

文章编号:1006-3080(2009)05-0740-06

## 身份认证方案的安全性分析

黄建华, 田昌勇, 宋国新

(华东理工大学计算机科学与工程系, 上海 200237)

**摘要:**针对入侵容忍身份认证方案的安全性进行了详细分析,并用状态转移图描述了系统的安全行为。该方案的特点是基于 Shamir 秘密共享算法将用户密码分片后存储在多个认证服务器中,使得少数服务器受到入侵时仍能继续提供正确的认证服务,且在认证身份时不需要重构用户原来的密码数据,提高了认证系统的可用性、完整性和机密性。

**关键词:**认证; 入侵容忍; 秘密共享; 安全分析

中图分类号:TP393

文献标志码:A

## Security Analysis of An Authentication Scheme

HUANG Jian-hua, TIAN Chang-yong, SONG Guo-xin

(Department of Computer Science and Engineering, East China University of  
Science and Technology, Shanghai 200237, China)

**Abstract:** This paper gives a detailed analysis on the security of an authentication scheme with intrusion-tolerant feature. A state transition diagram is used to describe the security behavior of the system. The characteristics of the proposed scheme are that a user password is split to store in distributed shared servers by using Shamir's secret sharing. Thus, valid authentication services are continuously available even though the minority of shared servers are compromised. Moreover, the original password data is not required to be constructed during authentication processes. Hence, the availability, integrity and confidentiality of authentication system will be enhanced by means of the present scheme.

**Key words:** authentication; intrusion-tolerant; secret sharing; security analysis

身份认证就是对于一方所声明的身份进行认证的过程。假定 Alice 需要与 Bob 进行加密通信,在通信前需要先确认对方的身份并获得会话密钥。传统的相互认证协议的做法是 Alice 和 Bob 分别与受信的认证服务器 KDC 有一共享密钥  $K_A$  和  $K_B$ ,通过认证协议它们可以获得通信用的会话密钥  $K_S$ 。传统的相互认证协议的实现只使用一个 KDC,一旦 KDC 受到恶意入侵,将导致  $K_A$ 、 $K_B$  和  $K_S$  等密钥信息泄露,此外,KDC 也面临字典式攻击<sup>[1]</sup>、系统内部发起的攻击、单点故障等安全问题。因此,需要寻找一种新的方法来解决这些安全问题。

秘密共享作为一种入侵容忍技术,目前在安全模型的研究中得到了广泛应用。本文首先介绍一个能较好地解决上述安全问题的入侵容忍身份认证方案<sup>[2]</sup>,然后从安全模型的可用性、完整性和机密性等要素方面对该认证方案的安全性进行讨论和分析。

### 1 认证方案

入侵容忍身份认证方案的基本思想是基于 Shamir<sup>[3]</sup>秘密共享算法。将用户的密码信息分成  $n$  片后分别存储在  $n$  个共享认证服务器中,只要不超

过  $t-1$  个服务器遭到入侵危害,攻击者就不能从中还原出用户的密码信息。同时只要有  $t$  个认证服务器正常运行就可以完成认证过程,以此达到入侵容忍的目的。

假设有  $n$  台认证服务器  $AS_i$  共同参与认证数据的存储管理和对用户的认证过程,通信双方为 Alice 和 Bob。认证方案包括用户注册与身份认证两个部分,如图 1 所示。

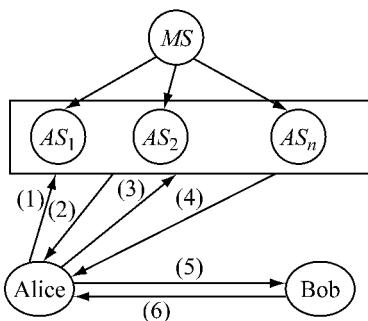


图 1 基于入侵容忍的身份认证方案

Fig. 1 Authentication schema based on intrusion-tolerant

用户注册通过管理服务器 MS 进行。假设输入的用户标识为  $ID$ , 密码为  $p$ , 管理服务器对  $p$  使用  $(t,n)$  Shamir 秘密共享算法产生  $n$  个不同分片密码  $(p_1, p_2, \dots, p_n)$ , 并生成一个随机值  $S$  为种子。随后计算分片密码  $p_i$  的哈希值  $K_{pa_i}$ ,  $K_{pa_i} = H(S, ID, p_i)$ , 其中  $1 \leq i \leq n$ 。最后将值  $(ID, S, K_{pa_i})$  通过管理服务器和认证服务器之间的 SSL 进行连接,发送给认证服务器  $AS_i$  存储。

假设  $ID_A$  表示 Alice 的身份标识,  $ID_B$  表示 Bob 的身份标识,  $S_a$  表示 Alice 的种子,  $S_b$  表示 Bob 的种子,  $K_{pa_i}$  和  $K_{pb_i}$  分别表示 Alice 和 Bob 的第  $i$  片分片密码哈希值,  $K_{ab}$  表示 Alice 和 Bob 的会话密钥,  $AS_i$  表示第  $i$  个认证服务器,  $L$  表示生存期,  $T$  表示时间戳,  $N$  表示随机数,  $N_i$  表示随机数  $N$  的第  $i$  个分片,  $E_{K_{ab}}[\cdot]$  表示用密钥  $K_{ab}$  对消息进行加密, 加密算法为 DES。Alice 与 Bob 确认对方身份并获得会话密钥的认证过程如下:

- (1) Alice →  $AS_i: ID_A \parallel ID_B$
- (2)  $AS_i \rightarrow Alice: S_a \parallel S_b$
- (3) Alice →  $AS_i: E_{K_{pa_i}}[ID_A \parallel ID_B \parallel N_i \parallel V(N)]$
- (4)  $AS_i \rightarrow Alice: E_{K_{pb_i}}[ID_A \parallel T \parallel L \parallel N_i \parallel V(N)] \parallel E_{K_{pa_i}}[ID_B \parallel T \parallel L \parallel N_i]$
- (5) Alice → Bob:  $E_{K_{pb_i}}[ID_A \parallel T \parallel L \parallel N_i \parallel V(N)] \parallel E_{K_{ab}}[ID_A \parallel T] \parallel S_b$

- (6) Bob → Alice:  $E_{K_{ab}}[T+1]$

认证过程的具体解释如下:

(1) Alice 发出访问 Bob 的请求,该请求经代理转发给所有可用的认证服务器。

(2) 认证服务器根据  $ID_A$  和  $ID_B$  查找到 Alice 和 Bob 的种子  $S_a$  和  $S_b$  并返回给 Alice。

(3) Alice 用 Shamir 秘密共享算法对密码  $p$  进行分片以获得  $pa_i$ , 并计算分片密码哈希值  $K_{pa_i}$ ,  $K_{pa_i} = H(S_a, ID_A, pa_i)$ 。同时 Alice 选择一个安全随机数  $N$ , 用 Shamir 算法对  $N$  进行分片, 产生分片随机数  $N_i$ ,  $N_i = f_{t,n}(N, i)$ 。最后 Alice 以  $K_{pa_i}$  作为密钥对包括  $ID_A$ 、 $ID_B$ 、分片随机数  $N_i$  以及验证信息  $V(N)$  的消息进行加密, 然后将加密消息发送给对应的认证服务器  $AS_i$ ,  $V(N)$  的详细内容将在后文阐述。

(4) 认证服务器  $AS_i$  收到 Alice 发送来的消息后, 用存储在服务器端的 Alice 的  $K_{pa_i}$  对加密消息  $E_{K_{pa_i}}[ID_A \parallel ID_B \parallel N_i \parallel V(N)]$  进行解密, 并用消息中 Alice 所请求访问对象 Bob 的  $ID_B$  作为索引, 查找出 Bob 在认证服务器  $AS_i$  中的共享分片  $K_{pb_i}$ , 并用  $K_{pb_i}$  对时间戳  $T$ 、分片随机数  $N_i$ 、生存期  $L$  和校验信息  $V(N)$  进行加密, 生成加密消息  $E_{K_{pb_i}}[ID_A \parallel T \parallel L \parallel N_i \parallel V(N)]$ , 同时用  $K_{pa_i}$  对  $[B \parallel T \parallel L \parallel N_i]$  进行加密生成另一加密消息, 随后将两个消息发送给 Alice。

(5) Alice 产生会话密钥  $K_{ab} = H(ID_A, ID_B, N)$ , 此处  $H()$  同样为单向哈希函数, 用此会话密钥加密  $ID_A$  和时间戳  $E_{K_{ab}}[ID_A \parallel T]$ , 并连同收到的加密消息  $E_{K_{pb_i}}[ID_A \parallel T \parallel L \parallel N_i \parallel V(N)]$  和 Bob 的种子  $S_b$  一并发送给 Bob。

(6) Bob 通过其密码、身份标识  $ID_B$  和种子  $S_b$  生成共享分片密码哈希值  $K_{pb_i}$ , 用  $K_{pb_i}$  对相应的分片加密信息  $E_{K_{pb_i}}[ID_A \parallel T \parallel L \parallel N_i \parallel V(N)]$  进行解密, 并利用 Shamir 秘密共享算法的逆运算, 在校验信息  $V(N)$  的协助下还原随机数  $N$ 。在正确还原出随机数  $N$  后就可计算会话密钥  $K_{ab}$ ,  $K_{ab} = H(ID_A, ID_B, N)$ , 用此会话密钥解密  $E_{K_{ab}}[ID_A \parallel T]$ , 求出  $ID_A$  和  $T$ , 并与另一消息解密出的 Alice 的  $ID_A$  和时间戳  $T$  (用于防止重放攻击) 进行比较, 若相同, 则可确认与 Alice 共享了会话密钥。Bob 的时间戳加 1, 用共享会话密钥对其进行加密, 发送给 Alice。Alice 收到该消息后, 用共享会话密钥对消息进行解密, 若时间戳一致, 则 Alice 就可确定 Bob 的身份, 并确定与 Bob 共享了会话密钥  $K_{ab}$ 。

## 2 安全性分析

计算机网络系统的安全可由一系列指标来衡量,但基本归类为3个主要的指标,即可用性、完整性和机密性。可用性指系统持续提供服务的能力;完整性指系统提供的信息和服务是合法的和可信的;机密性指系统具有保密和掌控信息的能力。通过采用冗余和多样性技术,上述认证方案能够从导致失去可用性、完整性和机密性的安全攻击中恢复正常,达到了入侵容忍的目的。

秘密共享是一种分发、保存、恢复秘密信息的方法,其基本思路是通过算法将用户密码 $p$ 分成 $n$ 片,然后将分片分别存储在不同的服务器,在需要时可以利用任意 $t(t \leq n)$ 个分片数据来重构原密码数据 $p$ ,而少于 $t$ 个的分片则不能完成恢复过程。也就是说即使有 $n-t$ 个服务器被毁坏或被恶意入侵,仍然可以保持机密性,并重构原来的数据 $p$ 。这种方案也称为 $(t,n)$ 门限方案,其中 $t$ 是门限值, $n$ 是分片的数目。设 $q$ 是一个素数, $q > n$ 且 $q > p$ ,可信中心根据Shamir算法给 $n$ 个参与者 $P_1, P_2, \dots, P_n$ 分配共享分片的过程如下:

(1) 可信中心秘密地随机选取 $t-1$ 个元素 $a_1, a_2, \dots, a_{t-1} \in Z_q$ 构成多项式 $a(x) = a_0 + a_1x + \dots + a_{t-1}x^{t-1}$ ,其中常数 $a_0 = p$ 。

(2) 可信中心从 $Z_q$ 中选取 $n$ 个不同的非零元 $x_1, x_2, \dots, x_n, n \leq q-1$ ,计算 $y_i = a(x_i), 1 \leq i \leq n$ 。

(3) 将 $(x_i, y_i)$ 分配给参与者 $P_i, 1 \leq i \leq n, x_i$ 是公开的, $y_i$ 就是秘密的共享分片。

每对 $(x_i, y_i)$ 就是曲线 $a(x)$ 上的一个点。因为 $t$ 个点唯一确定一个 $t-1$ 次多项式 $a(x)$ ,所以 $p$ 可以从 $t$ 个分片重构出来,但少于 $t$ 个分片无法确定 $a(x)$ ,即 $p$ 。上述定义计算 $n$ 个分片的时间复杂度为 $O(nt)$ ,恢复密码 $p$ 的时间复杂度为 $O(t \log_2 t)$ ,且当 $n=2t-1$ 时满足鲁棒性要求。显然,根据上述算法,如果某台服务器被入侵,并不需要重新计算所有的分片,只需要对被入侵的服务器进行重构,重新选取一个 $x_i$ 值,重新计算该台服务器存储的分片 $y_i = a(x_i)$ 即可;同样,改变所有的分片值并不需要用户改变其密码,只要重新选择一个新的多项式即可,这种改变可以增强系统的安全性。门限方案是一种相当灵活的秘密共享方法,改变门限值 $t$ 的大小,可以适当地调整系统安全的可用性和机密性,这里有一个性能、可用性和机密性之间的权衡。较大的 $t$ 意味着需要更多的分片来重构密码,增加了入侵者攻击的难度,因此系统更安全,但代价是较低的性能。

较大的 $t$ 需要更多的 $n$ ,因此增强了系统的生成能力,提高了可用性。选取较小的 $t$ 可以提高性能,但却降低了机密性。如果对 $n$ 和 $t$ 的值作出合适的选择,加之常规的防护措施,该认证方案是很难攻击的。

要提供入侵容忍,系统还依赖于两个基本的操作,即检测入侵和减轻入侵的不利影响。通过在各服务器中部署IDS以及对请求和响应执行接受测试可以发现入侵,从而使得系统可以通过修复、重配等方法来恢复正常。秘密共享方案本身提供了数据和服务器的冗余,使得系统只要有 $t$ 个认证服务器能发送正常的分片就可以完成认证过程;选择不同的服务器程序、不同的操作系统和数据库来实现多样性,减少了因操作系统和应用程序代码的漏洞而危害系统的可能性。以上措施使系统在少数服务器遭受入侵的情况下仍然能持续提供认证服务,实现了系统的可用性目标。

秘密共享方案的一个基本假设是在恢复原密码时,所有受托人都给出自己真实的子密码。但是,如果某个或某些受托人出示虚假的子密码,则在他们自己得到真正的原密码的同时,可以使其他人得到的都是错误的原密码。Tompa<sup>[4]</sup>的研究结果证明,Shamir方案在防欺骗方面是很脆弱的。因此,上述认证过程中设计了避免恢复用户密码的方案,即进行认证时不需要恢复用户原密码,从而有效地避免了攻击者的欺骗行为。另外,认证方案中验证信息 $V(N)$ 也用于判断分片信息的正确性,防止认证服务器欺骗。例如,在 $(t,n)$ 门限方案中,即使 $t-1$ 个认证服务器被入侵,入侵者仍然不能得到Alice和Bob的密码信息以及会话密钥。但是,当 $t-1$ 个服务器发送伪造分片或者错误消息时,Bob收到的任何一组 $t$ 个分片的集合都会还原出一个不同的 $N$ ,在这种情况下,Bob就无法判断出共享分片的合法性。理论上,Alice和Bob可以尝试用所有的 $t$ 个分片的组合进行握手,直到成功为止。但实际上,如果组合数非常大,这种方法将不可行。此外,为了提高协议的效率,并不希望Alice和Bob之间通过多次交换信息才能产生会话密钥。鉴于此,在加密分片中加入了冗余验证信息 $V(N)$ 用于判断分片信息的正确性, $V(N) = (H(N_1), H(N_2), \dots, H(N_n))$ ,其中 $H()$ 是单向哈希函数。正常的认证服务器将返回正确的分片 $N_i$ 以及正确的验证信息。假设收到的加密分片信息中,一半以上都是正确的,则Bob将收到的分片解密后得到 $N_i$ ,通过哈希函数 $H()$ 计算出每一个 $H(N_i)$ ,并与所有冗余验证信息中的

$H(N_i)$ 相比较,如果与大多数  $H(N_i)$  相等,则说明此分片为正确信息。以上措施实现了系统安全的完整性目标。

另外,从静态角度分析,系统中客户端并不存储任何密钥信息,用户仅在登录时输入密码  $p$ ,在对  $p$  进行 Shamir 秘密共享算法分片后,可以将  $p$  立即从内存中擦除,以有效地提高客户端的安全性。认证服务器中存储的是用户的共享分片验证数据。分片验证数据是将用户密码  $p$  通过 Shamir 秘密共享算法分片并经过单向哈希函数作用后的哈希值,由于单向哈希函数的不可逆性,无法从哈希值逆向恢复出用户密码分片,因此在认证服务器受到入侵时,直接获取用户密码的威胁就减少了。但是仅仅依靠单向哈希函数加密的用户密码表依然十分脆弱,不能有效地防范共享字典式攻击。因此在单向函数中将种子  $S$  与用户密码连接来计算哈希值。由于  $S$  是随机产生的,所以攻击者不得不产生每个可能的种子值的单向哈希值,进行初始化矢量的简单尝试,因此实际上消除了对常用密码的字典式攻击。虽然  $S$  能防止一般的字典式攻击,但是不能防止对单个密码的一致攻击。由于认证服务器所存储的是分片密码和  $S$  连接的哈希值,即使采用字典式攻击,并成功获取了用户的分片密码,但是仅仅依靠单一的分片密码并不能重构出用户的密码。换句话说,即使攻击者对认证服务器采用字典式攻击取得了少数的分片密码,但是依然不能从这些有限的信息中恢复出用户的密码。

从动态角度分析,在整个方案的认证过程中,无论在认证服务器还是在代理服务器,都没有对用户密码进行重构。因为一旦在任何位置对用户密码重构,则该重构点就面临安全威胁,重构点受到入侵或攻击,无疑将暴露用户的密码信息,对认证系统的安全性造成威胁。在认证方案的步骤(3)中,若 Alice 为非法用户,则她将无法推导出正确的分片密码,若认证服务器以系统内存储的分片数据作为密钥通过 DES 逆运算不能推导出相应的正确信息,则可判断 Alice 为非法用户,拒绝提供认证服务。在步骤(4)中认证服务器的身份同样可以得到 Alice 的认证,伪装的认证服务器因为不知道 Alice 的正确分片密码哈希值,因此不能推导出 Alice 发过来的原文,从而不可能正确生成回送 Alice 的加密消息  $E_{K_{pa_i}}[ID_B \parallel T \parallel L \parallel N_i]$ ,从而 Alice 可以通过检验从各个认证服务器回送的加密消息来认证服务器的身份,通过步骤(3)和步骤(4),Alice 和认证服务器间的身份相互得到了认证。在步骤(6)中,Bob 可以

对 Alice 的身份进行认证,通过对 Alice 转发的认证服务器生成的消息,Bob 可以推导出分片随机数  $N_i$ ,借助校验信息  $V(N)$ ,通过对  $N_i$  进行重构,可以推导出随机数  $N$ ,再通过哈希函数,Bob 就可以求出 Alice 和 Bob 所共享的会话密钥  $K_{ab}$ ,用  $K_{ab}$  可以解密 Alice 加密的另一消息  $E_{K_{ab}}[ID_A \parallel T]$ ,求出时间戳  $T$ 。通过对另一消息  $E_{K_{pb_i}}[ID_A \parallel T \parallel L \parallel N_i \parallel V(N)]$  解密出的 Alice 的  $ID_A$  和时间戳  $T$  进行比较,若两者相等,则可以确定 Alice 的身份。因为 Alice 的身份在得到认证前无法推导出时间戳  $T$ ,这样就间接地对 Alice 的身份进行了认证。此外,时间戳  $T$  和生存期  $L$  的作用是防止攻击者截取到数据包后进行重放攻击。假定整个系统的时钟是同步的,则 Bob 通过对时间戳  $T$  的判断就可以检验此加密消息的即时性,有效地防止了重放攻击。生存期  $L$  可以动态调节,这样在生存期内,Alice 和 Bob 进行通信就不必重复认证,有效地减少了 Alice 和认证服务器间的通信量,降低了认证服务器的负载,提高了认证过程的效率。最后,Bob 的身份得到了 Alice 的认证,因为若 Bob 为非法用户,则将无法推导出会话密钥及时间戳,就不能产生正确的加密消息。通过以上步骤,Alice 和认证服务器、Bob 和认证服务器、Alice 和 Bob 之间的身份都间接得到了认证,并且 Alice 和 Bob 之间共享了会话密钥。从整个认证过程来看,在任何位置都没有对双方的密码进行重构,且攻击者无法通过对数据包的截取分析推导出任何用户的相互认证密钥信息,同时,时间戳  $T$  和生存期  $L$  有效地防止了重放攻击。会话密钥为 Alice 和 Bob 在相互认证后的通讯进行加密,即使会话密钥泄密,仅仅在其生存期内威胁当前会话,但并不会造成用户密码信息的泄密。

### 3 认证方案的状态转移图

状态转移图可以用来描述入侵容忍系统的安全行为。图 2 是本文描述的入侵容忍认证系统的状态转移图,该图描述了在威胁等级不断变化的环境中系统所进行的操作。

系统最初处于正常状态 G,在该状态下,系统未发现安全漏洞。通常有 3 种基本方法用于发掘系统的安全漏洞。

(1) 入侵者可以访问易引起系统安全问题的常见漏洞列表。例如,如果入侵者知道用户不严格地选取了密码,就可以用密码窃取法来猜测用户密码,从而使系统不能抵御基于密码的攻击。

(2) 入侵者可以主动探测系统中未知的漏洞。例如,入侵者可以设法发现系统内核或应用程序是否实施了缓存溢出保护方案,缓存溢出的弱点常被用于设置陷阱或建立其他的后门。

(3) 可能出现不满的雇员改变或毁坏某项信息的情况。

基于这些原因,系统将从理想状态 G 进入存在潜在安全隐患的状态 V。状态 V 表明系统存在某些安全漏洞,入侵者可能会利用这些漏洞未经授权读取或修改信息,未经授权访问系统资源或发动 DoS 攻击。在此期间,系统管理员也可能定位这些漏洞并采取措施堵上漏洞。如果在攻击者发动攻击前堵上了所有的漏洞,系统将返回到状态 G,否则,入侵者将利用这些漏洞成功侵入系统。系统遭到攻击后,内部的 IDS 模块在系统遭受危害之前可能检测到攻击,在此情况下,系统将继续处于 V 状态,否则侵入系统将对认证服务器造成某些损害。如果系统的安全机理检测到危害,且系统的冗余设计可以容忍引起的危害,则攻击后系统仍将回到状态 V。如果恶意攻击只危害了少数认证服务器,且系统能够检测到这些危害,只要被危害的服务器少于  $t$  个,系统通过将状态改变到 V1 可以容忍这些损害并继续提供认证服务。如果系统检测到危害但被危害的服务器达到或超过  $t$  个,则系统将进入失败安全状态 FS。在 FS 状态,系统停止对外服务以阻止返回错误的响应,防止危害进一步扩散。

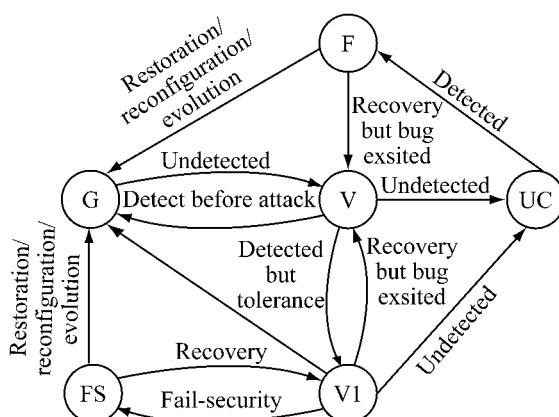


图 2 认证系统状态转移图

Fig. 2 State transition diagram of authentication schema

如果在返回错误的响应前,系统的所有安全机理都检测不到损害,则系统处于不可检测损害状态 UC。在此状态下系统的行为是不可预测的,系统服务得不到保障,这是最不愿看到的情况。但实际上通过审计控制模块观察审计记录中的异常或通过人工监督还是可以检测到损害,系统将进入失败状态

F。在此状态下,系统将经历大量的诊断和恢复工作。系统处于 FS 状态时也将经历少量的恢复和重配工作,以便恢复到正常的工作状态。在恢复状态,通过升级软件、替换硬件和修改系统配置可以排除安全漏洞,当所有的漏洞都被堵上后,系统回到 G 状态。状态转移图描述了入侵容忍系统在抵御入侵时可能所处的状态。为进一步采取自适应响应策略提供了依据。

## 4 实现与结果

按以上方案,本文实现了一个原型系统,程序采用 Java 语言实现。测试环境由若干 PC 组成,代理服务器程序和认证服务器程序被分别部署在 Window 2003 Server, FreeBSD, Linux 等不同平台。认证系统所需的 Java 环境为 J2SE1.4.2 版,认证系统通过 100 Mbps 的局域网连接。数据库管理系统包括 SQL Server 2000SP4, MySql4.1.12, 及 Oracle10g。由于 Java 出色的跨平台性,操作系统及数据库管理系统的异构性并不影响程序的正常运行,且由于操作系统及数据库管理系统的多样性,有效地提高了身份认证系统整体的安全性。本系统对客户端的操作系统环境并无特定要求,但由于客户端参与了验证过程的计算,因此对客户端的计算能力有一定要求。从成本、性能和实际应用考虑,门限值大于 3 的方案需要使用更多的认证服务器,而且身份验证所花费的时间也比较长,在实际应用中并不合适,而且也是没有必要的。在(3,5)的门限方案下系统最多可以容忍 2 台服务器被入侵,且系统还能提供正常的认证服务,同时,入侵容忍的自适应机制可以发现和修复异常的服务器,使系统恢复正常状态。所以本测试用 Alice 作为用户名,密码为 a1b2c3,在(2,3)及(3,5)门限方案下进行测试。测试内容为可靠性、可用性和性能测试 3 个方面,测试结果见表 1。

表 1 性能测试

Table 1 Performance test

$(t, n)$	Proof time/s
(2,3)	2.78
(3,5)	4.13

(1) 可靠性测试。测试方案是在认证系统的管理服务器中生成了用户的帐号及密码,并进行分片,然后分发给各认证服务器。用户密码的范围包括仅包含数字密码、仅包含字符密码以及数字字符混合

密码3种形式。随机抽取用户帐号在(2,3)及(3,5)门限方案下进行测试。测试结果表明:无论是(2,3)门限方案,还是(3,5)门限方案,认证系统对随机抽取出的帐号均能提供正确、可靠的身份认证服务。从测试结果中可以看出,本文所提出的认证方案是正确的、可靠的。

(2) 可用性测试。可用性测试方案也是在(2,3),(3,5)门限方案下进行,结果和 Shamir 秘密共享方案预期的一致,只要有满足门限值  $t$  台协同认证服务器正常工作,系统就可以提供正确可靠的认证服务。

(3) 性能测试。门限方案和加密技术的应用增加了身份认证的计算量,即增加认证的时间开销。表1的数据说明了在两种门限方案下完成认证所需要的时间。显然  $t$  越大,认证耗时越多。虽然较大的  $t$  增加了认证所需的时间,但完成认证过程所需的时间对于使用者的心理等待时间来说是可以接受的,说明系统设计方案在满足系统正常性能要求的同时,增强了认证系统的安全性和生存能力,达到了预期的目标。

认证系统是系统的安全瓶颈,同时也是系统性能瓶颈,因此理想的认证系统应该同时具有高度的可用性、完整性和机密性。通过将冗余和多样性引

入身份认证系统的设计,使得身份认证系统实现了入侵容忍的目标,并具有以下特点:

(1) 可用性:少数服务器受到攻击或出现故障时,认证系统仍能继续对合法用户提供安全的认证服务。

(2) 完整性:认证过程中无需对用户密码进行还原,并可以验证分片合法性,防止入侵者的欺骗行为。

(3) 机密性:用户的密码采用( $t, n$ )门限方案分片存储在共享认证服务器中,并采用单向哈希函数和 S 种子进行加密保护层,少数服务器受到攻击,并不会泄漏用户的完整密码信息。

## 参考文献:

- [1] 黄建华,程晓松,宋国新. 具有入侵容忍特性的身份认证系统[J]. 计算机工程,2006,32(18):157-159.
- [2] Wu Thomas. A real-world analysis of kerberos password security[C]// Proceedings of Network and Distributed System Security Symposium. San Diego, Calif: ISOC, 1999:3-5.
- [3] Adi Shamir. How to share a secret[J]. Commun ACM, 1979, 22(11): 612-613.
- [4] Tompa M, Woll H. How to share a secret with cheater[J]. Journal of Cryptology, 1988, 1: 133-139.