被篡改或伪造的消息哈希值 $h(N-i+seed+R_c)$ 。即使攻击者通过侦听截获了明文的 $N-i$ 、 $seed$, 但无法获得 R_c , 也就无法伪造 $h(N-i+seed+R_c)$, 使得认证不能继续进行。所以攻击者无法成功进行小数攻击。

3.2 新方案特点

1) 与文献[4-5]中的改进方案比较, 新方案实现了双向认证的关键验证因子由服务器端保存, 故用户不需要携带存储卡或智能卡, 仅需记住用户 ID 和口令即可进行身份认证。

2) 与文献[6-9]相比, 新方案不需要时间同步或设置混沌随机数发生器, 更易于实施。

4 结语

改进后的 S/KEY 方案实现了用户端与服务器端之间的双向认证, 并实现了对协议关键消息种子值 $seed$ 和序列数 N 的完整性保护, 有效地抵御了冒充服务器攻击和篡改攻

击, 具有更高的安全度。同时与现有的改进方案不同, 新方案没有增加用户的存储要求, 保留了传统 S/KEY 方案简单灵活、易于实施的特点, 适用于远程登录、政府企业网和校园网等各类对安全度有较高要求的网络应用系统。

参考文献:

- [1] LAMPORT L. Password authentication with insecure communication [J]. Communications of the ACM, 1981, 24(11): 770-772.
- [2] HALLER N M. The S/Key one-time password system [C]// Proceedings of the Internet Society Symposium on Network and Distributed System Security. San Diego, CA: [s. n.], 1994: 151-158.
- [3] MITCHELL C J, CHEN L. Comments on the S/KEY user authentication scheme [J]. ACM Operating System Review, 1996, 30(4): 12-16.
- [4] 谢志强, 郭军, 杨静. 新型 S/KEY 认证方案的分析与设计[J]. 计算机工程, 2009, 35(5): 175-176.
- [5] LIAO I E, LEE C C, HWANG M S. A password authentication scheme over insecure networks [J]. Journal of Computer and System Sciences, 2006, 72(4): 727-740.
- [6] 梁结, 方勇. 融入混沌理论的一种 OTP 方案研究[J]. 计算机工程, 2007, 33(7): 160-162.
- [7] 姜楠杨, 杨德礼, 王德高. 基于混沌理论的身份认证方案[J]. 吉林大学学报: 理学版, 2008, 46(4): 710-715.
- [8] 高雪, 张焕国, 孙晓梅. 一种改进的一次性口令认证方案[J]. 计算机应用研究, 2006, 23(6): 127-128.
- [9] CHEN G, CHEN S M. Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards [J]. IEEE Transactions on Consumer Electronics, 2004, 50(1): 204-206.

(上接第 944 页)

由表 1 数据可知, 实验模拟了 15 次数据异常情况, 系统有高达 12 次收到错误数据却认为成功更新, 并更新了启动标志, 导致启动失败后系统停止运行。造成这一问题的原因是该实验系统重点不在验证校验的性能, 仅在数据校验上作了一些简单处理, 校验不充分。

从表 1 中第 2, 3, 4 种异常的模拟方式得到的数据可以看出, 在没有收到完整数据, 或者意外断电的情况, 由于没有更新启动的硬件标志, 系统都不会出现停止运行的情况。

从总的结果来看, 如果加入大量数据校验, 可以预见, 在没有正确更新时, 系统将不会从新版程序中启动。

该系统存在一个隐患, 即加入大量的校验之后, 仍然有很小概率可能将错误的数据校验成功, 并最终更新启动标志, 导致启动失败。这里提出一种“启动尝试”的方法:

1) 更新完成后设置可尝试启动次数 $N = m$ (N 为硬件标志, 不随复位发生改变)。

2) 每次启动到执行入口判断前, 首先 $N-1$, 然后根据入口标志给程序指针赋值。

3) 进入程序后, 程序首先设备自检, 运行一次所有代码, 将结果报告中心站。如果自检成功, 中心站发送命令使 $N+1$, 即 N 的值不变。

4) 如果死机导致自检失败, 则极可能误判更新成功, 中心站不发送 $N+1$ 命令。由于外部看门狗不断复位, 多次复位 $N-1$, N 最终为 0。

5) 当 $N=0$ 时, 则启动了 m 次都失败, 系统否定之前对更新成功的判断结果, 恢复原入口标志, 同时置 N 为任意非零整数。

在“双系统”的前提下, 这种方法理论上可以保证任何情况下系统都不会停止运行。

8 结语

实验结果表明, 在通信、校验等条件满足的情况下, 本文给出的存储体系结构可以保证设备在更新失败时仍可以正常运行; 扩展的存储结构也为大型工程的远程升级提出了支持。文中最后的潜在隐患的分析, 也为解决数据校验失败以及其他各种小概率错误提供了解决方案。

参考文献:

- [1] 章杰. 基于 ARM7 的远程升级的实现[J]. 福建电脑, 2009, 25(11): 175-176.
- [2] Uml0237 LPC24XX User Manual Rev. 03 [EB/OL]. [2009-09-15]. http://www.zlgmcu.com/philips/NXP_ARM_2400.asp.
- [3] 周立功. 深入浅出 ARM7[M]. 北京: 北京航空航天大学出版社, 2005: 426-438.
- [4] 张舞杰, 南亦民. 基于 STM32F103VB 的应用编程技术的实现[J]. 计算机应用, 2009, 29(10): 2820-2822.
- [5] 姜晓梅. 基于 ARM 的 IAP 在线远程升级技术[J]. 计算机应用, 2008, 28(2): 519-521.
- [6] 孙雅如. 计算机操作系统[M]. 西安: 西安电子科技大学出版社, 2003: 76-100.
- [7] 夏爽. ARM 处理器分散加载及特殊应用研究[J]. 单片机与嵌入式系统应用, 2009(4): 36-39.
- [8] 王恒. 基于 Bootloader 的可靠嵌入式软件远程更新机制[J]. 微计算机信息, 2007, 23(20): 57-59.
- [9] 王恒. 一种高可靠的嵌入式软件远程自更新机制的研究与实现[J]. 工业控制计算机, 2007, 20(9): 39-43.
- [10] 王磊. U-Boot 从 NAND FLASH 启动的实现[J]. 电子设计工程, 2010, 18(5): 98-100.
- [11] 库少平, 田云芳. 基于 Nand Flash 的 VIVI 装载器的分析与改进[J]. 微计算机信息, 2009, 25(8): 76-78.