

基于置乱排序的身份认证协议研究

姜楠^{1,2}, 杨日璟¹, 袁克杰¹, 刘向东¹

(1 大连民族学院 计算机科学与工程学院, 辽宁 大连 116600; 2 大连理工大学 系统工程研究所, 辽宁 大连 116024)

摘 要: 在分析经典的身份认证方案, 特别是一次性口令认证方案中 S/KEY 方案存在问题的基础上, 提出了一种新的身份认证协议 COTP. 在本协议中, 采用了基于混沌置乱排序表的设计算法, 在用户登录过程中用置乱排序后的次数代替真正的迭代次数, 对协议中用到的种子进行了加密处理. 理论分析与仿真实验表明, COTP 认证协议能抵御小数攻击、重放攻击及内部人员攻击, 能对传输的种子和迭代次数进行加密, 并实现了客户和服务器的双向认证.

关键词: 混沌; 置乱排序表; Hash 函数; 认证协议

中图分类号: TP309

文献标识码: A

文章编号: 1000-7180(2010)06-0001-04

Study of Identity Authentication Protocol Based on Scramble Sort

JIANG Nan^{1,2}, YANG Ri-jing¹, YUAN Ke-jie¹, LIU Xiang-dong¹

(1 School of Computer Science & Engineering, Dalian Nationalities University, Dalian 116600, China;

2 Institute of Systems Engineering, Dalian University of Technology, Dalian 116024, China)

Abstract: By analyzing typical authentication schemes, especially S/KEY schemes in one time password, a new authentication protocol COTP is proposed. Different from other one time password protocols, it gives the algorithm of chaotic scramble sort table and applies the scramble sort algorithm instead of real iteration times in the log on process. Gives the idea of encrypting to the seeds of protocol, theoretical analysis and computer simulation demonstrate that the COTP can resist little numbers attack, replay attack, internal attack and encryption seed and the times of iteration for transform, that realizes authentication between client and server.

Key words: chaos; chaotic scramble sort table; Hash function; authentication protocol

1 引言

互联网是一个高度开放的网络, 随着互联网应用的普及, 网络安全的问题也日益明显. 为了保证互联网上电子交易的安全性, 防范交易及支付过程中的欺诈行为, 在信息传输过程中一方面必须采用足够强度的加密技术来保障信息传输的安全, 另一方面还必须采用多种安全认证技术来确认信息发送者的身份, 验证信息的完整性, 确认信息未被篡改过.

目前使用的验证机制大多数基于知识证明, 其中最常用的是使用固定口令进行用户身份验证, 但以固定口令为基础的验证方式存在很多问题, 它的

安全性仅仅基于用户口令的保密性, 一旦约定的口令密码泄漏或被截取, 任何非授权人都可以冒充, 在网络中传输的口令也容易遭受攻击.

针对上述问题, 文中提出了一种新的能够抵御常见攻击, 能进行用户与服务器双向认证的基于混沌置乱排序和混沌 Hash 函数的一次性口令身份认证协议 COTP.

2 典型的一次性口令认证方案分析

通常一个登录者要访问一个系统资源需提供用户名和口令 (Password), 其中口令被看做登录者是“用户名标识的用户”的证据. 系统经核查属实后, 允

收稿日期: 2009-04-28; 修回日期: 2009-06-28

基金项目: 国家自然科学基金项目 (60573124)

许该用户访问系统资源. 固定口令的安全隐患是窃听, 窃听者之后可以重放这个口令. 一次性口令 (One-Time Password, OTP) 解决了这个问题. 美国科学家 Leslie Lamport 在 1980 年代初首次提出了利用 Hash 函数产生一次性口令^[1]的思想, 即用户每次同服务器进行身份认证时, 在网上传输的认证口令都是经过 Hash 函数运算后得到的密文, 并且在每次认证时这些密文形式的认证口令是不同的, 也就是说密文口令是一次有效的.

自从 Lamport 的一次口令认证方案提出后, 国内外很多学者都在进行这方面的研究, 其中非常著名的是 1991 年贝尔通信研究中心提出的基于 Lamport 方式的一次性口令身份认证系统 S/KEY 口令序列认证方案^[2]. 在 S/KEY 口令序列认证方案中, 有客户端和服务器两个实体参与操作. 服务器端产生挑战数据, 并校验随后客户端送来的一次性口令应答数据, 最后将接收的有效的认证数据及相应的序列号存储在数据记录中. 服务器产生的挑战数据由两部分构成, 即种子 Seed 和迭代次数 Seq 组成, 迭代次数 Seq 为一个整数. 客户端通过用户口令和服务器发来的挑战数据计算一次性口令.

S/KEY 口令序列认证方案的安全性是基于安全 Hash 函数, 在 S/KEY 口令序列认证方案中可以选择 MDx 系列或 SHA 系列作为安全 Hash 函数. 最近, 山东大学王小云教授等人已经设计出对 MD5、HAVAL-128、MD4 和 RIPEMD 等四个著名密码算法快速而有效的碰撞攻击算法^[3,4]. 他们的研究表明, 从理论上讲, 基于 MD5 和 SHA-1 的身份认证可以伪造, 因而必须重新设计更安全的单向 Hash 函数^[5], 以保证电子商务的安全.

S/KEY 口令序列认证方案并没有提供对传输数据的加密, 因此在 S/KEY 口令序列认证方案中, 种子和迭代次数都是以明文形式在网络上传输, 同时 S/KEY 口令序列认证方案也无法防范假冒的服务器欺骗合法用户^[6].

为了抵御常规的 S/KEY 口令序列认证方案^[7]中的安全缺陷和通用 Hash 函数的安全缺陷, 文中设计了基于混沌 Hash 函数和置乱排序的新的一次性口令身份认证协议. 文中从以下方面进行改进: 使用混沌映射构造的 Hash 函数作为安全 Hash 函数, 在服务器端和客户端都用相同置乱排序表对迭代次数进行替换; 在客户端和服务器端采用相同的混沌 Hash 函数算法分别进行散列计算, 然后进行认证.

3 混沌置乱排序算法

为了避免在网上传输的迭代次数和种子被攻击者截获后对系统进行攻击, 文中采用了一种混沌置乱排序算法生成置乱排序表, 置乱的目的在于把原本在网上明文显示的真正的迭代次数转换为变换后的无序的一个数字, 而这个数字并不是 Hash 函数迭代的真正次数, 这使得攻击者在网上传输不到真正的 Hash 函数的迭代次数. 置乱排序算法在客户端和服务端采用相同的混沌系统, 得到相同的混沌置乱排序表. 可以利用任一混沌迭代 $x_{n+1} = f(x_n)$, $x_i \in A$ 来产生混沌实值序列. 置乱排序算法如下^[8]:

Step1: 给定迭代初始值 x_1 ;

Step2: 经 $N - 1$ 次混沌迭代运算得到混沌实值序列 $\{x_1, x_2, \dots, x_N\}$ (N 为最大迭代次数);

Step3: 通过排序变换, 将实值序列集合 $\{x_1, x_2, \dots, x_N\}$ 中的 N 个值由小到大排序, 形成有序序列 $\{x'_1, x'_2, \dots, x'_N\}$;

Step4: 确定混沌实值序列 $\{x_1, x_2, \dots, x_N\}$ 中的每个 x_i 在有序序列 $\{x'_1, x'_2, \dots, x'_N\}$ 中的位置编号, 形成置乱排序表 $T = \{t_1, t_2, \dots, t_N\}$; 其中 $t_i \in T$, $i = 1, 2, \dots, N$, 在客户端与服务器端都存储同样的置乱排序表;

Step5: 按置乱排序表 $\{t_1, t_2, \dots, t_N\}$ 对 i ($i = 1, 2, \dots, N$) 进行置换.

4 新的身份认证协议描述与分析

设 U 为客户端, S 为服务器端, ID_u 为用户标识, N 为最大迭代次数, i 为第 i 迭代次数, pw 为用户口令, $T = \text{table}$ 为置乱排序表, 用 $X // Y$ 表示字段 X 和 Y 的顺序联接, $Seed$ 为服务器在混沌 Hash 函数初值范围内任意选取的初值, h 为混沌 Hash 函数.

用户要想通过客户端访问服务器进行网上购物等活动, 必须首先进行注册成为合法用户, 然后经过身份认证之后才能访问服务器. 新用户第一次登录服务器时, 就必须先进行注册工作. 注册工作由用户和服务器共同参与完成, 协商并保存在认证过程中将要用到的数据, 确保这些数据的机密性.

该协议在注册的过程中首先在服务器端生成一个置乱排序表, 然后使用客户端传送过来的对种子和口令经过 N 次迭代的 Hash 值作为密钥加密置乱排序表, 并传送给客户端, 客户端用同样的密钥解

密. 协议中在网上发送的 i 并不是实际的迭代次数, 实际的迭代次数 t_i 是通过置乱排序表进行代换后得到的, 置乱排序表是以密文形式发送的, 所以攻击者没有办法窃听到真正的迭代次数. 这样不仅可以抵御小数攻击并且减小了内部人员攻击的威胁. 本协议中采用的是对称加密算法对置乱排序表进行加密, 要比公钥算法节省运算时间, 执行速度比较快, 满足了电子商务系统实现过程安全快速的需求.

4.1 用户注册步骤

① 用户在客户端发送一个用户的身份标识 IDu 和口令 pw 给服务器 S 提交注册请求.

② 服务器 S 在收到注册请求后, 先检查认证数据库是否存在该用户 IDu, 若其中已有该 IDu, 则它是不合法的, 要求用户重新选择. 若认证数据库中不存在该用户 IDu, 则该 IDu 为合法的用户身份标识, 服务器为该 IDu 选择随机种子 Seed 和最大迭代次数 N , 然后发送给用户.

③ 用户收到 Seed 和 N 后, 使用客户端计算程序(与服务器端使用相同的混沌 Hash 函数算法)计算 $h^N = h_N(\text{Seed} // \text{pw})$ 即对 Seed//pw 进行 N 次 Hash 运算, 同时选择一个随机数 R_1 计算 $h(R_1)$, 并将计算结果 h^N 与 $h(R_1)$ 发送给认证服务器 S.

④ 服务器收到 h_N 与 $h(R_1)$ 后, 将 h_N 和 $h(R_1)$ 与对应的用户 IDu 关联起来, 秘密保存在认证数据库中, 同时将最大迭代次数减 1 后保存, 即保存 $N-1$, 然后用服务器端置乱程序生成置乱排序表 $T = \text{table}$, 用 h_N 作为密钥加密 table, 得到 $C = E_{h_N}(T)$, 再把 C 发送给客户端.

⑤ 用户收到 C 后用 h_N 解密 C 得到 T , 保存在客户端. 用户注册结束之后, 只有用户与服务器知道 h_N .

以上步骤就完成了用户的注册工作, 用户只需记住其登录 IDu、口令 pw、随机数 R_1 , 以及在客户端保存的置乱排序表 T 即可.

4.2 用户登录步骤

① 用户每次登录都要进行身份认证, 如果是第一次, 用户首先将自身的身份标识 IDu 发送给服务器, 服务器收到登录请求和用户 IDu 后, 在认证数据库中查询该 IDu 是否存在. 若该 IDu 不存在, 则拒绝此次请求; 若该 IDu 存在, 则从数据库中取出相对应的种子 Seed 和迭代次数 $N-1$ 以及 $h(R_1)$, 并将这三个数据用 h_N 加密后得到 $C' = E_{h_N}(\text{Seed} // N-1 // h(R_1))$, 将其传送给用户的客户端.

② 客户端收到 C' 后首先用 h_N 解密, 得到种子

数 Seed、迭代次数 $N-1$ 以及 $h(R_1)$, 然后使用客户端存储的 R_1 计算 $h'(R_1)$ (对 R_1 进行一次 Hash 运算), 比较 $h(R_1)$ 与 $h'(R_1)$ 是否相同, 如两者不相同, 服务器的身份没有通过认证, 用户将中止此次登录; 如两者相同, 则服务器的身份得到认证, 用户继续登录步骤.

③ 用户查找客户端的置乱排序表 T , 找到 $N-1$ 对应的 t_{N-1} , 用与服务器端相同的计算程序计算 $h_{N-1} = h^{t_{N-1}}(\text{Seed} // \text{pw})$ (用 Hash 函数进行 t_{N-1} 次迭代) 并保存在客户端, 作为后续登录时的加密密钥, 同时选择一个随机数 R_2 计算 $h(R_2)$, 使用 h_N 作为密钥对 h_{N-1} 和 $h(R_2)$ 进行加密, 得到 $C_1 = E_{h_N}(h_{N-1} // h(R_2))$, 然后将计算结果 C_1 发送给服务器. 此时客户端保留 h_{N-1} 作为对称加密的密钥对后续登录数据加密, 然后删除 R_1 保存 R_2 .

④ 服务器收到数据 C_1 后, 首先用 h_N 作为解密密钥对 C_1 进行解密, 得到 h_{N-1} 和 $h(R_2)$, 然后服务器端的计算程序计算 $h'_N = h^{N-t_{N-1}}(h^{t_{N-1}}(\text{Seed} // \text{pw}))$, 验证 h_N 与 h'_N 是否相同, 若相同, 则认证通过, 用户成功登录, 把 h_{N-1} 和 $h(R_2)$ 存入认证数据库中并与 IDu 关联起来, 并把 $h(R_1)$ 从认证数据库中删除, h_{N-1} 作为后续登录的服务器端解密密钥, 即把 h_{N-1} 作为后续登录时客户端与服务器端的对称密钥; 如果 h_N 与 h'_N 不相同, 则认证失败, 服务器拒绝用户的登录请求. 若服务器通过了对用户的身份认证, 将迭代次数减 1 后保存在认证数据库中, 以便下一次认证时使用.

⑤ 一般地, 当已注册的用户 U 向服务器发送登录请求时, 如果是第 i 次登录, 用户首先将用户 IDu 发送给服务器, 服务器收到登录请求和用户 IDu 后, 在认证数据库中查询该 IDu 是否存在. 若 IDu 不存在, 则拒绝此次请求; 若 IDu 存在, 则从数据库中取出相对应的种子 Seed 和迭代次数 $N-i$ 以及 $h(R_i)$, 并将这三个数据用 h_{N-1} 加密后得到 $C = E_{h_{N-1}}(\text{Seed} // N-i // h(R_i))$, 并传送给用户的客户端.

⑥ 客户端收到加密的数据 C 后用 h_{N-1} 进行解密, 得到种子数 Seed、迭代次数 $N-i$ 以及 $h(R_i)$, 首先使用客户端存储的 R_i 计算 $h'(R_i)$, 比较 $h(R_i)$ 与 $h'(R_i)$ 是否相同, 如两者不相同, 服务器的身份没有通过认证, 用户将中止此次登录; 如两者相同, 则服务器的身份得到认证, 用户继续登录步骤.

⑦ 用户查找客户端的置乱排序表 T , 找到 $N-i$

i 对应的 t_{N-i} , 用与服务器端相同的计算程序计算 $h_{N-i} = h^{t_{N-i}}(\text{Seed} // \text{pw})$ (用 Hash 函数进行 t_{N-i} 次迭代), 同时选择一个随机数 R_{i+1} 计算 $h(R_{i+1})$, 此时客户端保留 R_{i+1} 然后删除 R_i , 使用 h_{N-1} 作为密钥对 h_{N-i} 和 $h(R_{i+1})$ 进行加密, 得到 $C_i = E_{h_{N-1}}(h_{N-i} // h(R_{i+1}))$, 然后将计算结果 C_i 发送给服务器。

⑧ 服务器收到数据 C_i 后, 首先用 h_{N-1} 作为解密密钥对 C_i 进行解密, 得到 h_{N-i} 和 $h(R_{i+1})$, 然后用服务器端的计算程序计算 $h'_N = h^{N-t_{N-i}}(h^{t_{N-i}}(\text{Seed} // \text{pw}))$, 验证 h'_N 与认证数据库里的 h_N 是否相同, 若相同, 则认证通过, 用户成功登录, 把 $h(R_{i+1})$ 存入认证数据库; 否则, 认证失败, 服务器拒绝用户的登录请求。若服务器通过了对用户的身份认证, 将迭代次数减 1 后保存, 以便下一次认证使用, 并且不再保存上一次认证的 Hash 值和 $h(R_i)$, 登录过程结束。

4.3 协议 COTP 的安全性分析

① 该协议能够抵御重放攻击、小数攻击及内部人员攻击。

② 该协议满足了需传递信息的机密性要求, 即在网上传输的迭代次数 i 和种子 Seed 不再以明文传输, 而是以经过对称密钥加密处理的密文形式传递, 做到了传递信息的机密性, 增加了协议的安全性。

③ 该协议能够实现服务器与客户的双向认证。该协议不仅实现了服务器对用户的身份认证, 也实现了用户对服务器的身份认证, 这样攻击者不能进行假冒服务器进行攻击。

④ 该协议能够实现密钥的交换, 该协议具有协商与交换客户与服务器端通信密钥的功能, 在第一次登录时产生了双方的通信密钥, 因而用户与服务器间传递的信息可以用双方共享的会话密钥进行加密, 这样可以保证数据的安全传输。

5 结束语

分析了 S/KEY 口令认证协议存在的安全问

题, 并且基于身份认证协议的基本理论和混沌理论, 提出了改进的一次性口令认证协议 COTP。改进的协议中引进了置乱排序算法和混沌 Hash 函数, 所以能够抵御重放攻击, 还能抵御小数攻击及内部人员攻击, 并且能够防止攻击者进行截获种子的攻击。该协议不仅实现了服务器对用户的身份认证, 也实现了用户对服务器的身份认证, 这样攻击者不能进行假冒服务器进行攻击。另外该协议还具有协商与交换对称密钥的功能, 在第一次登录时产生了对称密钥, 因而用户与服务器间传递的信息可以用双方共享的用对称密钥进行加密, 这样可以保证数据的安全传输。该协议中采用对称加密算法, 因而运行速度快、效率高, 适合电子商务安全快速的需求。

参考文献:

- [1] Leslie Lamport. 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[J]. Internet Society Symposium on Network and Distributed System Security, 1994(2): 151 - 158.
- [3] 王小云, 冯登国, 于秀源. HAVAL - 128 的碰撞攻击[J]. 中国科学 E 辑信息科学, 2005, 35(4): 405 - 416.
- [4] Wang XY, Yu HB. How to Break MD5 and other hash functions[C]// Proc. of the Eurocrypt 2005. Berlin, Heidelberg: Springer - Verlag, LNCS 3494, 2005: 19 - 35.
- [5] 张利华. 基于随机数和 Hash 函数的认证方案[J]. 微电子学与计算机, 2007, 24(6): 80 - 83.
- [6] 肖广顺, 左春荣. 网格身份认证和访问控制的研究[J]. 微电子学与计算机, 2008, 25(6): 175 - 181.
- [7] 姜楠, 杨德礼, 鲍明宇, 等. 基于二维混沌系统的 Hash 函数构造算法[J]. 北京师范大学学报: 自然科学版, 2008, 44(4): 371 - 375.
- [8] 刘向东, 焉德军, 朱志良, 等. 基于排序变换的混沌图像置乱算法[J]. 中国图象图形学报, 2005, 10(5): 656 - 660.

作者简介:

姜楠女, (1964 -), 博士, 副教授. 研究方向为信息安全。