doi:10.11959/j.issn.1000-436x.2016145

无证书的层次认证密钥协商协议

苏航，刘建伟，陶芮
（北京航空航天大学电子信息工程学院，北京 100191）

摘 要：提出了一种无证书的层次认证密钥协商协议，协议的安全性基于计算性 Diffie-Hellman 困难假设，并在 eCK（extended Canetti–Krawczyk）模型下证明了该协议的安全性。该协议中，根 PKG 为多层的域 PKG 验证身份并生成部分私钥，域 PKG 为用户验证身份并生成部分私钥，私钥则由用户选定的秘密值和部分私钥共同生成。与已有协议相比，协议不含双线性映射配对运算，且具有较高的效率。
关键词：无证书；层次认证密钥协商协议；计算性 Diffie-Hellman 困难假设；eCK 模型
中图分类号：TN918.1 **文献标识码**：A

Hierarchical certificateless authenticated key agreement protocol

SU Hang, LIU Jian-wei, TAO Rui
(School of Electronic and Information Engineering, Beihang University, Beijing 100191, China)

Abstract: A hierarchical certificateless authenticated key agreement protocol was proposed, and the proposed protocol was proved secure in extended Canetti-Krawczyk (eCK) model, the security of the protocol was based on the computational Diffie-Hellman assumption. In the protocol, a root PKG authenticates the identity and generates a partial private key for lower-level PKG which authenticate the identity and generate a partial private key for users, private key generated by partial private key and user selected secret value. Comparing with the existing protocols, the protocol is efficient without bilinear pairings computation.
Key words: certificateless, hierarchical authenticated key agreement protocol, computational Diffie-Hellman assumption, eCK model

1 引言

密钥协商是保障节点间安全通信的重要机制。在传统公钥基础设施[1]（PKI, public key infrastructure）中，需要证书来验证用户的身份。因此，PKI 中涉及大量的证书管理问题。为了简化这一问题，Shamir[2]于 1984 年提出基于身份的密码体制（IBC, ID based cryptography）。在该体制中，选取用户的身份作为公钥，私钥由可信的私钥生成中心 PKG 生成。然而在 IBC 中，PKG 可掌握所有用户的私钥，这就是 IBC 中固有的密钥托管问题。在 2003 年，Al-Riyami 和 Paterson[3]提出了无证书的公钥密码体制（CLPKC, certificateless public key cryptography），这一体制解决了 IBC 中的密钥托管问题。利用 CLPKC，学者们提出了大量的无证书认证密钥协商协议[4~10]。

在空间信息网络中[11]，含有多种异构网络和大量网络节点，集中式的管理模式会因数据流过于集中，而导致网络拥塞和服务延时及单点失效问题。空间节点的软硬件处理能力相对较低，处理资源和带宽相对较少，电源和推进能力有限。对于部分实时性要求较高的任务，通常不允许在运算过程中耗费过多时间。上述机制均是在单一 PKG 环境下提出的，不适用于含有大量节点的空间信息网。Gentry

收稿日期：2015-11-04；**修回日期**：2016-01-20
基金项目：国家重点基础研究发展计划（“973”计划）基金资助项目（No.2012CB315905）；国家自然科学基金资助项目（No.61272501）；中央高校基本科研业务费专项基金资助项目（No.YWF-15-GJSYS-059）
Foundation Items: The National Basic Research Program of China (973 Program) (No.2012CB315905), The National Natural Science Foundation of China (No.61272501), The Fundamental Research Funds for the Central Universities (No.YWF-15-GJSYS-059)

等[12]提出了基于身份的层次加密体制（HIBC, hierarchical ID based cryptography），该体制中包含一个根 PKG 及多层的域 PKG，根 PKG 对域 PKG 进行验证并为其生成私钥，上层域 PKG 验证下层域 PKG 并为其生成私钥，直至用户的上一层域。利用 HIBC 体制，Cao 等[13]和 Liu 等[14]分别提出了基于身份的层次认证密钥协商协议，不过这些协议没有解决 IBC 固有的密钥托管问题。Chow、Roth 和 Rieffel[15]于 2008 年首次对无证书的分层密码体制（HCLC, hierarchical certificateless cryptography）进行了研究。这一体制既保留了 HIBC 体制的优点，又避免了 HIBC 体制中的密钥托管问题。

本文基于 HCLC 体制，提出一种无证书的层次认证密钥协商协议，并在 eCK 模型下证明了其安全性。在本文协议中，多层 PKG 有效防止了单点失效问题，减轻了 PKG 的运行压力，提高了系统的承载能力，而且，解决了 IBC 体制中固有的密钥托管问题。协议的计算开销与双方用户所处层级呈线性关系，运算过程不含双线性对运算，具有较高的效率，适用于计算能力小的空间节点。

2 预备知识

2.1 椭圆曲线

椭圆曲线 $\frac{E}{\mathbb{F}_P}$ 可用等式表示为[16]

$$y^2 \equiv x^3 + ax + b \bmod p,\ a,b \in \mathbb{F}_P \text{ 且}$$

$$4a^3 + 27b^2 \neq 0 \bmod p$$

群 $\mathbb{G} = \{(x,y) \mid x,y \in \mathbb{F}_P, (x,y) \in \frac{E}{\mathbb{F}_P}\} \cup \{O\}$，$O$ 为无穷远点。

群 $\mathbb{G}$ 为循环加法群，群运算为加法运算（点乘运算），描述如下

$$kP = \underbrace{P + P + \cdots + P}_{k}, k \in \mathbb{Z}_q^*$$

2.2 困难问题及假设

计算性 Diffie-Hellman 问题（CDH, computational Diffie-Hellman problem）：$\mathbb{G}$ 为 q 阶的循环加法群，给定 $P, aP, bP \in \mathbb{G}$（$a,b \in \mathbb{Z}_q^*$ 是未知的随机数），计算 $abP \in \mathbb{G}$。

定义 1 多项式时间算法 $\mathcal{A}$ 在安全常数 λ 下解决 CDH 问题的优势定义如下

$$Adv_{\mathcal{A}}^{\mathrm{CDH}}(\lambda) = \Pr[\mathcal{A}(P, aP, bP) = abP : P \in \mathbb{G}, a,b \in \mathbb{Z}_q^*]$$

定义 2 CDH 假设指出，不存在多项式时间算法 $\mathcal{A}$，使其具有不可忽略的优势

$$Adv_{\mathcal{A}}^{\mathrm{CDH}}(\lambda) \geqslant \varepsilon$$

可以解决 CDH 问题。

2.3 无证书的层次密钥协商协议定义

结合无证书的密钥协商协议[4~10]及无证书的层次密码体制定义[15]，本节给出无证书的层次认证密钥协商协议的定义。

1) $(pp, msk) \leftarrow$ Root-Setup(λ)：系统建立算法 Root-Setup 以安全常数 $\lambda \in \mathbb{N}$ 作为输入，输出全局性系统参数 pp 及主私钥 msk。

2) $(k) \leftarrow$ Partial-Private-Key-Extract(msk,ID)：部分私钥生成算法 Partial-Private-Key-Extract 以主密钥 msk 和任意一个用户身份向量 $\boldsymbol{ID} = (I_1, I_2, \cdots, I_t)$ 作为输入，输出为该用户 $\boldsymbol{ID}$ 的部分私钥 k。

3) $(x) \leftarrow$ Set-Secret-Value(pp, $\boldsymbol{ID}$)：秘密值生成算法 Set-Secret-Value 以全局性系统参数 pp 和用户 $\boldsymbol{ID}$ 作为输入，输出为该用户 $\boldsymbol{ID}$ 的私有秘密值 x。

4) $(d) \leftarrow$ Set-Private-Key(k, x)：私钥生成算法 Set-Private-Key 以部分私钥 k 和用户的私有秘密值 x 作为输入，输出为该用户 $\boldsymbol{ID}$ 的私钥 d。

5) $(pk) \leftarrow$ Set-Public-Key(pp,x)：公钥生成算法 Set-Public-Key 以全局性系统参数 pp 和用户 $\boldsymbol{ID}$ 的私有秘密值 x 作为输入，输出为用户的公钥。

6) $(k) \leftarrow$ Partial-Delegate(k', $\boldsymbol{ID}$)：部分私钥委托算法 Partial-Delegate 以用户身份向量 $\boldsymbol{ID}' = (I_1, I_2, \cdots, I_{t-1})$ 的部分私钥 k'、用户身份向量 $\boldsymbol{ID} = (I_1, I_2, \cdots, I_t)$ 作为输入，输出为用户 $\boldsymbol{ID}$ 的部分私钥 k。

7) $(T) \leftarrow$ Temporary(d)：临时信息生成算法 Temporary 以用户私钥 d 作为输入，输出为用于密钥协商阶段的临时信息 T。

8) $(sk) \leftarrow$ Agreement(pk', T', d)：密钥协商算法 Agreement 以参与密钥协商的用户的私钥 d 和另一用户的公钥 pk' 及其临时信息 T' 作为输入，输出为协商的会话密钥 sk。

2.4 安全模型

在无证书的密钥协商协议中存在 2 种类型的敌手[6]，其具有的攻击能力如下。

敌手 1 敌手 $\mathcal{A}_1$ 是一个不诚实的参与者，他不能获取系统的主私钥，但是可以任意替换参与密钥协商的用户的私有秘密值和公钥。

敌手 2 敌手 $\mathcal{A}_2$ 是一个恶意的 PKG，但是不可

以替换参与密钥协商用户的私有秘密值和公钥。

Lippold 等[17]将传统的 eCK 模型扩展成无证书体制下的 eCK 模型。每个参与者被模拟为多项式时间图灵机，可以并行地执行多项式通信会话，称会话 $\prod_{I,J}^{s}$ 为参与者 I 和 J 的第 s 个会话。此模型是通过挑战者 $\mathcal{C}$ 和敌手 $\mathcal{A}$ 之间的游戏来定义的，游戏可分为 2 个阶段。

阶段 1　$\mathcal{A}$ 被允许以任意顺序做如下查询。

Creat(ID_i)：$\mathcal{C}$ 是身份为 ID_i 的参与者生成公私钥对。称不被 $\mathcal{A}$ 控制的用户为诚实用户。

RevealMasterKey：$\mathcal{C}$ 返回系统主私钥给 $\mathcal{A}$ 。

RevealSessionKey($\prod_{I,J}^{s}$)：如果会话 $\prod_{I,J}^{s}$ 已经生成会话密钥，则返回会话密钥给 $\mathcal{A}$ ，否则返回空值。

RevealPartialPrivateKey(ID_i)：$\mathcal{C}$ 返回 ID_i 的部分私钥给 $\mathcal{A}$ 。

RevealSecretValue(ID_i)：$\mathcal{C}$ 返回 ID_i 的私有秘密值给 $\mathcal{A}$ 。

ReplacePublicKey(ID_i, pk)：$\mathcal{C}$ 将 ID_i 的私有秘密值和公钥替换为 $\mathcal{A}$ 选定的值。

RevealEphemeralKey($\prod_{I,J}^{s}$)：$\mathcal{C}$ 返回 ID_i 的临时私钥给 $\mathcal{A}$ 。

Send($\prod_{I,J}^{s}, m$)：$\mathcal{A}$ 向会话 $\prod_{I,J}^{s}$ 发送消息 m，$\prod_{I,J}^{s}$ 按协议规定作出回答。若 m 为空，则该会话作为发起者发起一次会话；否则，它担任响应者的角色。

一旦 $\mathcal{A}$ 认为第一阶段可以结束了，则选择一个新鲜会话开始第 2 阶段的游戏。

阶段 2　Test($\prod_{I,J}^{s}$)：$\mathcal{A}$ 针对已经结束的新鲜会话 $\prod_{I,J}^{s}$ 进行唯一一次的 Test 查询。Test 查询随机选择一比特 $b\in\{0,1\}$ 。如果 $b=0$ ，返回真实的会话密钥；如果 $b=1$ ，返回与真实的会话密钥同分布的一个随机值。

在游戏的最后，$\mathcal{A}$ 需要输出一比特 $b'\in\{0,1\}$ 。如果 $b'=b$ ，则 $\mathcal{A}$ 赢得此次游戏。定义 $\mathcal{A}$ 赢得游戏的优势为

$$Adv_{\mathcal{A}}=\left|\Pr(b'=b)-\frac{1}{2}\right|$$

定义 3　匹配会话。若 2 个会话 $\prod_{I,J}^{s}$ 和 $\prod_{J,I}^{r}$ 在一次协议运行完成后，得到相同的会话识别符 SID，那么它们互称为匹配会话。其中，会话识别符 SID 为会话所发送和接收的消息及会话发起者和响应者身份的串联。

定义 4　新鲜会话。令 $\prod_{I,J}^{s}$ 为诚实参与者 I 和 J 已经结束的会话。如果下列所有条件都不成立，则称 $\prod_{I,J}^{s}$ 是新鲜的。

1) $\mathcal{A}$ 查询了 $\prod_{I,J}^{s}$ 或者其匹配会话（如果存在）的会话密钥。

2) 如果 $\prod_{I,J}^{s}$ 有匹配会话 $\prod_{J,I}^{r}$，$\mathcal{A}$ 或者查询了 I 的部分私钥或私有秘密值和 $\prod_{I,J}^{s}$ 的临时私钥；或者查询了 J 的部分私钥或私有秘密值和 $\prod_{J,I}^{r}$ 的临时私钥。

3) 如果 $\prod_{I,J}^{s}$ 没有匹配会话，$\mathcal{A}$ 或者查询了 I 的部分私钥或私有秘密值和 $\prod_{I,J}^{s}$ 的临时私钥；或者查询了 J 的部分私钥。

定义 5　安全性。若一个无证书密钥协商协议满足下列条件。

1) 在 $\prod_{I,J}^{s}$ 和其匹配会话 $\prod_{J,I}^{r}$ 之间有一忠实转发消息的良性攻击者，$\prod_{I,J}^{s}$ 和 $\prod_{J,I}^{r}$ 能计算得到相同的随机均匀分布在会话密钥空间上的会话密钥。

2) $\mathcal{A}$ 赢得游戏的优势 $Adv_{\mathcal{A}}$ 是可以忽略的。

那么本文称该协议在 eCK 模型下是安全的。

3　协议设计

3.1　协议描述

本文提出的无证书的层次认证密钥协商协议包括 Root-Setup、Partial-Private-Key-Extract、Set-Secret-Value、Set-Private-Key、Set-Public-Key、Partial-Delegate、Temporary、Agreement 这 8 个部分，协议的具体构造过程如下。

1) $(pp, msk)\leftarrow$ Root-Setup(λ)：系统建立算法选取满足安全常数 λ 的阶为 q 的椭圆曲线循环加法群 $\mathbb{G}$，即 $|q|=\lambda$，$\mathbb{G}$ 的生成元为 P。选取安全的散列函数：$H_1:\{0,1\}^*\times\mathbb{G}\rightarrow\mathbb{Z}_q^*$，$H_2:\{0,1\}^*\times\mathbb{G}\times\mathbb{G}\rightarrow\mathcal{K}$，其中，$\mathcal{K}$ 为会话密钥空间。选取主私钥 $msk=s$，计算公钥 $P_{\text{pub}}=sP$。输出共享的全局性系统参数

$$pp=\{\mathbb{G},q,P,P_{\text{pub}},H_1,H_2\}$$

2) $(k)\leftarrow$ Partial-Private-Key-Extract (msk, $\boldsymbol{ID}$)：给定主私钥 msk 和任意一个用户身份向量 $\boldsymbol{ID}=(I_1,I_2,\cdots,I_t)$，部分私钥生成算法随机选取

$g_1,\cdots,g_t\in\mathbb{Z}_q^*$，计算 $r_i=H_1(I_i\|g_iP)$，其中，$1\leqslant i\leqslant t$。输出该用户所对应的部分私钥

$$k=s+\sum_{i=1}^{t}(g_ir_i)$$

若 $k=0$，则需要重新选取 $g_1,\cdots,g_t\in\mathbb{Z}_q^*$。通过安全信道将 $\{g_1P,\cdots,g_tP,k\}$ 发送给用户 ***ID***，其中，$g_1P,\cdots,g_tP$ 为用户 ***ID*** 的部分公钥。用户验证等式为

$$kP=P_{\text{pub}}+\sum_{i=1}^{t}(H_1(I_i\|g_iP)g_iP)$$

若等式不成立，则拒绝此部分私钥。

3) $(x)\leftarrow$ Set-Secret-Value(pp, ***ID***)：给定全局性系统参数 pp，秘密值生成算法随机选取 $x\in\mathbb{Z}_q^*$，输出用户 ***ID*** 的私有秘密值为 x。

4) $(d)\leftarrow$ Set-Private-Key(k, x)：给定用户 ***ID*** 的部分私钥 k 和私有秘密值 x，输出该用户 ***ID*** 的私钥

$$d=k+x$$

若 $d=0$，则需要重新执行 Set-Secret-Value 算法。

5) $(pk)\leftarrow$ Set-Public-Key(pp, x)：给定全局性系统参数 pp 和用户 ***ID*** 的私有秘密值 x，输出该用户 ***ID*** 的公钥

$$pk=\{\boldsymbol{ID},g_1P,\cdots,g_tP,xP\}$$

6) $(k)\leftarrow$ Partial-Delegate(k',***ID***)：部分私钥委托算法由用户 $\boldsymbol{ID}=(I_1,I_2,\cdots,I_t)$ 的上层 PKG 运行，其中，$\boldsymbol{ID}_{\text{PKG}}=(I_1,I_2,\cdots,I_{t-1})$，PKG 的部分私钥为 $k'=s+\sum_{i=1}^{t-1}(g_ir_i)$，部分公钥为 $\{g_1P,\cdots,g_{t-1}P\}$。随机选取 $g_t\in\mathbb{Z}_q^*$，计算 $r_t=H_1(I_t\|g_tP)$。PKG 为用户 ***ID*** 生成部分私钥

$$k=k'+g_tr_t=s+\sum_{i=1}^{t-1}(g_ir_i)+g_tr_t=s+\sum_{i=1}^{t}(g_ir_i)$$

若 $k=0$，则需要重新选取 $g_t\in\mathbb{Z}_q^*$。通过安全信道将 $\{g_1P,\cdots,g_tP,k\}$ 发送给用户 ***ID***，其中，$g_1P,\cdots,g_tP$ 为用户 ***ID*** 的部分公钥。用户验证等式为

$$kP=P_{\text{pub}}+\sum_{i=1}^{t}(H_1(I_i\|g_iP)g_iP)$$

若等式不成立，则拒绝此部分私钥。若等式成立，用户可执行 Set-Secret-Value、Set-Private-Key 和 Set-Public-Key 算法生成自己的私钥和公钥。

以用户 A 和 B 为例，其中，用户 A 所处的层级为 l_A，$\boldsymbol{ID}_A=(I_1,I_2,\cdots,I_{l_A})$，$A$ 的私钥为 d_A，公钥 pk_A 为 $\{\boldsymbol{ID}_A,g_1P,\cdots,g_{l_A}P,x_AP\}$。用户 B 所处的层级为 l_B，$\boldsymbol{ID}_B=(I_1',I_2',\cdots,I_{l_B}')$，$B$ 的私钥为 d_B，公钥 pk_B 为 $\{\boldsymbol{ID}_B,g_1'P,\cdots,g_{l_B}'P,x_BP\}$。

7) $(T)\leftarrow$ Temporary(d)：用户 A 和 B 分别执行临时信息生成算法，A 随机选取临时私钥 $t_A\in\mathbb{Z}_q^*$，计算 $T_A=t_Ad_AP$，发送 $\{T_A,pk_A\}$ 给 B。B 随机选取临时私钥 $t_B\in\mathbb{Z}_q^*$，计算 $T_B=t_Bd_BP$，发送 $\{T_B,pk_B\}$ 给 A。

8) $(sk)\leftarrow$ Agreement(pk',T',d)：用户 A 和 B 分别执行密钥协商算法，计算会话密钥。

A 计算

$$k_{AB}=d_A(T_B+t_A(P_{\text{pub}}+x_BP+\sum_{i=1}^{l_B}(H_1(I_i'\|g_i'P)g_i'P)))$$

$$t_Ad_AT_B=t_At_Bd_Ad_BP$$

会话密钥

$$sk_A=H_2(ID_A\|ID_B\|k_{AB}\|t_At_Bd_Ad_BP)$$

B 计算

$$k_{BA}=d_B(T_A+t_B(P_{\text{pub}}+x_AP+\sum_{i=1}^{l_A}(H_1(I_i\|g_iP)g_iP)))$$

$$t_Bd_BT_A=t_At_Bd_Ad_BP$$

会话密钥

$$sk_B=H_2(ID_A\|ID_B\|k_{BA}\|t_At_Bd_Ad_BP)$$

3.2 协议正确性

要证明协议中用户 A 和 B 获得相同的会话密钥，即 $sk_A=sk_B$，只需证明 $k_{AB}=k_{BA}$，证明如下。

用户 A 做如下计算

$$k_{AB}=d_A(T_B+t_A(P_{\text{pub}}+x_BP+\sum_{i=1}^{l_B}(H_1(I_i'\|g_i'P)g_i'P)))$$

$$=d_A(t_Bd_BP+t_Ad_BP)=(t_A+t_B)d_Ad_BP$$

用户 B 做如下计算

$$k_{BA}=d_B(T_A+t_B(P_{\text{pub}}+x_AP+\sum_{i=1}^{l_A}(H_1(I_i\|g_iP)g_iP)))$$

$$=d_B(t_Ad_AP+t_Bd_AP)=(t_A+t_B)d_Ad_BP=k_{AB}$$

因此，用户 A 与用户 B 可计算获得相同的会话密钥。

4　安全性分析

本节在基于身份的eCK模型[7]下证明本文协议的安全性，在证明过程中将模拟根 PKG 为不同层次的节点生成部分私钥的功能，密钥托管功能可由相同方式模拟。假定散列函数 H_1 和 H_2 是随机预言机，由定义可知匹配会话 $\prod_{I,J}^{s}$ 和 $\prod_{J,I}^{r}$ 可计算得到相同的会话密钥。

4.1　引理 1 的证明

引理 1　假定CDH假设成立，则本文协议在 $\mathcal{A}_{\mathrm{I}}$ 的攻击下在 eCK 模型下是安全的。

证明　假设 $\mathcal{A}_{\mathrm{I}}$ 以不可忽略的优势 Adv_{A1} 在多项式时间内赢得在 2.4 节中定义的游戏，那么 $\mathcal{C}$ 可利用 $\mathcal{A}_{\mathrm{I}}$ 的能力解决 CDH 问题。$\mathcal{C}$ 选择 $s\in\mathbb{Z}_q^*$，并令 $P_{\mathrm{pub}}=P_0=sP$。令全局系统参数 $pp=\{\mathbb{G},q,P,P_0,H_1,H_2\}$，并发送 pp 给敌手 $\mathcal{A}_{\mathrm{I}}$。

令 n_0 表示每个参与者最多发起的会话数，假定敌手 $\mathcal{A}_{\mathrm{I}}$ 最多激活 n_1 个诚实用户，最多做 n_2 次散列 H_2 查询。当 $\mathcal{A}_{\mathrm{I}}$ 做 Test 查询后，只有 3 种方法来赢得此游戏。

1) 猜测攻击：$\mathcal{A}_{\mathrm{I}}$ 直接猜测出会话密钥。

2) 密钥复制攻击：$\mathcal{A}_{\mathrm{I}}$ 使 Test 会话的一个非匹配会话拥有和 Test 会话一样的会话密钥，通过查询该非匹配会话的会话密钥赢得游戏。

3) 伪造攻击：在某时刻，$\mathcal{A}_{\mathrm{I}}$ 向 $\mathcal{C}$ 进行了值为 $(I\|J\|Z_1\|Z_2)$ 的 H_2 查询。此时，$\mathcal{A}_{\mathrm{I}}$ 计算出了正确的 Z_1 和 Z_2 值。

通过协议构造和 2.4 节中的定义可知，猜测攻击的优势可以忽略。而由于 H_2 中的会话标识符包含了通信双发的身份及发送和接受的消息，非匹配会话不可能拥有与 Test 会话相同的会话密钥，密钥复制攻击的优势也可忽略。因此，只需考虑伪造攻击的情形。

令 $Adv_{\mathcal{C}}$ 为 $\mathcal{C}$ 解决 CDH 问题的优势。给定 CDH 问题实例 $U=uP,V=vP$，$\mathcal{C}$ 的任务是计算 $W=CDH(U,V)=uvP$。$\mathcal{C}$ 模拟 2.4 节定义的游戏，对 $\mathcal{A}_{\mathrm{I}}$ 的查询做出以下回答。

首先，$\mathcal{C}$ 选择 $\prod_{I,J}^{T}$ 作为 Test 会话，其中，$I=(I_1,I_2,\cdots,I_{l_I}),J=(I_1',I_2',\cdots,I_{l_J}'),I\neq J,l_I>1$，$l_J>1,T\in\{1,\cdots,n_0\}$，则 $\prod_{I,J}^{T}$ 为 Test 会话的概率 $\frac{1}{n_0n_1^2}$。此时本文只需考虑 2 种情形。

1) 存在诚实的用户拥有 Test 会话的匹配会话 $\prod_{J,I}^{L}$。

2) 不存在诚实的用户拥有 Test 会话的匹配会话。

4.1.1　存在诚实的用户拥有 Test 会话的匹配会话 $\prod_{J,I}^{L}$

$\mathcal{A}_{\mathrm{I}}$ 可以通过 ReplacePublicKey 查询获得参与者的私有秘密值 x 和公钥 xP。此时可分为以下 4 种情形。

1) $\mathcal{A}_{\mathrm{I}}$ 既不知道 I 的临时私钥，也不知道 J 的部分私钥

$\mathcal{C}$ 对 $\mathcal{A}_{\mathrm{I}}$ 的查询做出以下回答。

Creat(ID_i)：假设 $\boldsymbol{ID}_i=(I_1,I_2,\cdots,I_{l_i})$，$\mathcal{C}$ 维护初始为空的列表 L_C，记录格式为 $\{\boldsymbol{ID}_i,k_i,R_i^1,\cdots,R_i^{l_i},x_i,P_i\}$ 的元组。$\mathcal{C}$ 按如下规则生成元组数据。

若 $\boldsymbol{ID}_i=J$，$\mathcal{C}$ 在 L_C 列表中找形如 $\{\boldsymbol{ID}_j,*,*,\cdots,*,*,*\}$ 的元组，其中，$\boldsymbol{ID}_j$ 为 $\boldsymbol{ID}_i$ 的上层用户，取其中层级最大的用户，记为 $\boldsymbol{ID}_j=(I_1,I_2,\cdots,I_{l_j}),l_j\leqslant l_i$。选取 L_{H_1} 列表中相应的 r_i^n,R_i^n，随机选择 $r_i^{l_j},\cdots,r_i^{l_i},x_i\in\mathbb{Z}_q^*$。计算

$$R_i^n=(l_i-l_j)^{-1}(U-k_jP)(r_i^n)^{-1},n\in\{l_j,\cdots,l_i\}$$

$$P_i=x_iP$$

令 $H_1(I_n\|R_i^n)=r_i^n,n\in\{1,\cdots,l_i\}$，记录 L_C 元组为 $\{\boldsymbol{ID}_i,\perp,R_i^1,\cdots,R_i^{l_i},x_i,P_i\}$，记录 L_{H_1} 元组为 $\{I_n,R_i^n,r_i^n\}$，$n\in\{1,\cdots,l_i\}$。

若 $\boldsymbol{ID}_i\neq J,l_i=1$，$\mathcal{C}$ 随机选择 $R_i^1,r_i^1,k_i,x_i\in\mathbb{Z}_q^*$。计算

$$P_i=x_iP$$

若 $\boldsymbol{ID}_i\neq J,l_i\neq1$，$\mathcal{C}$ 在 L_C 列表中找形如 $\{\boldsymbol{ID}_j,*,*,\cdots,*,*,*\}$ 的元组，其中，$\boldsymbol{ID}_j$ 为 $\boldsymbol{ID}_i$ 的上层用户，取其中层级最大的用户，记为 $\boldsymbol{ID}_j=(I_1,I_2,\cdots,I_{l_j}),l_j\leqslant l_i$。选取 L_{H_1} 列表中相应的 r_i^n,R_i^n，随机选择 $r_i^{l_j},\cdots,r_i^{l_i},k_i,x_i\in\mathbb{Z}_q^*$。计算

$$R_i^n=(l_i-l_j)^{-1}(k_iP-k_jP)(r_i^n)^{-1},n\in\{l_j,\cdots,l_i\}$$

$$P_i=x_iP$$

令 $H_1(I_n\|R_i^n)=r_i^n,n\in\{1,\cdots,l_i\}$，记录 L_C 元组为 $\{\boldsymbol{ID}_i,k_i,R_i^1,\cdots,R_i^{l_i},x_i,P_i\}$，记录 L_{H_1} 元组为 $\{I_n,R_i^n,r_i^n\},n\in\{1,\cdots,l_i\}$。

$H_1(I_n,R_i^n)$ 查询：$\mathcal{C}$ 维护初始为空的列表 L_{H_1}，记录格式为 $\{I_n,R_i^n,r_i^n\}$ 的元组。若 (I_n,R_i^n) 在列表 L_{H_1} 中，则返回 r_i^n 给 $\mathcal{A}_1$；若不在，则随机选择 $r_i^n\in\mathbb{Z}_q^*$ 返回给 $\mathcal{A}_1$，并记录 $\{I_n,R_i^n,r_i^n\}$。

$H_2(\boldsymbol{ID}_i,\boldsymbol{ID}_j,Z_1,Z_2)$ 查询：$\mathcal{C}$ 维护初始为空的列表 L_{H_2}，记录格式为 $\{\boldsymbol{ID}_i,\boldsymbol{ID}_j,T_i,T_j,pk_i,pk_j,Z_1,Z_2,sk\}$ 的元组。若 L_{H_2} 中有记录则返回对应的 sk 给 $\mathcal{A}_1$；若没有记录，则按如下操作。

若 $\boldsymbol{ID}_i=J$，$\mathcal{C}$ 在 L_S 列表中找 $\{\boldsymbol{ID}_i,\boldsymbol{ID}_j,T_i,T_j,pk_i,pk_j,sk\}$ 的元组。若找到，则计算

$$Z_1'=T_j+t_i(P_0+P_j+\sum_{n=1}^{l_j}(H_1(I_n\|R_j^n)R_j^n))$$

$$\overline{Z_1}=Z_1-x_iZ_1'$$

$$\overline{Z_2}=Z_2-x_it_it_j(k_j+x_j)P$$

$\mathcal{C}$ 检查 $(P_0+\sum_{n=1}^{l_i}(H_1(I_n\|R_i^n)R_i^n))$、$Z_1'$、$\overline{Z_1}$ 是否是 DDH 元组，检查 $(P_0+\sum_{n=1}^{l_i}(H_1(I_n\|R_i^n)R_i^n))$、$t_it_j(k_j+x_j)P$、$\overline{Z_2}$ 是否是 DDH 元组。若是，则 Z_1、Z_2 计算正确，在 L_{H_2} 中记录 $\{\boldsymbol{ID}_i,\boldsymbol{ID}_j,T_i,T_j,pk_i,pk_j,Z_1,Z_2,sk\}$ 的元组，sk 为 L_S 中的值。否则，随机选取 $sk\in\mathcal{K}$，在 L_{H_2} 中记录 $\{\boldsymbol{ID}_i,\boldsymbol{ID}_j,T_i,T_j,pk_i,pk_j,Z_1,Z_2,sk\}$ 的元组。返回 sk 给 $\mathcal{A}_1$。

若 $\boldsymbol{ID}_i\neq J$，$\mathcal{C}$ 在 L_S 列表中找 $\{\boldsymbol{ID}_i,\boldsymbol{ID}_j,T_i,T_j,pk_i,pk_j,sk\}$ 的元组。若找到，则在 L_{H_2} 中记录 $\{\boldsymbol{ID}_i,\boldsymbol{ID}_j,T_i,T_j,pk_i,pk_j,Z_1,Z_2,sk\}$ 的元组，sk 为 L_S 中的值。否则，随机选取 $sk\in\mathcal{K}$，在 L_{H_2} 中记录 $\{\boldsymbol{ID}_i,\boldsymbol{ID}_j,T_i,T_j,pk_i,pk_j,Z_1,Z_2,sk\}$ 的元组。返回 sk 给 $\mathcal{A}_1$。

RevealSecretValue($\boldsymbol{ID}_i$)：$\mathcal{C}$ 查询 L_C 列表，若找到形如 $\{\boldsymbol{ID}_i,*,*,\cdots,*,*,*\}$ 的元组，则返回 x_i 给 $\mathcal{A}_1$；否则执行 Creat($\boldsymbol{ID}_i$)，再返回 x_i 给 $\mathcal{A}_1$。

ReplacePublicKey($\boldsymbol{ID}_i,pk$)：$\mathcal{C}$ 查询 L_C 列表，若找到形如 $\{\boldsymbol{ID}_i,*,*,\cdots,*,*,*\}$ 的元组，则替换 x_i,P_i 为 $\mathcal{A}_1$ 所选值；否则执行 Creat($\boldsymbol{ID}_i$)，再替换 x_i,P_i 为 $\mathcal{A}_1$ 所选值。

RevealEphemeralKey($\prod_{I,J}^s$)：若 $\prod_{I,J}^s=\prod_{I,J}^T$，则 $\mathcal{C}$ 停止模拟；否则，$\mathcal{C}$ 发送临时私钥给 $\mathcal{A}_1$。

RevealPartialPrivateKey($\boldsymbol{ID}_i$)：若 $\boldsymbol{ID}_i=J$，则 $\mathcal{C}$ 停止模拟；否则，$\mathcal{C}$ 发送部分私钥给 $\mathcal{A}_1$。

RevealSessionKey($\prod_{I,J}^s$)：若 $\prod_{I,J}^s=\prod_{I,J}^T$ 或 $\prod_{I,J}^s=\prod_{J,I}^L$，则 $\mathcal{C}$ 停止模拟；否则，$\mathcal{C}$ 发送会话密钥给 $\mathcal{A}_1$。

Send($\prod_{I,J}^s,m$)：$\mathcal{C}$ 维护初始为空的列表 L_S，记录格式为 $\{\boldsymbol{ID}_i,\boldsymbol{ID}_j,T_i,T_j,pk_i,pk_j,sk\}$ 的元组。按如下规则回答 $\mathcal{A}_1$ 的查询。

若 $\prod_{I,J}^s=\prod_{I,J}^T$，则返回 $T_i=V$ 给 $\mathcal{A}_1$。

若 $\boldsymbol{ID}_i=J$，$\mathcal{C}$ 随机选取 $t_i\in\mathbb{Z}_q^*$，并计算

$$Z_1'=T_j+t_i(P_0+P_j+\sum_{n=1}^{l_j}(H_1(I_n\|R_j^n)R_j^n))$$

$$\overline{Z_1}=Z_1-x_iZ_1'$$

$$\overline{Z_2}=Z_2-x_it_it_j(k_j+x_j)P$$

$\mathcal{C}$ 检查 $(P_0+\sum_{n=1}^{l_i}(H_1(I_n\|R_i^n)R_i^n))$、$Z_1'$、$\overline{Z_1}$ 是否是 DDH 元组，检查 $(P_0+\sum_{n=1}^{l_i}(H_1(I_n\|R_i^n)R_i^n))$、$t_it_j(k_j+x_j)P$、$\overline{Z_2}$ 是否是 DDH 元组。若是，则 Z_1、Z_2 计算正确，在 L_S 中记录 $\{\boldsymbol{ID}_i,\boldsymbol{ID}_j,T_i,T_j,pk_i,pk_j,sk\}$ 的元组，sk 为 L_{H_2} 中的值。否则，随机选取 $sk\in\mathcal{K}$，在 L_S 中记录 $\{\boldsymbol{ID}_i,\boldsymbol{ID}_j,T_i,T_j,pk_i,pk_j,sk\}$ 的元组。

否则，按照协议规则进行回答。

Test($\prod_{I,J}^s$)：若 $\prod_{I,J}^s\neq\prod_{I,J}^T$，则 $\mathcal{C}$ 停止模拟；否则，$\mathcal{C}$ 随机选取 $\xi\in\mathcal{K}$，发送 ξ 给 $\mathcal{A}_1$。

假定 $\mathcal{A}_1$ 赢得此次游戏，则 $\mathcal{A}_1$ 必定计算出了正确的 Z_1、Z_2。$\mathcal{C}$ 则有 $\frac{1}{n_2}$ 的概率在 L_{H_2} 中找到对应的正确元组。又因为 $T_I=t_I(k_I+x_I)P=V$，则

$$\begin{aligned}Z_2&=t_It_J(k_I+x_I)(k_J+x_J)P\\&=t_Jx_JV+t_It_J(k_I+x_I)U\\&=t_Jx_JV+t_JW\end{aligned}$$

所以 $CDH(U,V)=(Z_2-t_Jx_JV)t_J^{-1}$。则 $\mathcal{C}$ 解决 CDH 问题的优势 $Adv_{\mathcal{C}}=\frac{1}{n_0n_1^2n_2}\ Adv_{\mathcal{A}_1}$。因为 $Adv_{\mathcal{A}_1}$ 不可忽略，则 $Adv_{\mathcal{C}}$ 也不可忽略。这与 CDH 假设矛盾。

2) $\mathcal{A}_1$ 既不知道 J 的临时私钥，也不知道 I 的部分私钥

这种情形与 1)中证明过程类似，只需交换 I 和 J

即可。

3) $\mathcal{A}_1$ 不知道 I 和 J 的部分私钥

除以下查询的回答不同外，其余查询均按照 1)中的规则进行。

Creat($\boldsymbol{ID}_i$)：假设 $\boldsymbol{ID}_i=(I_1,I_2,\cdots,I_{l_i})$，$\mathcal{C}$ 维护初始为空的列表 L_C，记录格式为 $\{\boldsymbol{ID}_i,k_i,R_i^1,\cdots,R_i^{l_i},x_i,P_i\}$ 的元组。$\mathcal{C}$ 按如下规则生成元组数据。

若 $\boldsymbol{ID}_i=I$，$\mathcal{C}$ 在 L_C 列表中找形如 $\{\boldsymbol{ID}_j,*,*,\cdots,*,*,*\}$ 的元组，其中，$\boldsymbol{ID}_j$ 为 $\boldsymbol{ID}_i$ 的上层用户，取其中层级最大的用户，记为 $\boldsymbol{ID}_j=(I_1,I_2,\cdots,I_{l_j})$，$l_j\leqslant l_i$。选取 L_{H_1} 列表中相应 r_i^n,R_i^n，随机选择 $r_i^{l_j},\cdots,r_i^{l_i},x_i\in\mathbb{Z}_q^*$。计算

$$R_i^n=(l_i-l_j)^{-1}(V-k_jP)(r_i^n)^{-1},n\in\{l_j,\cdots,l_i\}$$

$$P_i=x_iP$$

令 $H_1(I_n\|R_i^n)=r_i^n,n\in\{1,\cdots,l_i\}$，记录 L_C 元组为 $\{\boldsymbol{ID}_i,\perp,R_i^1,\cdots,R_i^{l_i},x_i,P_i\}$，记录 L_{H_1} 元组为 $\{I_n,R_i^n,r_i^n\}$，$n\in\{1,\cdots,l_i\}$。

$H_2(\boldsymbol{ID}_i,\boldsymbol{ID}_j,Z_1,Z_2)$ 查询：$\mathcal{C}$ 维护初始为空的列表 L_{H_2}，记录格式为 $\{\boldsymbol{ID}_i,\boldsymbol{ID}_j,T_i,T_j,pk_i,pk_j,Z_1,Z_2,sk\}$ 的元组。若 L_{H_2} 中有记录则返回对应的 sk 给 $\mathcal{A}_1$；若没有记录，则按如下操作。

若 $\boldsymbol{ID}_i=I$ 或 $\boldsymbol{ID}_i=J$，则按照 1)中 $\boldsymbol{ID}_i=J$ 时进行，其余按照 1)中的规则进行。

RevealPartialPrivateKey($\boldsymbol{ID}_i$)：若 $\boldsymbol{ID}_i=I$ 或 $\boldsymbol{ID}_i=J$，则 $\mathcal{C}$ 停止模拟；否则，$\mathcal{C}$ 发送部分私钥给 $\mathcal{A}_1$。

Send($\prod_{I,J}^s,m$)：$\boldsymbol{ID}_i=I$ 或 $\boldsymbol{ID}_i=J$，则按照 1)中 $\boldsymbol{ID}_i=J$ 时进行，其余按照 1)中的规则进行。

假定 $\mathcal{A}_1$ 赢得此次游戏，则 $\mathcal{A}_1$ 必定计算出了正确的 Z_1、Z_2。$\mathcal{C}$ 则有 $\frac{1}{n_2}$ 的概率在 L_{H_2} 中找到对应的正确元组，则

$$\begin{aligned}Z_2&=t_It_J(k_I+x_I)(k_J+x_J)P\\&=t_It_J(W+x_JV+x_IU+x_Ix_JP)\end{aligned}$$

$$CDH(U,V)=Z_2t_I^{-1}t_J^{-1}-x_JV-x_IU-x_Ix_JP$$

$\mathcal{C}$ 解决 CDH 问题的优势 $Adv_{\mathcal{C}}=\frac{1}{n_0n_1^2n_2}Adv_{\mathcal{A}_1}$。因为 $Adv_{\mathcal{A}_1}$ 不可忽略，则 $Adv_{\mathcal{C}}$ 也不可忽略。这与 CDH 假设矛盾。

4) $\mathcal{A}_1$ 不知道 I 和 J 的临时私钥

除以下查询的回答不同外，其余查询均按照 1)中的规则进行。

Creat($\boldsymbol{ID}_i$)：假设 $\boldsymbol{ID}_i=(I_1,I_2,\cdots,I_{l_i})$，$\mathcal{C}$ 维护初始为空的列表 L_C，记录格式为 $\{\boldsymbol{ID}_i,k_i,R_i^1,\cdots,R_i^{l_i},x_i,P_i\}$ 的元组。$\mathcal{C}$ 按如下规则生成元组数据。

$\mathcal{C}$ 在 L_C 列表中找形如 $\{\boldsymbol{ID}_j,*,*,\cdots,*,*,*\}$ 的元组，其中，$\boldsymbol{ID}_j$ 为 $\boldsymbol{ID}_i$ 的上层用户，取其中层级最大的用户，记为 $\boldsymbol{ID}_j=(I_1,I_2,\cdots,I_{l_j}),l_j\leqslant l_i$。选取 L_{H_1} 列表中相应的 r_i^n,R_i^n，随机选择 $r_i^{l_j},\cdots,r_i^{l_i},k_i,x_i\in\mathbb{Z}_q^*$。计算

$$R_i^n=(l_i-l_j)^{-1}(k_iP-k_jP)(r_i^n)^{-1},n\in\{l_j,\cdots,l_i\}$$

$$P_i=x_iP$$

令 $H_1(I_n\|R_i^n)=r_i^n,n\in\{1,\cdots,l_i\}$，记录 L_C 元组为 $\{\boldsymbol{ID}_i,k_i,R_i^1,\cdots,R_i^{l_i},x_i,P_i\}$，记录 L_{H_1} 元组为 $\{I_n,R_i^n,r_i^n\}$，$n\in\{1,\cdots,l_i\}$。

$H_2(\boldsymbol{ID}_i,\boldsymbol{ID}_j,Z_1,Z_2)$ 查询：$\mathcal{C}$ 维护初始为空的列表 L_{H_2}，记录格式为 $\{\boldsymbol{ID}_i,\boldsymbol{ID}_j,T_i,T_j,pk_i,pk_j,Z_1,Z_2,sk\}$ 的元组。若 L_{H_2} 中有记录则返回对应的 sk 给 $\mathcal{A}_1$；若没有记录，则按如下操作。

$\mathcal{C}$ 在 L_S 列表中找 $\{\boldsymbol{ID}_i,\boldsymbol{ID}_j,T_i,T_j,pk_i,pk_j,sk\}$ 的元组。若找到，则计算

$$Z_1'=t_i(k_i+x_i)(P_0+P_j+\sum_{n=1}^{l_j}(H_1(I_n\|R_j^n)R_j^n))$$

$$\overline{Z_1}=t_i(Z_1-Z_1')$$

$\mathcal{C}$ 检查 T_i、T_j、$\overline{Z_1}$ 是否是 DDH 元组，检查 T_i、T_j、Z_2 是否是 DDH 元组。若是，则 Z_1、Z_2 计算正确，在 L_{H_2} 中记录 $\{\boldsymbol{ID}_i,\boldsymbol{ID}_j,T_i,T_j,pk_i,pk_j,Z_1,Z_2,sk\}$ 的元组，sk 为 L_S 中的值。否则，随机选取 $sk\in\mathcal{K}$，在 L_{H_2} 中记录 $\{\boldsymbol{ID}_i,\boldsymbol{ID}_j,T_i,T_j,pk_i,pk_j,Z_1,Z_2,sk\}$ 的元组。返回 sk 给 $\mathcal{A}_1$。

RevealPartialPrivateKey($\boldsymbol{ID}_i$)：$\mathcal{C}$ 查找 L_C 列表，发送部分私钥给 $\mathcal{A}_1$。

RevealEphemeralKey($\prod_{I,J}^s$)：若 $\prod_{I,J}^s=\prod_{I,J}^T$ 或 $\prod_{I,J}^s=\prod_{J,I}^L$，则 $\mathcal{C}$ 停止模拟；否则，$\mathcal{C}$ 发送临时私钥给 $\mathcal{A}_1$。

Send($\prod_{I,J}^s,m$)：若 $\prod_{I,J}^s=\prod_{I,J}^T$，则返回 $T_i=U$ 给 $\mathcal{A}_1$；若 $\prod_{I,J}^s=\prod_{J,I}^L$，则返回 $T_i=V$ 给 $\mathcal{A}_1$；否则，按照协议规则进行回答。

假定 $\mathcal{A}_1$ 赢得此次游戏，则 $\mathcal{A}_1$ 必定计算出了正确的 Z_1、Z_2。$\mathcal{C}$ 则有 $\frac{1}{n_2}$ 的概率在 L_{H_2} 中找到对应的正确元组。又因为 $T_I=t_I(k_I+x_I)P=U$，$T_J=t_J(k_J+x_J)P=V$，则

$$Z_2=t_It_J(k_I+x_I)(k_J+x_J)P=W$$

所以 $CDH(U,V)=Z_2$。

$\mathcal{C}$ 解决 CDH 问题的优势 $Adv_C=\frac{1}{n_0n_1^2n_2}Adv_{\mathcal{A}_1}$。因为 $Adv_{\mathcal{A}_1}$ 不可忽略，则 Adv_C 也不可忽略。这与 CDH 假设矛盾。

4.1.2 *不存在诚实的用户拥有 Test 会话的匹配会话*

不存在诚实的用户拥有 Test 会话的匹配会话可分为以下 2 种情形。

1) $\mathcal{A}_1$ 拥有 I 的部分私钥和私有秘密值，根据新鲜性定义，敌手不能查询 I 的临时私钥。

2) $\mathcal{A}_1$ 不知道 I 的部分私钥和私有秘密值，根据新鲜性定义，敌手可以查询 I 的临时私钥。

上述 2 种情形分别与 4.1.1 节 1）和 2）类似，同理可证明其安全性。

4.2 引理 2 的证明

引理 2 假定 CDH 假设成立，则本文协议在 $\mathcal{A}_2$ 的攻击下在 eCK 模型下是安全的。

证明 假设 $\mathcal{A}_2$ 以不可忽略的优势 $Adv_{\mathcal{A}_2}$ 在多项式时间内赢得了在 2.4 节中定义的游戏，那么 $\mathcal{C}$ 可利用 $\mathcal{A}_2$ 的能力解决 CDH 问题。同引理 1 中证明类似，此时本文只需考虑 2 种情形。

1) 存在诚实的用户拥有 Test 会话的匹配会话 $\prod_{J,I}^{L}$。

2) 不存在诚实的用户拥有 Test 会话的匹配会话。

4.2.1 *存在诚实的用户拥有 Test 会话的匹配会话 $\prod_{J,I}^{L}$*

$\mathcal{A}_2$ 作为恶意 PKG 可以获得所有参与者的部分私钥 k。此时，存在诚实的用户拥有 Test 会话的匹配会话又可分为以下 4 种情形。

1) $\mathcal{A}_2$ 既不知道 I 的临时私钥，也不知道 J 的私有秘密值

$\mathcal{C}$ 对 $\mathcal{A}_2$ 的查询做出回答，除以下查询的回答不同外，其余查询均按照 4.1.1 节中 1)的规则进行。

Creat($\boldsymbol{ID}_i$)：假设 $\boldsymbol{ID}_i=(I_1,I_2,\cdots,I_{l_i})$，$\mathcal{C}$ 维护初始为空的列表 L_C，记录格式为 $\{\boldsymbol{ID}_i,k_i,R_i^1,\cdots,R_i^{l_i},x_i,P_i\}$ 的元组。$\mathcal{C}$ 按如下规则生成元组数据。

若 $\boldsymbol{ID}_i=J$，$\mathcal{C}$ 在 L_C 列表中找形如 $\{\boldsymbol{ID}_j,*,*,\cdots,*,*,*\}$ 的元组，其中，$\boldsymbol{ID}_j$ 为 $\boldsymbol{ID}_i$ 的上层用户，取其中层级最大的用户，记为 $\boldsymbol{ID}_j=(I_1,I_2,\cdots,I_{l_j})$，$l_j\leqslant l_i$。选取 L_{H_1} 列表中相应的 r_i^n,g_i^n,R_i^n，随机选择 $r_i^{l_j},\cdots,r_i^{l_i},g_i^{l_j},\cdots,g_i^{l_i}\in\mathbb{Z}_q^*$。计算

$$R_i^n=g_i^nP,n\in\{l_j,\cdots,l_i\}$$

$$P_i=U$$

$$k_i=s+\sum_{i=1}^{l_i}(g_i^nr_i^n),n\in\{1,\cdots,l_i\}$$

令 $H_1(I_n\|R_i^n)=r_i^n,n\in\{1,\cdots,l_i\}$，记录 L_C 元组为 $\{ID_i,k_i,R_i^1,\cdots,R_i^{l_i},\perp,P_i\}$，记录 L_{H_1} 元组为 $\{I_n,R_i^n,r_i^n,g_i^n\},n\in\{1,\cdots,l_i\}$。

若 $\boldsymbol{ID}_i\neq J$，$\mathcal{C}$ 在 L_C 列表中找形如 $\{\boldsymbol{ID}_j,*,*,\cdots,*,*,*\}$ 的元组，其中，$\boldsymbol{ID}_j$ 为 $\boldsymbol{ID}_i$ 的上层用户，取其中层级最大的用户，记为 $\boldsymbol{ID}_j=(I_1,I_2,\cdots,I_{l_j})$，$l_j\leqslant l_i$。若 $l_i\neq1$，则选取 L_{H_1} 列表中相应的 r_i^n,g_i^n,R_i^n，随机选择 $r_i^{l_j},\cdots,r_i^{l_i},g_i^{l_j},\cdots,g_i^{l_i},x_i\in\mathbb{Z}_q^*$。若 $l_i=1$，则 $l_j=0$，随机选择 $r_i^1,g_i^1,x_i\in\mathbb{Z}_q^*$。计算

$$R_i^n=g_i^nP,n\in\{l_j,\cdots,l_i\}$$

$$P_i=x_iP$$

$$k_i=s+\sum_{i=1}^{l_i}(g_i^nr_i^n),n\in\{1,\cdots,l_i\}$$

令 $H_1(I_n\|R_i^n)=r_i^n,n\in\{1,\cdots,l_i\}$，记录 L_C 元组为 $\{\boldsymbol{ID}_i,k_i,R_i^1,\cdots,R_i^{l_i},x_i,P_i\}$，记录 L_{H_1} 元组为 $\{I_n,R_i^n,r_i^n,g_i^n\},n\in\{1,\cdots,l_i\}$。

$H_2(\boldsymbol{ID}_i,\boldsymbol{ID}_j,Z_1,Z_2)$ 查询：$\mathcal{C}$ 维护初始为空的列表 L_{H_2}，记录格式为 $\{\boldsymbol{ID}_i,\boldsymbol{ID}_j,T_i,T_j,pk_i,pk_j,Z_1,Z_2,sk\}$ 的元组。若 L_{H_2} 中有记录则返回对应的 sk 给 $\mathcal{A}_2$；若没有记录，则按如下操作。

若 $\boldsymbol{ID}_i=J$，$\mathcal{C}$ 在 L_S 列表中找 $\{\boldsymbol{ID}_i,\boldsymbol{ID}_j,T_i,T_j,pk_i,pk_j,sk\}$ 的元组。若找到，则计算

$$Z_1'=T_j+t_i(P_0+P_j+\sum_{n=1}^{l_j}(H_1(I_n\|R_j^n)R_j^n))$$

$$\overline{Z_1}=Z_1-k_iZ_1'$$

$$\overline{Z_2}=Z_2t_i^{-1}-k_iT_j$$

$\mathcal{C}$ 检查 P_i、Z_1'、$\overline{Z_1}$ 是否是 DDH 元组，检查 P_i、

T_j、$\overline{Z_2}$ 是否是 DDH 元组。若是，则 Z_1、Z_2 计算正确，在 L_{H_2} 中记录 $\{\boldsymbol{ID}_i, \boldsymbol{ID}_j, T_i, T_j, pk_i, pk_j, Z_1, Z_2, sk\}$ 的元组，sk 为 L_S 中的值。否则，随机选取 $sk \in \mathcal{K}$，在 L_{H_2} 中记录 $\{\boldsymbol{ID}_i, \boldsymbol{ID}_j, T_i, T_j, pk_i, pk_j, Z_1, Z_2, sk\}$ 的元组。返回 sk 给 $\mathcal{A}_2$。

若 $\boldsymbol{ID}_i \neq J$，按 4.1.1 节中 1)的规则回答。

RevealSecretValue($\boldsymbol{ID}_i$)：$\mathcal{C}$ 按以下规则回答。

若 $\boldsymbol{ID}_i = J$，则 $\mathcal{C}$ 停止模拟；否则，按 4.1.1 节中 1)的规则回答。

RevealPartialPrivateKey($\boldsymbol{ID}_i$)：$\mathcal{C}$ 发送部分私钥给 $\mathcal{A}_2$。

RevealMasterKey：$\mathcal{C}$ 返回主私钥给 $\mathcal{A}_2$。

Send($\prod_{I,J}^s, m$)：若 $\prod_{I,J}^s = \prod_{I,J}^T$，则返回 $T_i = V$ 给 $\mathcal{A}_2$。

若 $ID_i = J$，$\mathcal{C}$ 随机选取 $t_i \in \mathbb{Z}_q^*$，并计算

$$Z_1' = T_j + t_i(P_0 + P_j + \sum_{n=1}^{l_j}(H_1(I_n \| R_j^n)R_j^n))$$

$$\overline{Z_1} = Z_1 - k_i Z_1'$$

$$\overline{Z_2} = Z_2 t_i^{-1} - k_i T_j$$

$\mathcal{C}$ 检查 P_i、Z_1'、$\overline{Z_1}$ 是否是 DDH 元组，检查 P_i、T_j、$\overline{Z_2}$ 是否是 DDH 元组。若是，则 Z_1、Z_2 计算正确，在 L_S 中记录 $\{\boldsymbol{ID}_i, \boldsymbol{ID}_j, T_i, T_j, pk_i, pk_j, sk\}$ 的元组，sk 为 L_{H_2} 中的值。否则，随机选取 $sk \in \mathcal{K}$，在 L_S 中记录 $\{\boldsymbol{ID}_i, \boldsymbol{ID}_j, T_i, T_j, pk_i, pk_j, sk\}$ 的元组。

否则，按照协议规则进行回答。

假定 $\mathcal{A}_2$ 赢得此次游戏，则 $\mathcal{A}_2$ 必定计算出了正确的 Z_1、Z_2。$\mathcal{C}$ 则有 $\frac{1}{n_2}$ 的概率在 L_{H_2} 中找到对应的正确元组。又因为 $T_I = t_I(k_I + x_I)P = V$，则

$$\begin{aligned} Z_2 &= t_I t_J (k_I + x_I)(k_J + x_J)P \\ &= t_J k_J V + t_I t_J (k_I + x_I)U \\ &= t_J k_J V + t_J W \end{aligned}$$

$$CDH(U,V) = (Z_2 - t_J k_J V)t_J^{-1}$$

$\mathcal{C}$ 解决 CDH 问题的优势 $Adv_{\mathcal{C}} = \frac{1}{n_0 n_1^2 n_2} Adv_{\mathcal{A}_2}$。因为 $Adv_{\mathcal{A}_2}$ 不可忽略，则 $Adv_{\mathcal{C}}$ 也不可忽略。这与 CDH 假设矛盾。

剩余情形的证明与引理1类似,同理可得引理2成立。

由引理 1 和引理 2 可得出定理 1。

定理 1　本文基于 CDH 假设的协议在 eCK 模型下是安全的。

5　效率分析

5.1　协议复杂度

本节对本文协议主要算法的计算复杂度进行分析，并与文献[13,14]协议相对比，结果如表 1 所示。由于3种协议在构造上存在差别，所以分别以 Key-Extract、Delegate 和 Agreement 代表 3 种协议的私钥生成、私钥委托和密钥协商算法。在分析计算复杂度时，主要考虑的计算类型有：椭圆曲线上的双线性对运算、循环加法群 $\mathbb{G}$ 中点乘运算及双线性群 $\mathbb{G}_T$ 中的指数运算及散列函数运算。用 P 表示双线性对运算，M 表示点乘运算，E 表示 $\mathbb{G}_T$ 中的指数运算，H 表示散列函数 $H: \{0,1\}^* \to \mathbb{G}$ 的运算。对于有系统最大层级限制的方案，用 L 表示系统最大层级，l 表示用户所处层级，l_A、l_B 为用户 A 和 B 的层级，i 表示为用户 A 和 B 在第 i 层有相同的节点。

表 1　计算复杂度分析

算法	文献[13]	文献[14]	本文协议
Key-Extract	$(L+l+3)$M	$2l$M$+l$H	$(l+1)$M
Delegate	$(L+l+4)$M	2M + H	2M
Agreement	$3(l_A+l_B-2i+2)$M + $6P$	2M + 10P + 2E + 6H	(l_A+l_B+8)M

5.2　算法执行耗时

为了对比协议的计算开销，本节选取 jpbc 库中 A 类椭圆曲线进行测试，其曲线方程为 $y^2 = x^3 + x$，$\mathbb{G}$ 群阶数为 512 bit，$\mathbb{Z}_q^*$ 域阶数为 160 bit。测试平台的相关配置信息如表 2 所示。

本节对 A 类曲线中的双线性对运算、$\mathbb{G}$ 群点乘运算、$\mathbb{G}$ 群乘法运算、$\mathbb{G}_T$ 群指数运算、$\mathbb{G}_T$ 群乘法运算、$\mathbb{Z}_q^*$ 域乘法运算、$\mathbb{Z}_q^*$ 域加法运算及散列函数运算的单次运算耗时分别进行了测试，结果如表3所示。

表 2　测试平台配置

项目	参数
CPU 型号	Intel Core i3-E7400
CPU 主频	2.80 GHz×2
内存	4 GB
系统类型	Win7–32 bit
软件平台	Eclipse + JDK 1.7.0
jpbc 库版本	2.0.0

表 3 单次运算执行耗时

测试项目	耗时/ms
双线性对运算	38.47
$\mathbb{G}$ 群点乘运算	32.37
$\mathbb{G}$ 群乘法运算	0.17
$\mathbb{G}_T$ 群指数运算	4.41
$\mathbb{G}_T$ 群乘法运算	0.04
$\mathbb{Z}_q^*$ 域乘法运算	0.01
$\mathbb{Z}_q^*$ 域加法运算	0.01
$H:\{0,1\}^*\to\mathbb{Z}_q^*$	0.12
$H:\{0,1\}^*\to\mathbb{G}$	76.73

假设系统层级为 7，表 4 和表 5 分别为 Key-Extract 和 Delegate 算法的执行耗时。在 Agreement 算法中，本节选取 3 种情形进行测试，假设用户所处层级分别为 $l_A=2$、$l_B=7$、$i=1$，$l_A=4$、$l_B=6$、$i=3$，$l_A=6$、$l_B=6$、$i=5$，$l_A=6$、$l_B=6$、$i=2$。测试结果如表 6 所示。

表 4 Key-Extract 算法执行耗时

层级	文献[13]耗时/ms	文献[14]耗时/ms	本文协议/ms
1	354.61	178.92	65.36
2	390.65	310.27	98.83
3	424.34	447.46	129.12
4	458.43	583.37	162.65
5	483.13	719.19	193.56
6	519.64	855.51	224.02
7	547.37	994.18	258.13

表 5 Delegate 算法执行耗时

层级	文献[13]耗时/ms	文献[14]耗时/ms	本文协议/ms
1	411.45	139.57	65.04
2	449.78	137.97	65.15
3	481.16	136.56	64.95
4	513.49	138.03	64.82
5	542.05	138.23	65.05
6	577.36	136.39	65.08

表 6 Agreement 算法执行耗时

层级	文献[13]耗时/ms	文献[14]耗时/ms	本文协议/ms
$l_A=2,l_B=7,i=1$	992.39	874.25	545.12
$l_A=4,l_B=6,i=3$	797.84	872.83	580.33
$l_A=6,l_B=6,i=5$	661.27	872.46	649.37
$l_A=6,l_B=6,i=2$	1055.32	872.46	649.37

测试结果表明，与同类协议相比，本文协议不仅解决了身份基密码体制固有的密钥托管问题，而且具有较高的执行效率。在私钥生成和私钥委托算法方面，本文协议具有不可比拟的优势。在密钥协商阶段，算法耗时与用户所处层级之和呈线性关系。虽然算法耗时随着用户所处层级之和的增加而不断增加，但与前2种协议相比，本文协议仍具有较高的效率。

6 结束语

本文提出一种适用于空间信息网的无证书的层次认证密钥协商协议，并对协议的安全性进行了证明。本文协议不仅解决了传统公钥基础设施中的证书管理问题，而且避免了基于身份密码体制固有的密钥托管问题。与同类协议相比，本文协议具有较高的执行效率。本文协议对系统层级深度没有限制，扩大了无证书的层次认证密钥协商协议的适用范围。

参考文献：

[1] HUNT R. PKI and digital certification infrastructure[C]// Ninth IEEE International Conference on Networks. IEEE, c2001: 234-239.

[2] SHAMIR A. Identity based cryptosystems and signature schemes[C]// Advances in Cryptology Crypto84. Berlin: Springer-Verlag, c1984: 47-53.

[3] AL-RIYAMI S S, PATERSON K G. Certificateless public key cryptography[M]. Advances in Cryptology-ASIACRYPT 2003, Springer Berlin Heidelberg, 2003: 452-473.

[4] HE D, CHEN Y, CHEN J, et al. A new two-round certificateless authenticated key agreement protocol without bilinear pairings [J]. Mathematical and Computer Modelling, 2011, 54(11): 3143-3152.

[5] HE D, CHEN J, HU J. A pairing - free certificateless authenticated key agreement protocol[J]. International Journal of Communication Systems, 2012, 25(2): 221-230.

[6] HE D, PADHYE S, CHEN J. An efficient certificateless two-party authenticated key agreement protocol[J]. Computers & Mathematics with Applications, 2012, 64(6): 1914-1926.

[7] TONG D, LIU J W, MAO K F, et al. Certificateless and pairing-free key agreement scheme for satellite network[C]//Communications Security Conference (CSC 2014). IET, c2014: 1-5.

[8] MOHAMED N A F, HASHIM M H A, BASHIER E, et al. Fully-secure and efficient pairing-free certificateless authenticated key agreement protocol[C]// 2012 World Congress on Internet Security (WorldCIS), IEEE, c2012: 167-172.

[9] SUN H Y, WEN Q Y, ZHANG H, et al. A strongly secure pairing-free certificateless authenticated key agreement protocol for low-power de-

vices[J]. Information Technology and Control, 2013, 42(2): 191-204.

[10] GHOREISHI S M, ABD R S, ISNIN I F, et al. New secure identity-based and certificateless authenticated key agreement protocols without pairings[C]// 2014 International Symposium on Biometrics and Security Technologies (ISBAST). IEEE, c2014: 188-192

[11] WANG Z, DU X, SUN Y. Group key management scheme based on proxy re-cryptography for near-space network[C]// 2011 International Conference on Network Computing and Information Security (NCIS). IEEE. c2011, 1: 52-56.

[12] GENTRY C, SILVERBERG A. Hierarchical ID-based cryptography[M]. Advances in cryptology—ASIACRYPT2002, Springer Berlin Heidelberg, 2002: 548-566

[13] 曹晨磊, 刘明奇, 张茹, 等. 基于层级化身份的可证明安全的认证密钥协商协议[J]. 电子与信息学报, 2014, 36(12): 2848-2854.
CAO C L, LIU M Q, ZHANG R, et al. Provably secure authenticated key agreement protocol based on hierarchical identity[J]. Journal of Electronics & Information Technology, 2014, 36(12):2848-2854.

[14] LIU W, LIU J, WU Q, et al. SAKE: scalable authenticated key exchange for mobile e‐health networks[J/OL]. Security and Communication Networks, http://onlinelibrary.wiley.com/doi/10.1002/sec.1198/epdf.

[15] CHOW S S M, ROTH V, RIEFFEL E G. General certificateless encryption and timed-release encryption [M]. Security and Cryptography for Networks, Springer Berlin Heidelberg, 2008: 126-143.

[16] HANKERSON D, VANSTONE S, MENEZES A J. Guide to elliptic curve cryptography [M]. Springer Science & Business Media, 2004.

[17] LIPPOLD G, BOYD C, NIETO J G. Strongly secure certificateless key agreement[M]. Pairing-Based Cryptography–Pairing 2009, Springer Berlin Heidelberg, 2009: 206-230.

作者简介：

苏航（1992-），男，安徽宿州人，北京航空航天大学硕士生，主要研究方向为空间网络安全、密码学。

刘建伟（1964-），男，山东莱州人，博士，北京航空航天大学教授、博士生导师，主要研究方向为密码学、网络与信息安全。

陶芮（1991-），女，天津人，北京航空航天大学硕士生，主要研究方向为密码学、网络与信息安全。