基于不同口令认证的跨域组密钥协议

周福才 周恩光 闫 虹 苏晓曦
(东北大学信息学院 沈阳 110004)

摘 要 近年来关于基于口令认证的密钥交换协议(PAKE)进行了广泛的研究,基于口令认证的组密钥交换协议已成为安全协议研究的焦点问题。Byun等人也先后提出了基于不同口令认证的跨域环境下端到端的两个客户之间的PAKE(C2C-PAKE)密钥交换协议。然而在实际应用中,往往还需要在多个客户或客户组之间建立安全的通信信道。因此,提出了基于不同口令认证的跨域组间密钥交换协议,该协议将Zhiguo Wan等人所提出的nPAKE+协议扩展到了两个域,实现了两个域中的客户组在域服务器的协助下,建立域间共享的组会话密钥的过程,并给出了安全分析和执行效率的代价分析。

关键词 不同口令认证,跨域,组密钥交换
中图法分类号 TP301 **文献标识码** A

Cross-realm Group PAKE Protocol Using Different Passwords

ZHOU Fu-cai ZHOU En-guang YAN Hong SU Xiao-xi
(School of Information Science & Engineering, Northeastern University, Shenyang 110004, China)

Abstract Recently, several group password-authenticated key exchange (PAKE) protocols have been proposed. Byun and Zhigou Wan et al. also proposed different passwords group PAKE protocols successively, and the clients in their protocols were all in the same realm. However, in practice we also need to establish a secure communication channel between groups who are in different realms. So, we proposed a cross-realm group PAKE protocol using different passwords. The proposed protocol extends Zhigou Wan et al. 's nPAKE+ protocol from single realm to two realms. And it enables two groups in different realms to agree on a common group session key with the help of servers. We also gave the security analysis and computational costs for our protocol.

Keywords Different password authentication, Cross-realm, Group key exchange

1 引言

在移动网络、家庭网络等无处不在的环境中,端到端的安全越来越被人们所重视。近年来,基于口令认证的密钥交换协议(PAKE)被广泛研究。由于PAKE协议不需要成本高昂的公钥密码系统,在实际的应用中有很强的吸引力。一系列基于不同口令认证的密钥交换协议被提出[1,2,5,16,19],这些协议中的客户位于同一个域,每个客户拥有各自不同的口令,2个(或者n个)客户在域内服务器的协助下,可以生成一个共享的会话密钥。Bellovin和Merrit[1]首先提出了一个能够抵御字典攻击的2方PAKE(2PAKE)协议,客户与服务器之间利用共享的口令产生一个会话密钥并进行密钥确认。随后,Steiner等人[2]提出了一个3方PAKE协议,实现两个客户在服务器的协助下利用不同的口令生成会话密钥的过程。但该协议容易遭受不可检测在线字典攻击和离线字典攻击[3,4]。于是,Lin等人[5]又提出了一个改进的LSSH-3PEKE协议,它能够抵御不可检测在线字典攻击和离线的字典攻击。2002年,Byun等学者[6]首次提出了一种跨域环境下基于口令认证的密钥交换协议,简称为C2C-PAKE(Client to Client Password-authenticated Key Exchange)协议,该协议的目的是使得位于不同域中的两个拥有不同口令的客户,在没有共享密钥的情况下,在各自域服务器的协助下建立起一个安全的通信信道,该安全通信信道只能用于这两个客户间的通讯。随后Chen等学者[7]便指出该C2C-PAKE协议不能抵御另一个域中的服务器所发动的字典攻击。Kim等学者[8]也指出该协议不能抵御Denning-Sacco攻击[9],并提出了改进协议。不过Kim等人的改进协议不能抵御口令泄露伪造攻击和单向中间人攻击[10],于是Yoon等学者[10]又提出了改进协议。2007年,Byun等学者[11]重新提出了一个高效且比较完善的EC2C-PAKE协议,并对跨域口令密钥交换协议进行了形式化的安全证明。2007年Gang等人[12]也提出了一种能抵御多种已知攻击的协议[8]。然而Phan等人[13]和Yoneyama等人[14]指出[6,8,10-12]容易遭受不可检测的在线字典攻击,Yoneyama等人[14]还提出了能够抵御不可检测在线字典攻击的改进协议,该协议引入了IBE(ID-based encryption)系统。

以上所描述的协议都只能应用于单域或跨域环境中两个

到稿日期:2008-04-16 本文受国家自然科学基金资助项目(69874038),国家高技术研究发展计划(863计划:2001AA115300),辽宁省自然科学基金(20062023)资助。
周福才(1964—),男,教授,博士,E-mail:fczhou@mail.neu.edu.cn;**周恩光**(1985—),男,硕士研究生。

客户之间的密钥协商。然而在实际应用中往往还需要在多个客户或客户组之间建立安全的通信信道，例如客户间的协同工作、视频会议和个人区域网络等，往往还需要在不同域中的多个客户组成一个组来共同完成任务。在这一方面，Bresson等人[15]首次将基于口令认证与Diffie-Hellman的组密钥交换协议结合在一起，提出了一个基于口令认证的Diffie-Hellman组密钥交换协议，用于实现多个客户之间共享会话密钥的建立。但在Bresson等人的协议中，所有的组成员共享一个相同的口令，这在实际应用中将导致两个问题：

(1)组成员的口令一旦被泄露，组成员之间共享的口令就必须更新，这会花费高昂的代价；

(2)无法区分组成员的身份，不能根据实际需要划分子组。

2005年，为了建立 n 个客户(组)之间的安全通信信道，Byun等人[16]提出了一种改进的基于口令认证的组密钥交换协议(nPAKE)。协议中每个组成员拥有不同的口令，在域服务器的协助下生成一个共享组密钥。2007年，Zhiguo Wan等学者[19]也提出了一个高效、安全的nPAKE+协议，该协议利用逻辑密钥树将客户端的幂计算代价降低到 $O(\log n)$，其中 n 为客户的个数。

目前基于口令认证的组密钥交换协议[15,16,18,19]都局限于单域环境中进行研究。本文提出了基于不同口令认证的跨域的组密钥交换协议。该协议用于实现位于两个不同域中，各自拥有不同的口令的多个客户——客户组，在两个域服务器的协助下建立域间共享的组会话密钥的过程。

2 回顾 nPAKE+协议

nPAKE+协议使用了Diffie-Hellman密钥树。协议中的Diffie-Hellman密钥树是一棵二叉树，树的叶子节点用来表示域内的客户。每一个节点被定义为 $\langle l,v\rangle$，其中 l 为节点所处层次，v 表示 l 层的第几个节点，$1\leqslant v\leqslant 2^l-1$，根节点位于第0层。一个密钥为 $K_{\langle l,v\rangle}$ 的节点，它的盲密钥 $BK_{\langle l,v\rangle}=g^{K_{\langle l,v\rangle}}$。叶子节点的密钥 $K_{\langle l,v\rangle}$(或 K_i)是客户 A_i 和服务器 KDC_A 相互协作生成的。叶子节点 A_i 需要一组称为辅助路径 CP_i 的盲密钥，计算出一组称为密钥路径 KP_i 的密钥，从而最终求得组密钥 $K_{\langle 0,0\rangle}$。辅助路径 CP_i 根据密钥路径 KP_i 可分为两个子集，L_i 和 R_i，L_i 为左盲密钥集合，R_i 为右盲密钥集合。

一棵有 n 个叶子节点的Diffie-Hellman二叉密钥树中，叶子节点 A_i 使用 L_i，K_i 和 $\{BK_j:1\leqslant j\leqslant n\}$ 能够计算出 L_{i+1}。类似 A_i 使用 R_i，K_i 和 $\{BK_j:1\leqslant j\leqslant n\}$ 能够计算出 R_{i-1}。另外，对每一个叶子节点 A_i 使用 L_i，R_i 和 K_i 都能计算出组密钥 $K_{\langle 0,0\rangle}$。

nPAKE+协议包含3个基本流程，其描述如下：

(1) A_1 选择一个随机数 r_1，计算出 $X_1=\varepsilon_{PA_1}(g^{r_1})$。然后将信息 $\{A_i\}_{i=1}^n|X_1$ 发送给下一个客户。该信息将传送给从 A_1 至 A_n 的每一个客户，直到到达服务器 KDC_A。而信息每经过一个客户 A_i 都会选择一个随机数 r_i，计算出 $X_i=\varepsilon_{PA_i}(g^{r_i})$，并将其加入到信息中，最后一个客户 A_n 将信息 $\{A_i\}_{i=1}^n|\{X_i\}_{i=1}^n$ 发送给服务器 KDC_A。

(2) 服务器 KDC_A 用相应客户的密码 PA_i 将收到 $\{X_i\}|_{i=1}^n$ 解密得到 $\{g^{r_i}\}|_{i=1}^n$，然后 KDC_A 对每一个客户 A_i 都会选择一个随机数 s_i 计算出 KDC_A 和 A_i 之间的密钥 $K_i=(g^{r_i})^{s_i}$。服务器还将计算出 $Y_i=\varepsilon_{PA_i}(g^{s_i})$，$\pi=BK_1|A_1|\cdots|BK_n|A_n$ 和 $\tau_i=H(\pi|g^{r_i}|g^{s_i}|K_i)$，最后将 $\pi|\{Y_i|\tau_i\}_{i=1}^n$ 发送给 A_n。第2个流程中信息的传送顺序和第1个流程相反，服务器首先将 $\pi|\{Y_i|\tau_i\}_{j=1}^n$ 发送给 A_n，然后 A_n 将信息传送给他的前一个客户，直到到达 A_1。当客户 $A_i(i=n,n-1\cdots,1)$ 接收到传来的信息后，首先用他的口令解密 Y_i 得到 g^{s_i}，计算出 A_i 和 KDC_A 之间的密钥 $K_i=(g^{s_i})^{r_i}=g^{r_is_i}$ 和盲密钥 $BK_i=g^{K_i}$。然后验证计算得到的 BK_i 是否和 π 中的 BK_i 相等。接着 A_i 还需要对 π 进行验证，计算 $H(\pi|g^{r_i}|g^{s_i}|K_i)$是否等于 τ_i。当 $i\neq n$ 时，A_i 要计算出 $SK_i=(BK_{i+1})^{K_i}$，并判断 $H(R_i|SK_i)$是否等于 ξ_i 来验证 R_i。如果所有的验证都通过，则 A_i 使用 K_i，R_i 和 π 计算出 R_{i-1}，$SK_{i-1}=(BK_{i-1})^{K_i}$ 和 $\xi_{i-1}=H(R_{i-1}|SK_{i-1})$，最后将 $\pi|\{Y_j|\tau_j\}_{j=1}^{i-1}|R_{i-1}|\xi_{i-1}$ 发送给 A_{i-1}。

(3) A_1 接收到信息后，进行如上所述的所有验证。如果验证都通过，则 A_1 用 R_1，K_1 和 π 计算出组密钥 GKA_1。然后 A_1 计算出 L_2，$\sigma_1=H(L_2|SK_1)$ 和 $\eta_1=H(A_1|A_2|\cdots|A_n|K_1)$，再将 $L_2|\sigma_1|\eta_1$ 发送给 A_2。接着每个客户 $A_i(i=2,3,\cdots,n)$ 都将依次收到 $L_i|\sigma_{i-1}|\{\eta_j\}_{j=1}^{i-1}$，并通过计算 σ_{i-1} 是否等于 $H(L_i|SK_{i-1})$来验证 L_i。如果验证通过，A_i 再使用 K_i，L_i 和 R_i 计算出组密钥 GKA_i。如果 $i\neq n$，A_i 将使用 K_i，L_i 和 π 计算出 L_{i+1} 和 $\sigma_i=H(L_{i+1}|SK_i)$。然后将 $L_{i+1}|\sigma_i|\{\eta_j\}_{j=1}^i$ 发送给 A_{i+1}。如果 $i=n$，A_n 将计算出 η_n 后，将 $\{\eta_j\}_{j=1}^n$ 发送给服务器 KDC_A。

服务器 KDC_A 收到 $\{\eta_j\}_{j=1}^n$ 后，对每一个 η_j 进行验证。当发现某个 η_j 出现错误时，KDC_A 即可发现该未授权用户，这样可有效防御不可检测在线字典攻击。

3 基于不同口令认证的跨域组密钥交换协议

基于上述单域环境中的nPAKE+协议，文中提出了基于不同口令认证的跨域组密钥交换协议，用于实现位于两个不同域中的客户组在各自域服务器的协助下建立起一个域间共享的组会话密钥。如图1所示，假设域 A 中的服务器是 KDC_A，保存了域中客户 A_i 的不同口令 $PA_i(1\leqslant i\leqslant n)$；域 B 中的服务器是 KDC_B，保存域中客户 B_j 的不同口令 $PB_j(1\leqslant j\leqslant m)$。服务器 KDC_A 和 KDC_B 之间共享一个对称密钥 K。

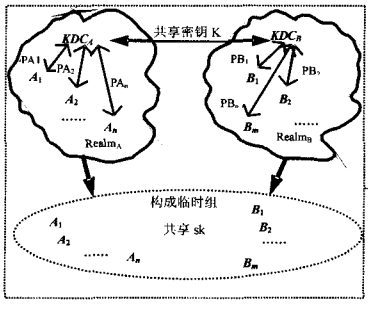

图1 跨域的组密钥交换

基于不同口令认证的跨域组密钥交换协议分为组密钥生成和验证两个阶段。

3.1 组密钥生成阶段

组密钥生成阶段包含9个基本流程，对协议的详细描述如下：

(1) A_1 选择一个随机数 r_1，并计算 $X_1=\varepsilon_{PA_1}(g^{r_1})$，然后将请求$\{A_i\}_{i=1}^n|\{B_j\}_{j=1}^m|X_1$ 发给域 A 中的下一个客户，该信息将传送给从 A_2 至 A_n 的每一个客户，直到到达服务器 KDC_A。而信息每经过一个客户 A_i 都会选择一个随机数 r_i，计算出 $X_i=\varepsilon_{PA_i}(g^{r_i})$，并将其加入到信息中，最后一个客户 A_n 将请求$\{A_i\}_{i=1}^n|\{B_j\}_{j=1}^m|\{X_l\}_{l=1}^n$发送给服务器 KDC_A。

(2) 服务器 KDC_A 依次解密信息 $X_i=\varepsilon_{PA_i}(g^{r_i})$，$1\leqslant i\leqslant n$，并为域 A 中的每一个客户 A_i 选择一个随机数 s_i 计算出 KDC_A 和 A_i 之间的密钥 $K_i=(g^{r_i})^{s_i}$。另外，服务器 KDC_A 还将计算出 $Y_i=\varepsilon_{PA_i}(g^{s_i})$，$\pi=BK_1|A_1|\cdots|BK_n|A_n|\{B_i\}_{i=1}^m$和 $\tau_i=H(\pi|g^{r_i}|g^{s_i}|K_i)$，然后将 $\pi|\{Y_j|\tau_j\}_{j=1}^n$发送给 A_n。A_n 收到来自于 KDC_A 发送来的信息后，以反方向将信息发送给下一个客户 A_{n-1}，直到到达 A_1。当每个客户 $A_i(i=n,n-1\cdots,1)$接收到传来的信息后，首先解密 Y_j 得到 g^{s_i}，计算出 A_i 和 KDC_A 之间的密钥 $K_i=(g^{s_i})^{r_i}=g^{r_is_i}$ 和盲密钥 $BK_i=g^{K_i}$。然后对 BK_i,π,τ_i 进行验证。当 $i\neq n$ 时，A_i 还要计算出 $SK_i=(BK_{i+1})^{K_i}$，并验证 R_i。如果所有的验证都通过，则 A_i 将计算出 R_{i-1}，$SK_{i-1}=(BK_{i-1})^{K_i}$ 和 $\xi_{i-1}=H(R_{i-1}|SK_{i-1})$，最后将 $\pi|\{Y_j|\tau_j\}_{j=1}^{i-1}|R_{i-1}|\xi_{i-1}$发送给 A_{i-1}。

(3)接收到信息后，再依次从 A_1 至 A_n 用 K_i,L_i,R_i 和 π 计算出组密钥 GKA，如果 $i\neq n$，A_i 将计算出 $L_{i+1},\sigma_i=H(L_{i+1}|SK_i)$ 和 $\eta_i=H(A_1|A_2|\cdots|A_n|K_i)$，并将 $L_{i+1}|\sigma_i|\{\eta_j\}_{j=1}^i$发送给 A_{i+1}；如果 $i=n$，A_i 将计算出 η_n 后，将$\{\eta_j\}_{j=1}^n$发送给服务器 KDC_A。

(4)服务器 KDC_A 收到$\{\eta_j\}_{j=1}^n$后，对每一个 η_j 进行验证。当所有 η_j 都验证通过后，KDC_A 则利用每一对密钥 K_i 和盲密钥 $BK_i(1\leqslant i\leqslant n)$计算出域 A 中 n 个客户的组密钥 GKA。接着，KDC_A 生成 $Ticket_B=\varepsilon_K(\{A_i\}_{i=1}^n|\{B_j\}_{j=1}^m|g^{GKA}|L)$，其中 L 为时间戳。然后将$\{A_i\}_{i=1}^n|\{B_j\}_{j=1}^m$和 $Ticket_B$ 发送给域 B 中的客户 B_1。

(5)B_1 接收到来自另一个域的信息后，从 B_1 至 B_m 将依次选择出各自的随机数 r'_j，计算出 $X'_j=\varepsilon_{PB_j}(g^{r'_j})$，$1\leqslant j\leqslant m$，并依次补充及传送$\{A_i\}_{i=1}^n|\{B_j\}_{j=1}^m|\{X'_j\}_{j=1}^m|Ticket_B$ 给下一个客户。并最终由 B_m 将信息$\{A_i\}_{i=1}^n|\{B_j\}_{j=1}^m|\{X'_j\}_{j=1}^m|Ticket_B$ 发送给服务器 KDC_B。

(6)服务器 KDC_B 首先使用与 KDC_A 的共享密钥 K 解密 $Ticket_B$，获得$\{A_i\}_{i=1}^n|\{B_j\}_{j=1}^m$和时间戳 L，并验证及确认信息的有效性。验证通过后，KDC_B 解密$\{X'_j\}|_{j=1}^m$得到$\{g^{r'_j}\}|_{j=1}^m$，然后 KDC_B 对每一个客户 B_j 选择一随机数 s'_j，计算出 KDC_B 和 B_j 之间的密钥 $K_j=g^{r'_js'_j}$ 及盲密钥 $BK'_j=g^{K'_j}$。服务器 KDC_B 还将计算出 $Y'_j=\varepsilon_{PB_j}(g^{s'_j})$，$\pi'=BK'_1|B_1|\cdots|BK'_m|B_m|\{A_i\}_{i=1}^n$，$\tau'_j=H(\pi'|g^{r'_j}|g^{s'_j}|K_j|g^{GKA})$，然后将信息 $\pi'|g^{GKA}|\{Y'_j|\tau'_j\}_{j=1}^m$以相反的顺序依次传送给 $B_m,B_{m-1},\cdots$，直到 B_1。当客户 $B_j(j=m,m-1\cdots,1)$接收到传来的信息后，首先解密 Y'_j 得到 $g^{s'_j}$，然后计算 B_j 和 KDC_B 之间的密钥 $K'_j=(g^{s'_j})^{r'_j}=g^{r'_js'_j}$ 和盲密钥 $BK'_j=g^{K'_j}$，并验证 BK'_j,π' 及 τ'_j。当 $j\neq m$ 时，B_j 还需要计算 $SK'_j=(BK'_{j+1})^{K'_j}$，并验证 R'_j。如果所有的验证都通过，则 B_j 使用 K'_j,R'_j 及 π' 计算出 R'_{j-1}，$SK'_{j-1}=(BK'_{j-1})^{K'_j}$ 以及 $\xi'_{j-1}=H(R'_{j-1}|SK'_{j-1})$，最后将 $\pi'|g^{GKA}|\{Y'_i|\tau'_i|\}_{i=1}^{j-1}|R'_{j-1}|\xi'_{j-1}$发送给 B_{j-1}。

(7)接收到信息后，再依次从 B_1 至 B_m 用 K'_j,L'_j,R'_j 和 π'计算出域 B 中的组密钥 GKB。如果 $j\neq m$，B_j 还将计算出 $L'_{j+1},\sigma'_j=H(L'_{j+1}|SK'_j)$和 $\eta'_j=H(B_1|B_2|\cdots|B_m|K'_j)$，并将 $L'_{j+1}|\sigma'_j|\{\eta'_i\}_{i=1}^j$发送给 B_{j+1}。如果 $j=m$，B_m 将计算出 η'_m 后，将$\{\eta'_j\}_{j=1}^m$发送给服务器 KDC_B。服务器 KDC_B 收到$\{\eta'_j\}_{j=1}^m$后，对每一个 η'_j 进行验证。验证通过后，服务器 KDC_B 首先计算出域 B 的组密钥 GKB，然后生成一个 $Ticket_A=\varepsilon_K(\{A_i\}_{i=1}^n|\{B_j\}_{j=1}^m|g^{GKB}|L')$，发送给 KDC_A。

(8)KDC_A 接收到来自 KDC_B 传送的信息后，解密并验证 $Ticket_A$ 的有效性。验证通过后，KDC_A 将 $g^{GKB}|\{MAC_{K_i}(g^{GKB})\}_{i=1}^n$依次发送给 $A_n,A_{n-1},\cdots,A_1$。$A_i(1\leqslant i\leqslant n)$

(9) A_i 在收到信息后，使用与服务器 KDC_A 之间的密钥 K_i 解密消息验证码 $MAC_{K_i}(g^{GKB})$，以此对 g^{GKB} 的完整性进行验证。验证通过后，A_i 即可计算出最终的跨域组密钥 $GK=(g^{GKB})^{GKA}$，并继续将信息 $g^{GKB}|\{MAC_{K_j}(g^{GKB})\}_{j=1}^{i-1}$传送给下一个客户 A_{i-1}，直到 A_1。至此，即完成了两个域中的客户生成跨域组密钥的过程。

3.2 组密钥验证阶段

如果每一个客户都需要确认是否与其他的客户生成了相同的组密钥，则需要进行验证。在验证阶段，每一个客户都将 $Auth_{identity}=(identity|GK)$发送给其他所有客户(例如用户 A_1 发送 $Auth_{A_1}=(A_1|GK)$)，这样每个客户需要验证 $m+n-1$ 个 $Auth$。

4 安全分析和执行效率

4.1 安全分析

重放攻击：该协议中所使用的 $r_i,s_i(1\leqslant i\leqslant n),r'_j,s'_j(1\leqslant j\leqslant m)$都是临时变量，时间戳 L 和 L' 又分别表示了 $Ticket_B$ 和 $Ticket_A$ 的生存周期。所以敌手进行重放攻击的概率可以忽略。

不可检测在线字典攻击：该协议中的每一个客户 A_i(或者 B_j)都要将 η_i(或者 η'_j)发送给服务器 KDC_A(或者 KDC_B)，服务器可以通过验证 η_i(或者 η'_j)来防止不可检测在线字典攻击。

离线字典攻击：现将攻击者分为两类，一类是外部攻击者，攻击者不知道组内客户的口令，要对组内成员的口令进行攻击；一类是内部攻击者，攻击者已知组内一个或者多个客户的口令，要对另一个客户的口令进行攻击。假设攻击者能够截获协议中的 $X_i,Y_i(1\leqslant i\leqslant n),X'_j,Y'_j(1\leqslant j\leqslant m)$，但是攻击者无法获取使用客户口令所加密的信息内容，所以无论是内部攻击者还是外部攻击者都无法对客户的口令发动离线字典攻击。

前向安全性：协议中如果客户 A_i(或 B_j)的口令丢失，则攻击者能够获取之前的 g^{GKA} 和 g^{GKB}，但基于 CDH 难题，攻击者无法计算出之前的组密钥 $GK=g^{GKA\cdot GKB}$。

密钥生成控制攻击：协议所生成的组密钥 $GK=g^{GKA\cdot GKB}$ 是所有组成员共同参与决定的。没有任何一个成员能够独立决定最后所生成的组密钥，或者使组密钥落入到一个预先决定的范围。

4.2 执行效率

基于不同口令认证的跨域组密钥交换协议，由于使用了 Diffie-Hellman 的二叉密钥树，因此它的计算成本和执行效率

与树的结构密切相关。如同 Zhiguo Wan 等人对 nPAKE+协议的执行效率分析，该协议的幂计算代价也在 O(log$n$)。协议的计算成本如表 1 所示。

表 1　计算成本

	域 A 中客户	KDC_A	KDC_B	域 B 中客户
指数运算	$n(7+\log n)$	$4n+2$	$3m+2$	$m(5+\log m)$

结束语　本文所提出的协议将 Zhiguo Wan 等学者提出的适用单域中的 nPAKE$^+$ 协议扩展到两个域中，该协议是一个基于不同口令认证的跨域间的组密钥交换协议，实现了两个域中的客户组在域服务器的协助下建立域间共享的组会话密钥的过程，其中每个用户都与一个可信任的服务器共享一个独立的密码。该协议在组密钥生成和协商时使用了 Diffie-Hellman 密钥树，因此在计算和通信代价方面具有更高的效率。通过对该协议的安全分析和执行效率的分析，表明该协议是安全高效的。

参 考 文 献

[1] Bellovin S, Merrit M. Encrypted key exchange: password based protocols secure against dictionary attacks // Proceedings of the Symposium on Security and Privacy. IEEE, 1992: 72-84

[2] Steiner M, Tsudik G, Waider M. Refinement and extension of encrypted key exchange // ACM Operation Sys. Review, 1995, 29 (3): 22-30

[3] Ding Y, Horster P. Undetectable on-line password guessing attacks. ACM Operating Systems Review, 1995, 29(4): 77-86

[4] Lin C, Sun H, Hwang T. Three-party encrypted key exchange: attacks and a solution. ACM Operating Systems Review, 2000, 34(4): 12-20

[5] Lin C, Sun H, Steiner M, et al. Three-party Encrypted Key Exchange Without Server Public-Keys. IEEE Communications Letters, IEEE Press, 2001, 5(12): 497-499

[6] Byun J W, Jeong I R, Lee D H, et al. Password-authenticated key exchange between clients with different passwords // Proceedings of ICICS'02. LNCS Vol. 2513. Springer-Verlag, 2002: 134-146

[7] Chen L. A Weakness of the Password - Authenticated Key Agreement between Clients with Different Passwords Scheme // The document was being circulated for consideration at the 27th the SC27/WG2 meeting. Paris, France, 2003-10-20/24

[8] Kim J Y, Kim S J, Kwak J, et al. Cryptanalysis and Improvement of Password Authenticated Key Exchange between Clients with Different Passwords // ICCSA 2004. LNCS 3043. 2004: 895-902

[9] Denning D, Sacco G. Timestamps in key distribution protocols. Communications of the ACM, 1981, 24 (8): 533-536

[10] YoonEn-J, Yoo K-Y. A Secure Password - Authenticated Key Exchange Between Clients with Different Passwords // APWeb Workshops 2006. LNCS 3842. Berlin Heidelberg: Springer-Verlag, 2006: 659-663

[11] Byun J W, Lee D H, Lim J. EC2C-PAKA: An efficient client-to-client password-authenticated key agreement

[12] Gang Y, Dengguo F, Xiaoxi H. Improved Client-to-Client Password-Authenticated Key Exchange Protocol // IEEE ARES 2007. 2007: 564-574

[13] Phan R C W, Goi B M. Cryptanalysis of an Improved Client-to-Client Password-Authenticated Key Exchange (C2C-PAKE) Scheme // Ioannidis J, Keromytis A D, Yung M, eds. ACNS 2005. LNCS, vol. 3531. Heidelberg: Springer, 2005: 33-39

[14] Yoneyama K, Ota H, Ohta K. Secure Cross-Realm Client-to-Client Password-Based Authenticated Key Exchange Against Undetectable On-Line Dictionary Attacks // AAECC 2007. LNCS 4851. Berlin Heidelberg: Springer-Verlag, 2007: 257-266

[15] Bresson E, Chevassut O, Pointcheval D. Group Diffie - Hellman Key Exchange Secure against Dictionary Attacks // Zheng Y, ed. ASIACRYPT 2002. LNCS, vol. 2501. Heidelberg: Springer, 2002

[16] Byun J W, Lee D H. N-Party Encrypted Diffie-Hellman Key Exchange Using Different Passwords // Ioannidis J, Keromytis A D, Yung M, eds. ACNS 2005. LNCS, vol. 3531. Heidelberg: Springer, 2005: 75-90

[17] Tang Q, Chen L. Weaknesses in two group Diffie-Hellman Key Exchange Protocols. Cryptology ePrint Archive, 2005, 197

[18] Byun J W, Lee D H, Lim J. Password - based Group Key Exchange Secure Against Insider Guessing Attacks // Proceedings of CIS'05. LNAI Vol. 3802. Springer-Verlag, 2005: 143-148

[19] Wan Zhiguo, Deng R H, Bao Feng, et al. nPAKE$^+$: A Hierarchical Group Password-Authenticated Key Exchange Protocol Using Different Passwords // ICICS 2007. LNCS 4861. Berlin Heidelberg: Springer-Verlag, 2007: 31-43

[20] Bresson E, Chevassut O, Pointcheval D, et al. Provably authenticated group diffie-hellman key exchange // Proceedings of 8th ACM Conference on Computer and Communications Security. 2001: 255-264

(上接第 57 页)

[15] Holloway G, Smith M D. The Machine - SUIF Control Flow Graph Library. http://www.eecs.harvard.edu/hube/software/nci/cfg.pdf, 2002

[16] Paul J, Purdom W, Moore E F. Immediate predominators in a directed graph. Communications of the ACM, 1972, 15(8): 777-778

[17] Allen F E, Cocke J. Graph-theoretic constructs for program flow analysis. Technical Report RC 3923 (17789). IBM Thomas J. Watson Research Center, July 1972

[18] Buchsbaum A L, Kaplan H, et al. Linear - time pointer - machine algorichms for least common ancestors, mst Verification, and dominztors // Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing. 1998: 279-288

[19] Zhao Kejia. GCC 基本块与控制流程图的数据结构分析. Technical Report. Creative Compiler Research Group, 2002