第28卷第5期 北京交通大学学报 Vol.28 No.5
2004年10月 JOURNAL OF BEIJING JIAOTONG UNIVERSITY Oct. 2004

文章编号:1673-0291(2004)05-0006-05

一种扩展应用边界安全设备认证能力的方法

彭双和[1],韩 臻[1],沈昌祥[2]
(1.北京交通大学 计算机与信息技术学院,北京 100044;2.海军计算技术研究所,北京 100841)

摘 要:为了保护内部网络的安全,必须设置应用边界安全设备.本文采取双服务器认证的方式,提出了一种扩展应用边界安全设备的认证方法,解决了一般的应用边界安全设备(如Socksv5服务器)使用用户名/口令字认证易受被动攻击等问题.新提出的用户名/口令字认证方式在安全性上比原有的方式具有更高的抗攻击的能力,而且用户还能自由地更改应用边界安全设备上自己的口令字.实验结果表明,效率上与原有的用户名/口令字认证方式基本保持一致.

关键词:应用边界安全设备;身份认证;双服务器认证;杂凑函数
中图分类号:TP309 **文献标识码**:B

A Way to Extend the Authentication Capability of Application Boundary Security Device

PENG Shuang-he[1], *HAN-Zhen*[1], *SHEN Chang-xiang*[2]
(1. School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China;
2. Institute of Naval Computer Technology, Beijing 100841, China)

Abstract: In order to improve the security of Intranet, application boundary security devices must be set. A scheme to extend authentication capacity at the application boundary security device is proposed here by using double servers, which solves the problems such as passive attack, etc. That is existed in the current password-based authentication schemes at the application boundary security devices like Socksv5 servers. Considering safety, the proposed scheme here has more power to resist against attacks while at the same time passwords stored in authentication database at the application boundary security devices can be freely changed by users. The experiment shows that the proposed scheme retains just the same efficiency as the old one is.

Key words: application boundary security device; authentication; double servers authentication; hash function

互联网信息技术的发展,使得众多企业组织的内部网络开始连接到Internet上.这样企业在实现访问外部世界并与之通信的同时,外部世界也同样可以访问企业内部网络并与之交互.这时为了保证企业组织的信息安全,企业就必须对其重要信息进行保护.为了安全起见,企业开始在该网络和Internet之间插入一个中介系统,竖起一道安全屏障.这道屏障的作用是提供扼守本网络的安全和审计的唯一关卡,可阻断来自外部通过网络对本网络的威胁和入侵,同时还能管理内部主体访问外界服务的权限.应用边界安全设备[1](如VPN安全网关等)便是能提供上述服务的一种设备.

应用边界安全设备为主体提供服务时必须验证主体的身份,以确保只有经过授权的合法主体才能

收稿日期:2004-02-24
基金项目:国家"863"计划项目(2002AA144020);国家"973"计划项目(TG1999035801)
作者简介:彭双和(1974—),女,湖南衡阳人,博士生.**email**:shhpeng@sohu.com
沈昌祥(1946—),男,浙江宁波人,院士,博士生导师.

得到相关服务.应用边界安全设备(比如Socksv5[2])现只支持用户名/口令字[3]和Kerberos[4]两种认证方法.尽管基于Kerberos认证的安全性较高,但配置起来有一定的难度.用户名/口令字认证方法与Unix操作系统的认证方法类似.这种认证方法简单,配置容易,但使用该方法对主体进行认证时,主体的口令字是以明文的形式在网上传输的,这样主体的口令字很容易被网络窃听者窃取,安全性很低.同时,在应用边界安全设备上创建主体数据库、设置主体口令字的方式对主体带来了不便,主体不能自由地更改应用边界安全设备上自己的口令字.而且,当主体想跨域访问另一个域中的资源时,如何实现安全的跨域间的信息资源访问,让客体域中的应用边界安全设备对主体域中的主体进行审计;如何使主体的口令字不容易受网络窃听者窃取;如何增强应用边界安全设备的认证能力;是本文论述的问题.

1　安全设备认证能力的方法

应用边界安全设备的用法如图1所示.

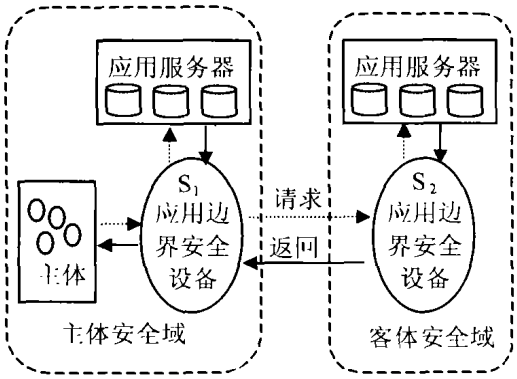

图1　基于应用边界安全设备域间信息资源的安全访问

Fig.1　Access to information resource between realms based on application security boundary device

主体安全域中的主体想访问资源时,首先向本地的应用边界安全设备发出资源访问请求,本地的应用边界安全设备对用户的资源请求进行审查.若判断出主体请求的是本地的资源,则对主体认证后,检索本地的相关资源返回给主体.若请求的是远程的资源,则需联合远程的应用边界安全设备对主体的访问请求进行审查.若允许,则转发本地主体的资源访问请求至远程的应用边界安全设备,最后由远程的应用边界安全设备获取相关资源后,返回给主体域中的主体.

文献[5]中提出了一种使用两台服务器增强用户名/口令字认证的安全方法,其方法为:利用两台服务器(一台命名为A,另一台命名为B)实现对主体的认证.

协议分两个阶段:注册阶段和认证阶段.在注册阶段,主体在A上的注册信息为$P_A=f(P)+R$(其中R为随机数,f为单向函数,P为主体的口令字);主体在B上的注册信息为$P_B=R$.认证阶段,主体向A发送$P_A'=f(P')+R'$,其中R'为主体登录A时输入的口令字,R'为主体登录A时选取的随机数;主体向B发送$P_B'=R'$,要验证主体是否合法,A计算$Q_A=P_A-P_A'=(f(P)-f(P'))+(R-R')$,B计算$Q_B=P_B-P_B'=(R-R')$,若主体登录时输入的口令字$P'$与注册时的口令字$P$一致,则有$Q_A=Q_B$.表示主体认证通过,否则认证失败.

上述方法具有简单、实用的优点.但上述的方法不适合两台应用边界安全设备对用户的认证环境.存在的问题是:①上述协议运行的前提条件是A与B之间的信息传输是通过安全通道进行的.在两台应用边界安全设备的环境下,这种安全通道在认证前还未建立.因此,若有网络窃听者窃听到了信息R'和$(R-R')$,则能计算出主体在B上的注册信息R.这样就会存在极大的安全问题.②主体不能直接跟远程的应用边界安全设备进行信息的交互.主体与远程的应用边界安全设备的信息传输只能经由本地的应用边界安全设备进行转发.为此,本文作者在文献[5]的基础上对上述的协议进行了修改,提出了一种适合两台应用边界安全设备环境下对主体进行认证,并且主体能方便、安全地修改自己口令字的高效、安全的实时认证方案.

1.1　有关的概念和记号

为讨论方便,引入了如下的记号:$U=\{s \parallel |s|=512, s$由0或1组成的比特串$\}$,即$U$为长度为512比特的串集合.$V=\{s \parallel |s|=128, s$由0或1组成的比特串$\}$,即$V$为长度为128比特的串集合.定义$V$上的3种操作:+(加)、-(减)、⊕(异或).其中+、-操作把128比特的串看成16字节的大整数进行运算,⊕操作把128比特的串进行逐位异或运算.$g(\cdot), h(\cdot): \{0,1\}^* \rightarrow V$单向杂凑函数,表示由任意长度的串映射成长度为128比特的串.N为客体安全域中应用服务器的个数.$T_i(i=1,2,\cdots,N)$为客体安全域中的第i台应用服务器.S_1为主体安全域中的应用边界安全设备.S_2为客体安全域中的应用边界安全设备.$\in_R$为表示从集合中任意取元素.D_j^i为T_i上主体j的标识,$D_j^i \in_R V$.K_j为主体j的口令字.R_j^i、$R_j^i \in_R U$为T_i上主体j选取的长度为512比特的随机数.

定义　$P_{i,j}=h(D_j^i, K_j)+h(R_j^i)$.

1.2 方法实施

该方案共分为3个阶段:主体注册阶段、主体认证与信息传输阶段、主体口令字更改阶段.

1.2.1 主体注册

主体若想访问远程域中的资源时,需在 S_2 和 S_1 上进行注册.注册时,不能在任何应用边界安全设备上泄漏自己的口令字信息.

当第 j 个主体想访问客体安全域中的服务时,主体需向客体安全域中的 S_2 进行注册,注册过程为:主体 D_j^i 随机选取一大整数 R_j^i,$R_j^i \in_R U$,经过哈希变换后送给 S_2,S_2 将主体 D_j^i 的注册信息$(D_j^i, h(R_j^i))$存放在主体认证数据库中.

主体除了要向 S_2 注册外,还需向 S_1 进行注册.其注册过程为:主体 D_j^i 选取口令字 K_j(口令字的长度可根据需要而定,这里假设为8字节),利用单向杂凑函数 $h(\cdot)$计算出 $P_{i,j}$,然后将消息$(D_j^i, P_{i,j})$发送给 S_1,S_1 将主体 D_j^i 的注册信息$(D_j^i, P_{i,j})$存放在主体认证数据库中.

1.2.2 主体认证及信息传输

(1)对主体 D_j 的认证过程如图2认证阶段所示,其过程如下.

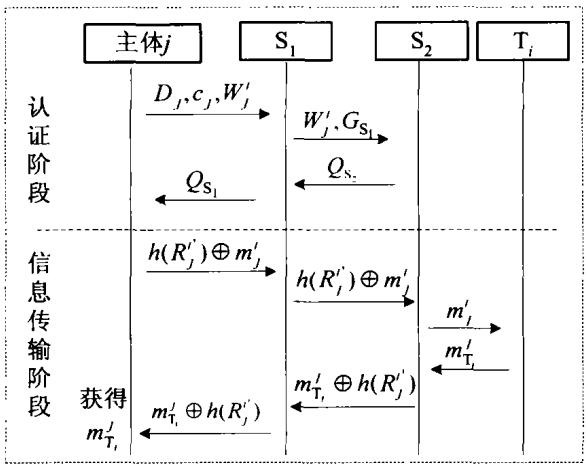

图2 主体认证及信息的传输

Fig.2 User authentication and information transfer

①主体 D_j 选取大整数 $R_j^{i'}$,$R_j^{i'} \in_R U$,$R_j^{i'} \neq R_j^i$,D_j 在自己的主机上输入登录口令字 K_j',并计算

$$c_j = h(D_j, K_j') \oplus h(R_j^{i'}),$$

$$W_j^i = h(R_j^{i'} \oplus h(R_j^i)),$$

向 S_1 发送消息:(D_j, c_j, W_j^i).

②S_1 接收到主体 D_j 发来的消息后,根据从 D_j 主体认证数据库中检索出 $P_{i,j}$ 后,计算

$$Q_{S_1} = [h(D_j, K_j') + h(R_j^{i'})] - P_{i,j} =$$

$$[h(D_j, K_j') - h(D_j, K_j)] +$$

$$[h(R_j^i) - h(R_j^i)],$$

$$G_{S_1} = g(Q_{S_1}),$$

并向 S_2 发送消息(W_j^i, G_{S_1}).

③S_2 接收到(W_j^i, G_{S_1})后计算

$$Q_{S_2} = W_j^i \oplus h(R_j^i) - h(R_j^i) =$$

$$[h(R_j^{i'}) \oplus h(R_j^i)) \oplus h(R_j^i) - h(R_j^i) =$$

$$(h(R_j^{i'}) - h(R_j^i)),$$

$$G_{S_2} = g(Q_{S_2}),$$

并判断从 S_1 接收到的 G_{S_1} 与计算出的 G_{S_2} 是否相等.若相等,则表示 S_1 通过了 S_2 的认证(因为只有 S_1 才能计算出 G_{S_1}),并向 S_1 返回消息 Q_{S_2}.

④S_1 收到 Q_{S_2} 后,将 Q_{S_2} 和 Q_{S_1} 进行比较,即判断 $Q_{S_2} = Q_{S_1}$,若相等,则表明 S_2 拥有 $h(R_j^i)$并能计算出 $h(R_j^{i'})$,这样,才能计算出 Q_{S_2},于是 S_2 通过了 S_1 的认证.同时,由知 $Q_{S_2} = Q_{S_1}$,知 $h(D_j, K_j') = h(D_j, K_j)$,即 $K_j' = K_j$,主体 D_j 登录时输入了正确的口令字,于是 S_1 认为主体 D_i 认证通过,向主体 D_j 返回消息 Q_{S_1};若 $Q_{S_1} \neq Q_{S_2}$,则主体 D_j 登录时输入了不正确的口令字,即 $K_j' \neq K_j$,主体认证 D_j 失败.

⑤若主体认证成功,从 S_1 接收到消息 Q_{S_1},主体将 Q_{S_1} 与 $h(R_j^{i'}) - h(R_j^i)$进行比较,若相等,则表明 S_1 通过了主体端的认证,即 S_1 不是冒充的.因为,只有本地应用边界安全设备才能计算出 Q_{S_1}.

(2)信息传输过程如图2中信息传输阶段所示,步骤如下.

①主体 j 将认证阶段产生的随机数 $R_j^{i'}$ 经哈希变换后当成密钥,将要发送的请求 m_j^i 变换成 $h(R_j^{i'}) \oplus m_j^i$ 后发送给 S_1.

②S_1 将消息 $h(R_j^{i'}) \oplus m_j^i$ 转发给 S_2.

③S_2 根据认证阶段得到的 W_j^i,并根据服务请求的发送者 D_j,从主体认证数据库中检索出 $h(R_j^i)$,进行如下的计算

$$W_j^i \oplus h(R_j^i) = [h(R_j^{i'}) \oplus h(R_j^i)] \oplus h(R_j^i) =$$

$$h(R_j^{i'}),$$

S_2 再将接收到的消息 $h(R_j^{i'}) \oplus m_j^i$ 与上面运算得到的 $h(R_j^{i'})$进行如下的运算

$$h(R_j^{i'}) \oplus m_j^i \oplus h(R_j^{i'}) = m_j^i.$$

④S_2 将解密出的消息 m_j^i 转发给 T_i.T_i 接收到该服务请求后,产生对该服务请求的响应信息 $m_{T_i}^j$,并将该响应信息 $m_{T_i}^j$ 经由 S_2 向主体 j 返回应答.为保证信息传输的机密性,响应消息 $m_{T_i}^j$ 经过 S_2 时经

过变换,以 $m^i_{T_i} \oplus h(R_j^{i'})$ 的形式往外进行转发给 S_1.即 S_2 向 S_1 发送消息 $m^i_{T_i} \oplus h(R_j^{i'})$.

⑤S_1 向主体 j 转发消息 $m^i_{T_i} \oplus h(R_j^{i'})$.

⑥主体 j 接收到消息 $m^i_{T_i} \oplus h(R_j^{i'})$后进行运算

$$m^i_{T_i} \oplus h(R_j^{i'}) \oplus h(R_j^{i'}) = m^i_{T_i},$$

于是得到从 T_i 返回的对服务请求 m_j^i 的响应 $m^i_{T_i}$.

1.2.3　主体口令字更改

若第 j 个主体想更改自己的口令字,必须首先通过 S_1 的认证,只有合法的主体才能更改自己的口令字,主体口令字更改协议如图 3 所示.

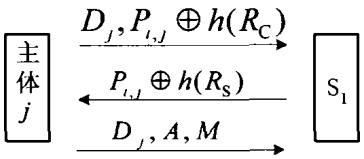

图 3　主体口令更改阶段

Fig. 3　Change user pasword phase

具体步骤如下.

(1)主体 D_j 随机地选取大整数 $R_C, R_C \in_R U$,在自己的主机上输入口令字 K_j,计算出 $P_{i,j}$,并向 S_1 发送消息$(D_j, P_{i,j} \oplus h(R_C))$.

(2)S_1 接收到主体 D_j 发来的消息后,根据 D_j,从主体认证数据库中检索出 $P_{i,j}$后,计算 $P_{i,j} \oplus h(R_C) \oplus P_{i,j} = h(R_C)$,得到 $h(R_C)$,并随机选取另一大整数 $R_S, R_S \in_R U$,向主体 D_j 返回消息 $P_{i,j} \oplus h(R_S)$.

(3)主体 D_j 收到上述的消息后,选取新的口令字 K_j',并计算

$$A = h(P_{i,j}, h(R_C), h(R_S)),$$

$$P_{i,j}' = h(D_j, K_j') + h(R_j^i),$$

$$M = P'_{i,j} \oplus h(P_{i,j}, h(R_C) + 1, h(R_S)),$$

然后向 S_1 发送消息(D_j, A, M).

(4)S_1 收到上述消息后,根据 D_j,从主体认证数据库中检索出 $P_{i,j}$,并计算

$$A' = h(P_{i,j}, h(R_C), h(R_S)),$$

判断 $A' = A$ 是否相等,若不等,则主体 D_j 口令字不正确,认证失败,S_1 拒绝主体修改口令字的请求;若 $A' = A$,则主体 D_j 认证通过,有权修改口令字.S_1 的计算为

$$M \oplus h(P_{i,j}, h(R_C) + 1, h(R_S)) = [P_{i,j}' \oplus h(P_{i,j}), h(R_C) + 1, h(R_S)] \oplus h(P_{i,j}, h(R_C) + 1, h(R_S)) = P_{i,j}.$$

将主体新的注册信息$(D_j, P_{i,j}')$,即$(D_j, h(D_j, K_j') + h(R_j^i))$存放在主体认证数据库中.

1.3　安全性与效率分析

1.3.1　安全性分析

(1)由注册阶段知,主体 D_j 在 S_2 上的注册信息为$(D_j, h(R_j^i))$,在 S_1 上的注册信息为$(D_j, P_{i,j}$.主体的口令字 K_j 没有存放在任何设备上,因此即使远程/本地应用边界安全设备的主体认证数据库受到攻击,也得不出主体的认证口令字 K_j 信息.而且,在对主体进行认证时,主体的口令字也不是以明文方式传输的,而是以 $h(D_j, K_j') + h(R_j^{i'})$的形式在不安全信道上进行传输的.因此,即使网络窃听者截获到了 $h(D_j, K_j') + h(R_j^{i'})$,由于网络窃听者不知道信息 $R_j^{i'}$,所以也很难得出主体的口令字 K_j.若随机数的长度为 512 比特,主体的口令字选取为 8 字节(64 比特),以穷举的方式猜测主体的口令字,则其猜测成功的概率为$\frac{1}{2^{512} \times 2^{64}}$.

(2)主体 D_j 要传输给 S_2 的另一秘密信息 $h(R_j^{i'})$在传输时,也不是以明文的形式进行,而是以 $h(R_j^{i'}) \oplus h(R_j^i)$的方式先传输给 S_1,然后再由 S_1 转交给 S_2 的.由于 S_1 并不知道 $h(R_j^i)$,也很难计算出 $h(R_j^{i'})$,因此,只有 S_2 才能得到信息 $h(R_j^{i'})$.网络窃听者从 S_2 向 S_1 返回的信息 Q_{S_2} 中也不能得到 $h(R_j^{i'})$,因为,网络窃听者无从得到 $h(R_j^i)$.

(3)由主体口令字更改阶段知,主体 D_j 发送给 S_1 的随机数 $h(R_C)$不是以明文的方式在网上传输的,而是以 $P_{i,j} \oplus h(R_C)$的方式传输的,由于网络窃听者不能得知主体 D_j 的$P_{i,j}$,因此 $h(R_C)$的传输能抗被动攻击.S_1 发送给主体 D_j 的随机数$h(R_S)$的分析同 $h(R_C)$.主体 D_j 的新口令字K_j'及$P'_{i,j}$在传送给 S_1 时,也不是以明文的形式进行传输的,而是以 $P_{i,j} \oplus h(P_{i,j}, h(R_C) + 1, h(R_S))$的形式进行传输的,由于只有 S_1 和主体 D_j 才能计算出$h(P_{i,j}, h(R_C) + 1, h(R_S))$,故 $P'_{i,j}$的传输也能抗被动攻击.

(4)在信息传输阶段消息流中,只有掌握会话密钥 $h(R_j^{i'})$的主体端与 S_2 才能正确地进行消息的接收与发送,保证了信息传输的安全性.

综上所述,上述方案是可行的,不但实现了 S_1 与 S_2 的双向认证,而且安全、实时地判断了主体 D_i 是否合法.

1.3.2　效率分析

(1)主体注册阶段,只用到了求杂凑函数值的运算 V 上的加法运算;主体认证和信息传输阶段只用到了求杂凑函数值的运算和 V 上的加法、减法、异或及比较运算;主体口令字更改阶段只用到了求杂

凑函数值的运算和 V 上的加法、异或及比较运算，因此，该认证方法的效率高.

(2)由认证和信息传输阶段可知，本文提出的认证协议与 Socksv5 所用的用户名/口令字认证协议[3]在流程上完全保持一致，只是传输的认证信息字段比原有的 Socksv5 认证信息多一点；信息的传输过程也与标准的 Socksv5 保持一致，只是传输的信息内容由原来的明文变成密文的形式.因此，实现起来，对原有的 Socksv5 协议改动不大，效率较高.

2 实验结果

该方案的实现环境如图 4 所示.操作系统平台为 RedHat Linux 7.3.实验中采用了 4 台 PC 机，其中一台作为 Socksv5 客户端，运行客户端程序，配置一块网卡，网卡速率为 10 Mbps.两台 PC 作为应用边界安全设备，一个为本地的，另一台为远程的.应用边界安全设备运行 Socksv5 服务，配置为双网卡，网卡速率为 10 Mbps.最后一台 PC 作为应用服务器，为客户端的请求提供服务.Socksv5 客户端与本地应用安全边界设备组成主体安全域，远程应用安全边界设备与应用服务器组成客体安全域，主体安全域与客体安全域之间通过 Internet 连接.实验中对客户端、应用边界安全设备的用户数据的管理采用了非关系数据库 gdbm 实现.单向杂凑函数采用 SHA－1[6]算法，信息传输时对信息的加密除可简单地用异或函数实现外，也可采用其它的加密算法，比如使用密钥强度为 128 比特的 IDEA[7]算法作为加密算法.

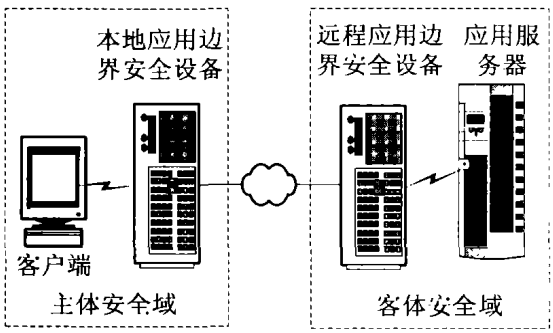

图 4 应用边界安全设备的应用环境

Fig. 4 Application environment for application security boundary device

从测试的实验结果可知，当客户端往应用服务器发送 1 000 个连接请求，每个连接请求传送 10 KB 数据时，Socksv5 标准的用户名/口令字认证方式运行时间为 19.162 s，信息传输的速率为 0.70 Mbps；而本文作者提出的增强的用户名/口令字认证方式运行时间为 20.401 s，信息传输的速率为 0.66 Mbps，其它的参数完全相同.这表明，在效率上，本文提出的用户名/口令字认证方式，同原有的 Socksv5 标准用户名/口令字认证方式具有基本一致的效率.而安全性上比原有的方式具有更高的抗攻击的能力.

3 结束语

文章首次提出了一种扩展应用边界安全设备认证能力的方法，该方法能有效地抗网络被动攻击，并使得不同域间的应用边界安全设备可以利用各自的实体认证数据库对要访问资源的实体进行认证.访问实体只需记住一个口令字就能得到远程资源的服务，减轻了系统管理员维护实体账号的工作量，而且，访问实体可在其登录的机器上，按照用户口令字更改协议自由地更改应用边界安全设备上自己的口令字，方便了实体的使用.本文提出的认证方法基本上适合于所有的域间资源安全访问，而不论实体访问的何种服务，具有较广的适用范围.

参考文献：

[1] 沈昌祥.构造积极防御的安全保障框架[R]. 北京：第 2 届网络安全应用高峰论坛上的报告，2003.
SHEN Chang-xiang. Building Activity and Counterattack Framework of Security Assurance[R]. Beijing: Report on the Second Peak Forum of Network Security Application, September, 2003. (in Chinese)

[2] Internet RFC1928, SOCKS Protocol Version5[S]. 1996.

[3] Internet RFC1929, Username/Password Authentication for SOCKS V5[S]. 1996.

[4] Internet RFC 1510, The Kerberos Authentication Service (v5)[S]. 1993.

[5] Brainard J, Juels A, Kaliski B, et al. A New Two-Servers Approach for Authentication with Short Secrets[EB/OL]. http://www. usenix. org/events/sec03/tech/brainard. html August 2003.

[6] RFC 3174, US Secure Hash Algorithm 1 (SHA1)[S]. 2001.

[7] Lai X, Massey J. A Proposal for A New Block Encryption Standard[A]. Proceedings of Eurocrypt'90[C]. Berlin: Springer-Verlag, 1991.389－404.