Computer Science and Application 计算机科学与应用, 2018, 8(5), 591-600
Published Online May 2018 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2018.85067

Hans汉斯

Cross-Domain Authentication Key Exchange Protocol Based on Certificateless in Cloud Computing Environment

Xinglan Zhang, Dingling Li*

Department of Information, Beijing University of Technology, Beijing
Email: *769583197@qq.com

Received: Apr. 12th, 2018; accepted: Apr. 26th, 2018; published: May 3rd, 2018

Abstract

Based on the discrete logarithm problem and bilinear pairings on the elliptic curve, this paper proposed a certificateless cross-domain authentication key exchange protocol in cloud computing environment and was proved secure in extended eCK model. The protocol satisfied the authentication between the different clouds and the user's private key in a cloud consisted of a secret value selected by the user and a partial private key generated by the authentication server in the cloud. It's fully guaranteed the security of the user's private key. The calculation in user identity authentication finished by the cloud authentication server and user authentication and key agreement were completed in an interaction. It has improved the execution efficiency of the agreement.

Keywords

Cloud Computing, Certificateless, Cross-Domain Authentication, Key Exchange, ECK Model

云计算下基于无证书的跨域认证密钥交换协议

张兴兰，黎丁玲*

北京工业大学信息学部，北京
Email: *769583197@qq.com

收稿日期：2018年4月12日；录用日期：2018年4月26日；发布日期：2018年5月3日

摘　要

基于椭圆曲线上的离散对数难题和双线性对运算，提出了一个云计算中基于无证书的跨域认证密钥交换

*通讯作者。

文章引用: 张兴兰, 黎丁玲. 云计算下基于无证书的跨域认证密钥交换协议[J]. 计算机科学与应用, 2018, 8(5): 591-600. DOI: 10.12677/csa.2018.85067

协议，并且在eCK (extended Canetti-Krawczyk)模型下证明所提出协议的安全性。该协议满足了不同云之间的认证，一个云中用户的私钥由用户自己生成的秘密值和该云中的认证服务器生成的部分私钥两部分组成，充分保证了用户私钥的安全。用户身份认证的计算交由云认证服务器完成，用户认证和密钥协商经由一次交互完成，提高了协议执行效率。

关键词

云计算，无证书，跨域认证，密钥交换，eCK模型

1. 引言

云计算作为一种新兴的资源共享模式，充分利用了分布式计算[1]和虚拟资源管理等一些技术，并且整合了网络中大量的资源，为用户提供了存储、计算、软件以及平台等服务[2]也给用户带来极大的便利条件，但云计算的不断发展和普及，随之而来的则是安全[3]方面的问题。

云计算依照部署模型可分为：公有云、私有云、社区云和混合云四种类型[4]。混合云一般由多个云组成，每个云都有自己的用户身份管理系统。每个云中，有云内用户信任的认证服务器，提供用户注册等请求。而云间的资源共享要求，用户在访问自己所属云之外的其它云时，要考虑与不同云用户的身份认证[5]和密钥协商[6]。

针对云计算中存在的跨域认证[7]问题，Xu C 等[8]基于零知识证明和密钥托管的思想提出了一个认证方案，该方法实现了匿名性和用户真实信息的追踪。Castiglione A 等[9]基于盲签名实现了一个跨域认证方案来支持用户身份验证，保护用户身份信息。这个方案满足了匿名性，可以避免中间人攻击，服务器的欺骗攻击等。解福[10]实现了基于验证元客户端到客户端的跨域认证协议，能够抵御主动和被动攻击，但该协议的交互复杂。现有的方案需要在认证服务器上保存用户的身份信息或者用来验证身份的信息，因此存在证书管理问题。2003 年，Al-Riyami 和 Paterson [11]提出无证书的公钥密码体制，这一体制解决了密钥托管问题。

本文提出了一个基于无证书的跨域认证方案，并且在 eCK 模型[12]下证明了所提出协议的安全性。在本文的协议中，私钥由两部分组成，不存在密钥托管问题。经过一次交互就完成了用户认证和密钥协商，提高了协议的效率，用户认证部分也充分利用了云服务器的计算能力，适合于云计算环境。

2. 预备知识

2.1. 椭圆曲线上的双线性映射

双线性对[13]是两个循环群间的线性映射关系。设 G_1 和 G_2 分别是阶为 q 的加法循环群和乘法循环群，q 是大素数，且在 G_1 和 G_2 中离散对数问题都是难解的。e 是 $G_1\times G_1\to G_2$ 的双线性映射，要满足以下性质：

1) 双线性：对于任意 $P,Q\in G_1,a,b\in Z_q^*$，满足 $e(aP,bQ)=e(P,Q)^{ab}$ ；对于任意 $P_1,P_2,Q\in G_1$，有 $e(P_1+P_2,Q)=e(P_1,Q)\times e(P_2,Q)$。

2) 非退化性：若 P 是 G_1 的生成元，则 $e(P,P)\in G_2$ 是 G_2 的生成元，即 $e(P,P)\neq 1$ 。

3) 可计算性：有多项式时间算法可计算映射 e。

2.2. 相关困难问题及假设

CDH(Computational Diffie-Hellman)问题：G 为 q 阶的循环加法群，P 为 G 的生成元，给定 $P, aP, bP \in G$ (其中 $a, b \in Z_q^*$)，计算 $abP \in G$ 。

定义 1：在安全参数 λ 下多项式时间算法 A 解决 CDH 问题的优势为 $Adv_A^{CDH}(\lambda) = \Pr\left[A(P, aP, bP) = abP \mid P \in G, a, b \in Z_q^*\right]$

定义 2：CDH 假设：对于任意多项式时间算法 A，$Adv_A^{CDH}(\lambda)$ 是可忽略的。

2.3. 安全模型

针对无证书认证密钥交换协议，有两种类型的敌手[14]。敌手 A_1 不知道系统主密钥，但是 A_1 可以把任意协议参与者的公钥替换成自己所选的值。敌手 A_2 知道系统所拥有的主密钥，但是不可以替换协议参与者的公钥。

$\prod_{i,j}^{s}$ 表示参与者 i 和 j 的第 s 个会话。

Lippold [15]等将传统的 eCK 模型扩展成为无证书下的 eCK 模型，该模型是通过挑战者 C 和敌手 $A \in \{A_1, A_2\}$ 之间进行的游戏来定义的。该游戏分成两个阶段：

阶段 1：敌手可以以任何顺序进行以下查询。

Create(i)：C 为身份是 ID_i 的协议参与者 i 生成公钥和私钥对。

RevealMasterKey：C 把系统主密钥返回给敌手 A。

RevealSessionKey($\prod_{i,j}^{s}$)：如果该会话已计算出会话密钥，则把该会话密钥返回，否则返回 $\perp$ 。

RevealPartialPrivateKey(i)：C 把参与者 i 的部分私钥返回给敌手 A。

RevealSecretValue(i)：C 把参与者 i 的秘密值返回给敌手 A。

ReplacePublicKey(i,pk)：C 把参与者 i 的公钥更换为 A 所选的值 pk。

RevealEphemeralKey($\prod_{i,j}^{s}$)：C 把参与者 i 的临时密钥返回给 A。

Send($\prod_{i,j}^{s}$,m)：A 向会话发送消息 m，然后依据协议的执行得到相应的响应信息。

如果 A 认为第一个阶段的查询结束了，则 A 选一个新鲜会话，然后进行游戏的第二阶段，执行 Test() 查询。

阶段 2：Test()：第一阶段询问结束，A 选一个新鲜会话 $\prod_{i,j}^{s}$ 执行 Test 查询。随机选 $b \in \{0,1\}$，如果 $b=0$，则返给 A 会话密钥，如果 $b=1$，则在会话密钥空间里选一个随机的值返回给 A。

在游戏的最后，A 输出对 b 的猜测 b'，如果 $b'=b$，则 A 赢得游戏。A 赢得该游戏的优势为 $Adv_A(k) = \left|\Pr[b'=b] - \frac{1}{2}\right|$，其中 k 为安全参数。

定义 3：新鲜会话。为参与者 i 和 j 已经结束的会话。如果下面的条件都不成立，则称是新鲜的。

1) A 查询了的会话密钥。

2) A 查询了 i 拥有的部分私钥或秘密值和会话的临时密钥。

定义 4：安全性。当认证密钥交换协议达到如下条件[16]时，则称该协议是安全的。

1) 若攻击者诚实的转发消息，且参与者接受该会话，则参与者能获得在会话密钥空间内均匀分布的密钥。

2) 对于任意的多项式时间敌手 A，能够赢得游戏的概率是可忽略的。

3. 基于无证书的跨域认证方案

基于苏航等[17]的无证书方案设计和 CDH 假设，提出了本文的协议。

3.1. 方案设计

1) 系统初始化算法

选取满足安全常数 λ 的阶为 q 的椭圆曲线循环群 G_1 和 G_2，即 $|q|=\lambda$，G_1 的生成元为 P。G_1 和 G_2 分别是加法循环群和乘法循环群，e 是 $G_1\times G_1\to G_2$ 的双线性映射。选取安全散列函数 $H_1:\{0,1\}^*\times G_1\to Z_q^*$，$H_2:G_1\to K$，其中 K 是会话密钥空间。

域 A 和域 B 分别选取 $s_A\in Z_q^*$，$s_B\in Z_q^*$ 作为系统主私钥即 $msk_A=s_A$，$msk_B=s_B$，计算公钥 $P_{pub_A}=s_AP$，$P_{pub_B}=s_BP$。域 A 和域 B 在各自域内公开系统参数 $params_A=\{G_1,G_2,e,q,P,H_1,H_2,P_{pub_A}\}$ 和 $params_B=\{G_1,G_2,e,q,P,H_1,H_2,P_{pub_B}\}$。

2) 用户部分私钥生成算法

域 A 内用户 user 向域 A 发出注册请求并发送自己的身份信息 ID_U，域 A 随机选 $g\in Z_q^*$，计算 $k=s_A+gH_1(ID_U\|gP)$ 并把它作为用户的部分私钥，并把 $\{gP,k\}$ 发送给 user。

user 验证 $kP=P_{pub_A}+gPH_1(ID_U\|gP)$，相等则接收 k。若等式不成立，则拒绝此部分私钥。

3) 用户公私钥生成算法

用户 user 随机选 $x\in Z_q^*$，计算私钥 $d_U=k+x$，输出公钥 $pk_U:\{gP,xP\}$。

4) 密钥协商算法

用户 user 选择临时密钥 $t_U\in Z_q^*$，计算临时信息 $T_U=t_Ud_UP$，$T_U'=t_UP$。选随机数 N_r，同时用域 B 的公钥加密 $H_1(ID_U\|gP)$ 和随机数 N_r，得到 $e_U=\{H_1(ID_U\|gP),N_r\}_{P_{pub_B}}$，把 T_U,T_U',e_U,pk_U 发送给域 B。

域 B 解密 e_U 得到 $H_1(ID_U\|gP)$ 和 N_r 验证等式 $e(T_U,P)=e(T_U',P_{pub_A})e(gPH_1(ID_U\|gP),T_U')e(T_U',xP)$，验证通过则选择临时密钥 $t_B\in Z_q^*$，计算临时信息 $T_B=t_Bs_BP$，计算出会话密钥 sk_B 后用该会话密钥加密 N_r 得到 $e_B=\{N_r\}_{sk_B}$，把 $\{T_B,e_B,P_{pub_B}\}$ 发送给用户 user。

用户 user 解密 e_B 得到 N_r'，确认 N_r 和 N_r' 是不是相等，若相等，则可确定域 B 的身份，进而双方可用计算得到的会话密钥进行安全的通信。

用户 user 计算：

$$k_{U_1}=d_UT_B+d_Ut_Us_BP$$

$$k_{U_2}=t_Ud_UT_B=t_Ut_Bd_Us_BP$$

会话密钥 $sk_U=H_2(T_U\|T_B\|k_{U_1}\|k_{U_2})$

域 B 计算：

$$k_{B_1}=s_BT_U+t_Bs_B(P_{pub_A}+gPH_1(ID_U\|gP)+xP)$$

$$k_{B_2}=t_Bs_BT_U=t_Bt_Us_Bd_UP$$

会话密钥 $sk_B=H_2(T_U\|T_B\|k_{B_1}\|k_{B_2})$

3.2. 正确性分析

1) 域 B 验证用户的正确性

$$
\begin{aligned}
e\left(T_U,P\right)&=e\left(t_U d_U P,P\right)=e\left(t_U\left(k+x\right)P,P\right)\\
&=e\left(t_U\left(s_A+gH_1\left(ID_U\|gP\right)+x\right)P,P\right)\\
&=e\left(\left(t_U s_A+t_U gH_1\left(ID_U\|gP\right)+t_U x\right)P,P\right)\\
&=e\left(t_U s_A P,P\right)e\left(t_U gH_1\left(ID_U\|gP\right)P,P\right)e\left(t_U xP,P\right)\\
&=e\left(P,P\right)^{t_U s_A}e\left(P,P\right)^{t_U gH_1\left(ID_U\|gP\right)}e\left(P,P\right)^{t_U x}
\end{aligned}
$$

$$
\begin{aligned}
&e\left(T'_U,P_{pub_A}\right)e\left(gPH_1\left(ID_U\|gP\right),T'_U\right)e\left(T'_U,xP\right)\\
&=e\left(t_U P,s_A P\right)e\left(gPH_1\left(ID_U\|gP\right),t_U P\right)e\left(t_U P,xP\right)\\
&=e\left(P,P\right)^{t_U s_A}e\left(P,P\right)^{t_U gH_1\left(ID_U\|gP\right)}e\left(P,P\right)^{t_U x}
\end{aligned}
$$

2) 协议的正确性

协议正确，则需证明用户 user 和域 B 计算得到了相同的会话密钥，即证明 $sk_U=sk_B$。

用户 user 的计算如下：

$$k_{U_1}=d_U T_B+d_U t_U s_B P=d_U t_B s_B P+d_U t_U s_B P=\left(t_B+t_U\right)d_U s_B P$$

域 B 的计算如下：

$$
\begin{aligned}
k_{B_1}&=s_B T_U+t_B s_B\left(P_{pub_A}+gPH_1\left(ID_U\|gP\right)+xP\right)=s_B T_U+t_B s_B\left(kP+xP\right)\\
&=s_B T_U+t_B s_B d_U P=s_B t_U d_U P+t_B s_B d_U P=\left(t_U+t_B\right)d_U s_B P
\end{aligned}
$$

由计算结果可知，用户 user 和域 B 能得到一样的会话密钥。

3.3. 安全性证明

下面给出本文提出的协议在 eCK 模型下的安全性的证明，H_1 和 H_2 是随机预言机。

引理 1：由于 CDH 问题是困难问题，则本文协议在 A_1 的攻击下是安全的。

证明：假设 A_1 能以不可忽略的优势 Adv_{A_1} 赢得 2.3 节中定义的游戏。那么挑战者 C 可以利用 A_1 的能力解决 CDH 问题。C 随机选 $P_0\in G_1$，设置 P_0 为协商用户所在域(域 A)的公钥，并令 $P_{pub_A}=P_0$。令域 A 内全局系统参数为 $\left\{G_1,G_2,e,q,P,H_1,H_2,P_0\right\}$，并把该参数发给 A_1。

令 n_0 为参与方可拥有的最多会话数，n_1 为最多激活的用户数，n_2 为最多进行的哈希询问次数。令 Adv_C^{CDH} 为 C 解决 CDH 问题的优势。为了解决 CDH 问题，给定 CDH 挑战 $U=uP$，$V=vP\left(u,v\in Z_q^*\right)$ 和一个预言机 DDH(*,*,*)，C 的任务是计算 $CDH\left(U,V\right)=uvP$。C 模拟 2.3 节中定义的游戏，在游戏中 C 要回答 A_1 的所有询问。

在游戏开始前，C 随机选 $u\in\left\{1\cdots n_1\right\}$，代表参与跨域认证的用户。随机选 $t\in\left\{1\cdots n_0\right\}$，选择 $\prod_{u,B}^{t}$ 作为 Test 会话。

将敌手攻击的情况[18]分为以下几种来讨论：

1) 不知道用户的部分私钥和域 B 的临时密钥。

Create(i)：C 维护一个初始为空的列表 L_C，其中存储的元组格式为 $\left(ID_i,k_i,G_i,x_i,P_i\right)$，如果 $i=u$，C 随机选 $x_i,h_i\in Z_q^*$，计算 $G_i=\left(U-P_0\right)h_i^{-1}$，$P_i=x_i P$，设置 $H_1\left(ID_i\|G_i\right)=h_i$，并在 L_C 中存储 $\left(ID_i,\perp,G_i,x_i,P_i\right)$，在 L_{H_1} 中存储 $\left(ID_i,G_i,h_i\right)$。如果 $i\neq u$，随机选 $k_i,x_i,h_i\in Z_q^*$，计算 $G_i=\left(k_i P-P_0\right)h_i^{-1}$，$P_i=x_i P$，设置 $H_1\left(ID_i\|G_i\right)=h_i$，并在 L_C 中存储 $\left(ID_i,k_i,G_i,x_i,P_i\right)$，在 L_{H_1} 中存储 $\left(ID_i,G_i,h_i\right)$。

$H_1(ID_i, G_i)$ 询问：C 维护一个初始为空的列表 L_{H_1}，其中存储的元组格式为 (ID_i, G_i, h_i)。如果 (ID_i, G_i) 在 L_{H_1} 中，则返回 h_i 给 A_1。否则随机选择 $h_i \in Z_q^*$ 返回给 A_1，并在 L_{H_1} 中存储 (ID_i, G_i, h_i)。

$H_2(ID_i, T_i, T_B, Z_1, Z_2)$ 询问：C 维护一个初始为空的列表 L_{H_2}，记录的元组格式为 $(ID_i, T_i, T_B, Z_1, Z_2, sk)$。如果查询的元组在 L_{H_2} 中，则返回 sk 给 A_1。否则按如下操作：

如果 $i = u$，C 在列表 L_S 中找形如 $(ID_i, T_i, T_B, pk_i, sk)$ 的元祖，若找到，则计算

$$\overline{Z_1} = Z_1 - x_i(T_B + t_i P_{pub_B})$$

$$\overline{Z_2} = Z_2 - x_i t_i T_B$$

C 检查分别输入 $(P_0 + G_i H_1(ID_i \| G_i), T_B + t_i P_{pub_B}, \overline{Z_1})$，$(P_0 + G_i H_1(ID_i \| G_i), t_i T_B, \overline{Z_2})$，DDH 预言机是否输出 1。如果 Z_1 和 Z_2 计算正确，在 L_{H_2} 中存储 $(ID_i, T_i, T_B, Z_1, Z_2, sk)$，$sk$ 为 L_S 中的值。如果计算错误，则随机选 $sk \in K$，在 L_{H_2} 中存储 $(ID_i, T_i, T_B, Z_1, Z_2, sk)$，并返回 sk 给 A_1。

如果 $i \neq u$，在列表 L_S 中找形如 $(ID_i, T_i, T_B, pk_i, sk)$ 的元祖，若找到，则在 L_{H_2} 中存储 $(ID_i, T_i, T_B, Z_1, Z_2, sk)$，$sk$ 为 L_S 中的值。否则随机选 $sk \in K$，在 L_{H_2} 中存储 $(ID_i, T_i, T_B, Z_1, Z_2, sk)$，并返回 sk 给 A_1。

RevealMasterKey：C 停止模拟。

RevealSessionKey($\prod_{i,j}^s$)：如果 $\prod_{i,j}^s = \prod_{u,B}^t$，则 C 停止模拟，否则把会话密钥 sk 返回给 A_1。

RevealPartialPrivateKey(i)：如果 $i = u$，则 C 停止模拟，否则把部分私钥 k_i 返回给 A_1。

RevealSecretValue(i)：C 查找 L_C 列表，若找到形如 $(ID_i, *, *, *, *)$ 的元组，则返回 x_i 给 A_1。否则执行 Create(i)，返回 x_i 给 A_1。

ReplacePublicKey(i,pk)：C 查找 L_C 列表，若找到形如 $(ID_i, *, *, *, *)$ 的元组，则替换 x_i 和 P_i 为 x_i' 和 P_i'，其中 $pk = P_i'$，$P_i' = x_i' P$。若没有找到，则执行 Create(i)，再替换 x_i 和 P_i 为 x_i' 和 P_i'。

RevealEphemeralKey(i)：如果 $\prod_{i,j}^s = \prod_{u,B}^t$，则 C 停止模拟，否则 C 发送临时密钥给 A_1。

Send($\prod_{i,j}^s$,m)：C 维护一个初始为空的列表 L_S，其中存储的元组格式为 $(ID_i, T_i, T_B, pk_i, sk)$。

如果 $\prod_{i,j}^s = \prod_{u,B}^t$，则返回 $T_B = V$ 给 A_1。

如果 $i = u$，C 随机选择 $t_i \in Z_q^*$，并计算

$$\overline{Z_1} = Z_1 - x_i(T_B + t_i P_{pub_B})$$

$$\overline{Z_2} = Z_2 - x_i t_i T_B$$

C 检查分别输入 $(P_0 + G_i H_1(ID_i \| G_i), T_B + t_i P_{pub_B}, \overline{Z_1})$，$(P_0 + G_i H_1(ID_i \| G_i), t_i T_B, \overline{Z_2})$，DDH 预言机是否输出 1，如果 Z_1 和 Z_2 计算正确，则在 L_S 中记录 $(ID_i, T_i, T_B, pk_i, sk)$，$sk$ 为 L_{H_2} 中的值。否则，随机选 $sk \in K$，在 L_S 中存储 $(ID_i, T_i, T_B, pk_i, sk)$。

如果 $i \neq u$，则按协议规则进行回答。

Test()：如果 $\prod_{i,j}^s = \prod_{u,B}^t$，则 C 停止模拟，否则，C 随机选 $sk \in K$，并把 sk 返回给 A_1。

假设 A_1 能够赢该游戏，那么 A_1 一定计算出了准确的 Z_1 和 Z_2。C 有 $\frac{1}{n_2}$ 的概率在 L_{H_2} 中找到正确的元组。C 计算：

$$\overline{Z_2} = Z_2 - x_i t_i T_B = t_i CDH(U, V)$$

$$CDH(U, V) = t_i^{-1}(Z_2 - x_i t_i V)$$

则 C 解决 CDH 问题的优势 $Adv_C^{CDH} \geq \frac{1}{n_0 n_1 n_2} Adv_{A_1}$，C 以不可忽略的优势解决了 CDH 问题，这与 CDH 假设冲突。

2) 不知道用户的临时密钥和域 B 的临时密钥。

Create(i)：C 维护一个初始为空的列表 L_C，其中存储的元组格式为 $(ID_i, k_i, G_i, x_i, P_i)$，C 随机选 $k_i, x_i, h_i \in Z_q^*$，计算 $G_i = (k_i P - P_0) h_i^{-1}$，$P_i = x_i P$，设置 $H_1(ID_i \| G_i) = h_i$，并在 L_C 中存储 $(ID_i, k_i, G_i, x_i, P_i)$，在 L_{H_1} 中存储 (ID_i, G_i, h_i)。

$H_2(ID_i, T_i, T_B, Z_1, Z_2)$ 询问：C 维护一个初始为空的列表 L_{H_2}，其中存储的元组格式为 $(ID_i, T_i, T_B, Z_1, Z_2, sk)$。如果查询的元组在 L_{H_2} 中，则返回 sk 给 A_1。否则按如下操作：

C 在列表 L_S 中找形如 $(ID_i, T_i, T_B, pk_i, sk)$ 的元祖，若找到，则计算

$$\overline{Z_1} = Z_1 - (k_i + x_i) T_B$$

C 检查分别输入 $(T_i, P_{pub_B}, \overline{Z_1})$，$(T_i, T_B, Z_2)$，DDH 预言机是否输出 1。如果 Z_1 和 Z_2 计算正确，在 L_{H_2} 中存储 $(ID_i, T_i, T_B, Z_1, Z_2, sk)$，$sk$ 为 L_S 中的值。如果计算错误，则随机选 $sk \in K$，在 L_{H_2} 中存储 $(ID_i, T_i, T_B, Z_1, Z_2, sk)$，并返回 sk 给 A_1。

若没有找到，则随机选 $sk \in K$，在 L_{H_2} 中存储 $(ID_i, T_i, T_B, Z_1, Z_2, sk)$，并返回 sk 给 A_1。

RevealPartialPrivateKey(i)：C 查找列表 L_C，把相应的部分私钥 k_i 返回给 A_1。

RevealEphemeralKey($\prod_{i,j}^s$)：如果 $\prod_{i,j}^s = \prod_{u,B}^t$，或者 $\prod_{i,j}^s = \prod_{B,u}^t$，则 C 停止模拟，否则 C 发送临时密钥给 A_1。

Send($\prod_{i,j}^s$,m)：C 维护一个初始为空的列表 L_S，其中存储的元组格式为 $(ID_i, T_i, T_B, pk_i, sk)$。

如果 $\prod_{i,j}^s = \prod_{u,B}^t$，C 返回 $T_i = U$ 给 A_1，如果 $\prod_{i,j}^s = \prod_{B,u}^t$，则返回 T_B 给 A_1。否则按协议规则进行回答。

除以上询问外，其它情况和 1)中相同。

假设 A_1 能够赢该游戏，则 A_1 一定计算出了准确的 Z_1 和 Z_2。C 有 $\frac{1}{n_2}$ 的概率在 L_{H_2} 中找到正确的元组。$CDH(U, V) = Z_2$。

则 C 解决 CDH 问题的优势 $Adv_C^{CDH} \geq \frac{1}{n_0 n_1 n_2} Adv_{A_1}$，C 以不可忽略的优势解决了 CDH 问题，这与 CDH 假设冲突。

引理 2：由于 CDH 问题是困难问题，则本文协议在 A_2 的攻击下是安全的。

证明：假设 A_2 能以不可忽略的优势 Adv_{A_2} 赢得 2.3 节中定义的游戏。那么挑战者 C 可以利用 A_2 的能力解决 CDH 问题。

C 随机选 $s \in Z_q^*$，设置 sP 作为用户所在域(域 A)的系统公钥，并发送 s 给 A_2。其它参数的设置和引理 1 的证明相同。

将敌手攻击的情况分为以下几种来讨论：

不知道用户的私有秘密值和域 B 的临时密钥。

Create(i)：C 维护一个初始为空的列表 L_C，其中存储的元组格式为 $(ID_i, k_i, G_i, x_i, P_i)$，如果 $i = u$，C 随机选 $g_i, h_i \in Z_q^*$，计算 $G_i = g_i P$，$P_i = U$，$k_i = s + g_i h_i$，设置 $H_1(ID_i \| G_i) = h_i$，并在 L_C 中存储 $(ID_i, k_i, G_i, \perp, P_i)$，在 L_{H_1} 中存储 (ID_i, G_i, h_i)。如果 $i \neq u$，随机选 $x_i, g_i, h_i \in Z_q^*$，计算 $G_i = g_i P$，$P_i = x_i P$，$k_i = s + g_i h_i$，设置 $H_1(ID_i \| G_i) = h_i$，并在 L_C 中存储 $(ID_i, k_i, G_i, x_i, P_i)$，在 L_{H_1} 中存储 (ID_i, G_i, h_i)。

$H_2(ID_i,T_i,T_B,Z_1,Z_2)$ 询问：C 维护一个初始为空的列表 L_{H_2}，其中存储的元组格式为 $(ID_i,T_i,T_B,Z_1,Z_2,sk)$。如果查询的元组在 L_{H_2} 中，则返回 sk 给 A_2。否则按如下操作：

如果 $i=u$，C 在列表 L_S 中找形如 (ID_i,T_i,T_B,pk_i,sk) 的元祖，若找到，则计算

$$\overline{Z_1}=Z_1-k_i\left(T_B+t_iP_{pub_B}\right)$$

$$\overline{Z_2}=Z_2-k_it_iT_B$$

C 检查分别输入 $\left(P_i,T_B+t_iP_{pub_B},\overline{Z_1}\right)$，$\left(P_i,t_iT_B,\overline{Z_2}\right)$，DDH 预言机是否输出 1。如果 Z_1 和 Z_2 计算正确，在 L_{H_2} 中存储 $(ID_i,T_i,T_B,Z_1,Z_2,sk)$，$sk$ 为 L_S 中的值。如果计算错误，则随机选 $sk\in K$，在 L_{H_2} 中存储 $(ID_i,T_i,T_B,Z_1,Z_2,sk)$，并返回 sk 给 A_2。

如果 $i\neq u$，在列表 L_S 中找形如 (ID_i,T_i,T_B,pk_i,sk) 的元祖，若找到，则在 L_{H_2} 中存储 $(ID_i,T_i,T_B,Z_1,Z_2,sk)$，$sk$ 为 L_S 中的值。否则随机选 $sk\in K$，在 L_{H_2} 中存储 $(ID_i,T_i,T_B,Z_1,Z_2,sk)$，并返回 sk 给 A_2。

RevealMasterKey：C 把主私钥返回给 A_2。

RevealPartialPrivateKey(i)：C 把部分私钥 k_i 返回给 A_2。

RevealSecretValue(i)：如果 $i=u$，则停止模拟，否则 C 查找 L_C 列表，若找到形如 $(ID_i,*,*,*,*)$ 的元组，则返回 x_i 给 A_2。否则执行 Create(i)，返回 x_i 给 A_2。

Send($\prod_{i,j}^{s}$,m)：C 维护一个初始为空的列表 L_S，记录的元组格式为 (ID_i,T_i,T_B,pk_i,sk)。

如果 $\prod_{i,j}^{s}=\prod_{u,B}^{t}$，则返回 $T_B=V$ 给 A_2。

如果 $i=u$，C 随机选择 $t_i\in Z_q^*$，并计算

$$\overline{Z_1}=Z_1-k_i\left(T_B+t_iP_{pub_B}\right)$$

$$\overline{Z_2}=Z_2-k_it_iT_B$$

C 检查分别输入 $\left(P_i,T_B+t_iP_{pub_B},\overline{Z_1}\right)$，$\left(P_i,t_iT_B,\overline{Z_2}\right)$，DDH 预言机是否输出 1。如果 Z_1 和 Z_2 计算正确，则在 L_S 中记录 (ID_i,T_i,T_B,pk_i,sk)，sk 为 L_{H_2} 中的值。否则，随机选 $sk\in K$，在 L_S 中存储 (ID_i,T_i,T_B,pk_i,sk)。

如果 $i\neq u$，则按协议规则进行回答。

除以上询问外，其它情况和引理 1 的证明 1)中相同。

假设 A_2 能够赢得该游戏，则 A_2 一定计算出了准确的 Z_1 和 Z_2。C 有 $\frac{1}{n_2}$ 的概率在 L_{H_2} 中找到正确的元组。C 计算：

$$\overline{Z_2}=Z_2-k_it_iT_B=t_iCDH(U,V)$$

$$CDH(U,V)=t_i^{-1}\left(Z_2-k_it_iV\right)$$

则 C 解决 CDH 问题的优势 $Adv_C^{CDH}\geq\frac{1}{n_0n_1n_2}Adv_{A_2}$，C 以不可忽略的优势解决了 CDH 问题，这与 CDH 假设冲突。

剩余情况和引理 1 类似。

由以上引理的证明，可知本文所提出的协议在 eCK 模型下是安全的。

4. 性能分析与比较

本文协议和文献[8] [10]的协议在构造上有差别，所以从方案的交互次数、加解密次数方面比较了这

Table 1. Performance comparison of different solutions
表 1. 各方案性能比较

性能	文献[8]	文献[10]	本文方案
方案交互次数	5	8	4
加解密次数	3	5	2
密钥托管问题	是	是	否

三个方案。

和文献[8]和文献[10]相比，本文解决了密钥托管问题，而且交互次数和加解密次数都相应的减少。文献[8]基于零知识证明实现了方案的匿名性和可追踪性，但是缺乏对所提出方案详细的安全性证明。文献[10]所提出协议的参与方有四方，交互过于复杂，且对于方案的安全性也是给出了相应的分析缺乏详尽的证明。分析说明，本文的方案保证了安全性的同时，还降低了开销。

5. 结语

本文基于无证书提出了一个云计算下的跨域认证密钥交换协议。实现了用户的跨域认证，降低了服务器的压力，充分保证了用户的密钥安全，与同类协议相比(表 1)，提高了认证和密钥协商的效率。

基金项目

国家自然科学基金(10007016201201)。

参考文献

[1] Buyya, R., Yeo, C.S. and Venugopal, S. (2008) Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities. *IEEE International Conference on High PERFORMANCE Computing and Communications*, 10-1016.

[2] 吴吉义, 傅建庆, 平玲娣, 谢琪. 一种对等结构的云存储系统研究[J]. 电子学报, 2011, 39(5): 1100-1107.

[3] 房晶, 吴昊, 白松林. 云计算安全研究综述[J]. 电信科学, 2011, 27(4): 37-42.

[4] 段志强. 混合云安全策略研究[J]. 计算机安全, 2014(8): 33-37.

[5] 张建晓. 身份认证技术及其发展趋势[J]. 信息通信, 2015(2): 125-126.

[6] 李文敏. 认证密钥协商协议的设计与应用[D]: [博士学位论文]. 北京: 北京邮电大学, 2012.

[7] 李小标. 跨域认证关键技术研究[D]: [博士学位论文]. 北京: 北京邮电大学, 2011.

[8] Xu, C. and He, J. (2015) A Cross-Domain Authentication Method for Cloud Computing. *International Journal of Security & Its Applications*, **9**, 285-292. https://doi.org/10.14257/ijsia.2015.9.3.22

[9] Castiglione, A., Palmieri, F., Chen, C.L., *et al.* (2016) A Blind Signature-Based Approach for Cross-Domain Authentication in the Cloud Environment. *International Journal of Data Warehousing & Mining*, **12**, 34-48. https://doi.org/10.4018/IJDWM.2016010103

[10] 解福. 云计算环境中认证与密钥协商关键技术研究[D]: [博士学位论文]. 济南: 山东师范大学, 2014.

[11] Al-Riyami, S.S. and Paterson, K.G. (2003) Certificateless Public Key Cryptography. In: *Advances in Cryptology-ASIACRYPT* 2003, Springer, Berlin, Heidelberg, 452-473.

[12] 田静. eCK 模型下认证密钥交换协议及其证明[D]: [硕士学位论文]. 沈阳: 东北大学, 2011.

[13] 翁江. 椭圆曲线密码中双线性对与离散对数问题研究[D]: [硕士学位论文]. 郑州: 解放军信息工程大学, 2012.

[14] 李贵莹. 基于无证书的两方认证密钥协商协议的研究与分析[D]: [硕士学位论文]. 济南: 山东大学, 2014.

[15] Lippold, G., Boyd, C. and Nieto, J.G. (2009) Strongly Secure Certificateless Key Agreement. In: *Pairing-Based Cryptography—Pairing* 2009, Springer, Berlin, Heidelberg, 206-230.

[16] 周彦伟，杨波，张文政. 一种改进的无证书两方认证密钥协商协议[J]. 计算机学报, 2017, 40(5): 1181-1191.

[17] 苏航，刘建伟，陶芮. 无证书的层次认证密钥协商协议[J]. 通信学报, 2016, 37(7): 161-171.

[18] He, D., Padhye, S. and Chen, J. (2012) An Efficient Certificateless Two-Party Authenticated Key Agreement Protocol. *Computers and Mathematics with Applications*, **64**, 1914-1926. https://doi.org/10.1016/j.camwa.2012.03.044