标准模型下基于身份的动态门限代理签名方案

于义科[1,2] 郑雪峰[1] 刘行兵[1] 韩晓光[1]
(北京科技大学信息工程学院 北京 100083)[1] (南昌航空大学信息工程学院 南昌 330063)[2]

摘 要 目前对于基于身份的门限代理签名方案来说,方案的安全性大多是在随机预言模型下证明的,并且这些门限代理签名方案中的代理签名密钥固定。相对而言,设计标准模型下的动态门限代理签名方案更有实际意义。针对以上两个问题,根据Paterson提出的基于身份的签名方案,利用计算Diffie-Hellman问题的困难性在标准模型下设计了一个代理签名密钥可变的动态门限代理签名方案。最后,利用双线性对技术对方案的正确性进行了严格证明,并在CDH困难假设下给出了方案安全性的详细分析过程和证明,结果表明方案是可证安全的和可靠的。

关键词 动态门限代理签名,标准模型,基于身份的密码,计算Diffie-Hellman问题
中图法分类号 TP393 **文献标识码** A

ID-based Proactive Threshold Proxy Signature in the Standard Model

YU Yi-ke[1,2] ZHENG Xue-feng[1] LIU Xing-bing[1] HAN Xiao-guang[1]
(School of Information Engineering, University of Science and Technology Beijing, Beijing 100083, China)[1]
(School of Information Engineering, Nanchang Hangkong University, Nanchang 330063, China)[2]

Abstract At present, the security of the identity based proactive threshold proxy signature schemes was almost proven in the random oracle model, and the threshold value of these schemes was almost changeless. It is more practical to design threshold proxy signature scheme in the standard model, compared with the existing ones. Aiming at the two problems pointed out above, this paper presented a proactive threshold proxy signature scheme by using the hardness of the computational Diffie-Hellman problem with changeable threshold value, based on the modification of Paterson' proposed identity based signature scheme. At last, the scheme's correctness was exactly proven in terms of bilinear pairing technique and its security analysis and prove were given in detail in the assumption of the computational Diffie-Hellman problem, therefore, this scheme was proven secure and reliable.

Keywords Proactive threshold proxy signature, Standard model, Identity based cryptography, Computational Diffie-Hellman problem

1996年Mambo等[1]首先提出了代理签名的概念。在代理签名方案中,原始签名人可将其签名权委托给代理签名人,由代理签名人代表原始签名人生成有效的代理签名。它在电子商务及电子政务中有着广泛的应用,例如电子投票、股票交易、电子支票或电子货币的分发等都涉及代理签名问题。目前保护代理的具有证书的部分代理签名是代理签名的研究热点,一般地,将这种代理签名简称为代理签名。

为了避免代理权过于集中,Zhang和Kim等[2,3]最早提出了门限代理签名方案。门限代理签名是建立在秘密共享[4]上的代理签名方法。(t,n)门限代理签名就是原始签名人授权n个代理签名人,n个代理签名人分别拥有各自的子代理签名密钥。各代理签名人利用自己的子代理签名密钥生成部分代理签名。当部分代理签名的个数大于或等于t时,将这部分代理签名按着某种方式结合,就可以产生有效的代理签名,而任何少于t个的代理人都无法产生有效的代理签名。

自门限代理签名的概念提出以来,人们对其进行了广泛的研究,提出了许多门限代理签名方案[5-8]。这些方案使用的是基于证书(Certificate)的公钥体系,用户的公钥分配是使用证书来实现的,存在着复杂的证书管理,用户必须依赖于CA(Certificate Authority)的第三方服务。为了避开复杂的证书管理,1984年Shamir[9]提出了基于身份的密码学思想。在基于身份的密码系统中,根据其身份计算得到用户的公钥,而用户的私钥则由可信第三方(PKG)产生。2001年,Boneh和Franklin[10]提出了一个实用的基于身份的加密方案。后来人们利用双线性映射构造了大量的基于身份的密码方案。

Xu等人[11]首次提出了基于身份的门限代理签名(IBTPS)方案,并在随机预言也模型下证明了它的安全性。近几年人们相继提出了一些基于身份的门限代理签名方案[12,13]。目前,这些方案尚存在以下问题:(1)方案普遍是在随机预言模型下可证安全的。然而,随机预言模型把Hash函数作为一个完全随机的理想模型,是一个很强的要求,在具体应用中无法构造相应的实例;(2)代理签名密钥都是不变

到稿日期:2010-04-10 返修日期:2010-09-19 本文受国家自然科学基金重点项目(60932003),国家自然科学基金项目(60803123,60674054),北京市重点学科建设项目(XK100080537),北京市重点学科建设项目-计算机系统结构资助。

于义科(1970—),男,博士,副教授,主要研究方向为密码学与信息安全,E-mail:yyk312@163.com;郑雪峰(1951—),男,教授,博士生导师,主要研究方向为网络安全;刘行兵(1973—),男,博士生,主要研究方向为无线网络、信息集成;韩晓光(1982—),男,博士生,主要研究方向为云计算。

的。门限代理签名的安全性很大程度上都取决于签名密钥的安全性，将代理签名密钥定期更新，能进一步提高方案的安全性。因此，设计标准模型下可证安全的动态门限代理签名方案更有实际意义。本文首先定义了基于身份的动态门限代理签名(IBPTPS)的安全模型，然后根据 Paterson[14] 提出的基于身份的签名方案提出了一个在标准模型下可证安全的基于身份的动态门限代理签名方案，最后在所定义的安全模型下没有使用随机预言也证明了方案是安全的，并且它的安全性是基于更一般的计算 Diffie-Hellman 假设的困难性，因此方案具有更高的安全性。

1 预备知识

1.1 双线性映射

设 G 和 G_1 是阶为素数 p 的两个循环群，g 是群 G 的生成元，如果映射 $e:G\times G\to G_1$ 具有如下性质：

(1)双线性：对所有的 $u,v\in G$ 和 $a,b\in_R \mathbb{Z}_p^*$，都有 $e(u^a,v^b)=e(u,v)^{ab}$；

(2)非退化性：$e(g,g)\neq 1$；

(3)可计算性：存在一个有效的算法计算 $e(u,v)$，其中 $u,v\in G$；

则称该映射为双线性映射。利用有限域中椭圆曲线上的 Weil 对或 Tate 对可以构造这样的双线性映射。

1.2 复杂性假设

已知 G 是阶为素数 p 的循环群，g 是群 G 的生成元，$a,b\in_R \mathbb{Z}_p^*$，已知 $g,g^a,g^b\in G$，计算 g^{ab}。这就是在 G 上的计算 Diffie-Hellman(CDH)问题。

如果不存在运行时间至多为 t、解决群 G 上 CDH 问题的概率至少为 ε 的算法，则称 (t,ε)-CDH 假设在群 G 上成立。

2 IBPTPS 体制定义

2.1 IBPTPS 的形式化定义

定义 1 一个基于身份的门限代理签名是由普通的基于身份的签名扩展而成的，设基于身份的标准签名方案为 $IBS=\{\mathcal{G},\mathcal{E},\mathcal{S},\mathcal{V}\}$，则基于身份的动态门限代理(IBPTPS)签名方案为：$IBPTPS=\{\mathcal{G},\mathcal{E},\mathcal{S},\mathcal{V},\mathcal{TH},\mathcal{TU},(\mathcal{TD},\mathcal{TP}),\mathcal{TPS},\mathcal{TPV},\mathcal{PS},\mathcal{PV}\}$。

(1)$\mathcal{G}$ 是系统参数产生算法，用于生成系统参数。

(2)$\mathcal{E}$ 是用户私钥产生算法，用于为参与签名的用户生成密钥对。

(3)$\mathcal{S}$ 是标准签名产生算法，用于产生用户的标准签名。

(4)$\mathcal{V}$ 是标准签名验证算法，用于验证用户的标准签名。

(5)$\mathcal{TH}$ 是代理签名人秘密分享算法，用于为代理签名组成员生成秘密分享值。

(6)$\mathcal{TU}$ 是代理签名人秘密分享更新算法，用于为代理签名组成员更新秘密分享值。

(7)$(\mathcal{TD},\mathcal{TP})$ 是代理指定协议，$\mathcal{TD}$ 把原始签名人的签名权委托给代理签名组的各个成员，$\mathcal{TP}$ 产生代理签名人的子代理签名私钥。

(8)$\mathcal{TPS}$ 是部分代理签名产生算法，用于产生部分代理签名。

(9)$\mathcal{TPV}$ 是部分代理签名验证算法，用于验证部分代理签名的合法性。

(10)$\mathcal{PS}$ 是代理签名产生算法，用于产生代理签名 p_σ。

(11)$\mathcal{PV}$ 是代理签名验证算法，用于验证代理签名的合法性。

2.2 IBPTPS 的安全要求

基于身份的门限代理签名方案应该满足下面的安全要求：

(1)可区分性(Distinguishability)。任何人都可以区别代理签名和标准签名。

(2)可验证性(Verifiability)。根据代理签名，验证人能确信原始签名人认可代理签名人所签的签名。

(3)强可识别性(Strong identifiability)。根据代理签名，任何人都可确定相应的代理签名人的身份。

(4)强不可伪造性(Strong unforgeability)。代理签名人能代表原始签名人产生有效的部分代理签名，而原始签名人和其他没有指定为代理人的第三方都不能产生有效的部分代理签名；只有超过 t 个代理签名人的合法部分代理签名才能产生有效的代理签名，任何少于 t 个的代理签名人或原始签名人都不能产生有效的代理签名。

(5)强不可否认性(Strong undeniability)。如果代理签名人代表原始签名人产生了有效的部分代理签名，他就不能否认他产生的部分代理签名；如果门限代理签名组代表原始签名人产生了有效的代理签名，他们就不能否认他们产生的代理签名。

3 本文的 IBPTPS 方案

3.1 方案描述

设 P_A 是原始签名人，身份为 ID_A。$PS=\{P_1,P_2,\cdots,P_n\}$ 是一个代理人组成的代理签名群，相应身份为 $ID_i(i=1,\cdots,n)$。t 为门限值，用户的身份 ID、授权证书 w 和消息 m 分别是长度为 n_u,n_w 和 n_m 的比特串。

(1)系统参数产生算法 $\mathcal{G}$

PKG 选择阶为素数 p 的循环群 G 和 G_1，生成元 $g\in G$，随机数 $a\in\mathbb{Z}_p,g_2,u',m',w'\in G;U=(u_1,u_2,\cdots,u_{n_u})\in G^{n_u}$，$W=(w_1,w_2,\cdots,w_{n_w})\in G^{n_w}$，$M=(m_1,m_2,\cdots,m_{n_m})\in G^{n_m}$。然后构造双线性映射 $e:G\times G\to G_1$，计算 $g_1=g^a$。公开参数为 $params=(G,G_1,e,g,g_1,g_2,u',U,w',W,m',M)$，主密钥为 $msk=g_2^a$。

(2)用户私钥产生算法 $\mathcal{E}$

已知用户身份 ID，令 $\mathcal{U}\subseteq\{1,2,\cdots,n_u\}$ 为 $ID[i]=1$ 的序号 i 的集合。PKG 随机选择 $r_{ID}\in\mathbb{Z}_p$，并计算

$$K_{ID}=(g_2^a(u'\prod_{i\in\mathcal{U}}u_i)^{r_{ID}},g^{r_{ID}})=(K_{ID,1},K_{ID,2})$$

则 K_{ID} 是身份为 ID 的标准签名私钥，将 K_{ID} 秘密发送给用户 ID。

(3)标准签名产生算法 $\mathcal{S}$

设 m 为要签名的消息，令 $\mathcal{M}\subseteq\{1,2,\cdots,n_m\}$ 为 $m[l]=1$ 的序号 l 的集合，用户 ID 任意选择 $r_m\in\mathbb{Z}_p$，并计算

$$\begin{aligned}\sigma&=(K_{ID,1}(m''\prod_{l\in\mathcal{M}}m_l)^{r_m},K_{ID,2},g^{r_m})\\&=(g_2^a(u'\prod_{i\in\mathcal{U}_{ID}}u_i)^{r_{ID}}(m'\prod_{l\in\mathcal{M}}m_l)^{r_m},g^{r_{ID}},g^{r_m})\\&=(\sigma_1,\sigma_2,\sigma_3)\end{aligned}$$

(4)标准签名验证算法 $\mathcal{V}$

通过下面等式验证签名 $\sigma=(\sigma_1,\sigma_2,\sigma_3)$ 的有效性：

$$e(\sigma_1,g)=e(g_1,g_2)e(u'\prod_{i\in\mathcal{U}_{ID}}u_i,\sigma_2)e(m'\prod_{l\in\mathcal{M}}m_l,\sigma_3)$$

(5)代理签名人秘密分享算法 $\mathcal{SH}$

每个代理成员 P_i 随机选取$\langle s_i \in \mathbb{Z}_p, s_i' \in \mathbb{Z}_p \rangle$为其子秘密对，公开 $r_i = g^{s_i}$ 和 $r_i' = g^{s_i'}$，并计算和广播 $R = \prod_{i=1}^{n} r_i, R' = \prod_{i=1}^{n} r_i'$。随机选取系数为$\mathbb{Z}_p$、次数为 $t-1$ 的两个多项式 $f_i(z)$和 $h_i(z)$：

$$f_i(z) = a_{i,0} + a_{i,1}z + \cdots + a_{i,t-1}z^{t-1}$$

$$h_i(z) = b_{i,0} + b_{i,1}z + \cdots + b_{i,t-1}z^{t-1}$$

令 $s_i = a_{i,0}, s_i' = b_{i,0}$ P_i 广播 $C_{i,d} = g^{a_{i,d}}$ $(d = 0, 1, \cdots, t-1), C_{i,d}' = g^{b_{i,d}}$ $(d = 0, 1, \cdots, t-1)$，计算秘密分享 $s_{i,j} = f_i(j)$和 $s_{i,j}' = h_i(j)$，然后把它们秘密发送给其它成员 P_j $(j = 1, 2, \cdots, n; j \neq i)$。

成员 P_j 从 P_i 那里收到分享 $s_{i,j}$ 和 $s_{i,j}'$ 后，用下面的等式验证其有效性：

$$g^{s_{i,j}} = \prod_{d=0}^{t-1} (C_{i,d})^{j^d} \bmod p, g^{s'_{i,d}} = \prod_{d=0}^{t-1} (C_{i,d}')^{j^d} \bmod p$$

如果没有通过验证，则 P_j 产生对 P_i 的控告。若多于 $t-1$ 个成员产生对 P_i 的控告，则认为 P_i 是不诚实的。若检查通过，P_i 计算秘密分享 $x_i = \sum_{k=1}^{n} f_k(i)$和 $y_i = \sum_{k=1}^{n} h_k(i)$，并公开 $X_i = g^{x_i}$ 和 $Y_i = g^{y_i}$。其余的代理签名人也可以通过计算 $X_i = \prod_{j=1}^{n}\prod_{k=0}^{t-1} C_{j,k}^{i^k}$和 $Y_i = \prod_{j=1}^{n}\prod_{k=0}^{t-1} C_{j,k}'^{i^k}$来分别得到 X_i 和 Y_i。若令 $f(z) = \sum_{i=1}^{n} f_i(z), h(z) = \sum_{i=1}^{n} h_i(z)$，则 $x_i = f(i), y_i = h(i)$。

(6)代理签名人秘密分享更新算法 $\mathcal{SU}$

每个代理成员 P_i 随机选取系数为$\mathbb{Z}_p$、次数为 $t-1$ 的多项式 $R_i(x)$：

$$R_i(x) = r_{i,1}x + \cdots + r_{i,t-1}x^{t-1}$$

P_i广播 $K_{i,d} = g^{r_{i,d}}$ $(d = 1, 2, \cdots, t-1)$，计算分享 $Q_{i,j} = R_i(j)$，然后秘密发送给其他的成员 P_j。

成员 P_j 从其他成员 P_i 那里收到分享 $Q_{i,j}$ 后，用下面的等式验证其有效性：

$$g^{Q_{i,j}} = \prod_{d=1}^{t-1} (K_{i,d})^{j^d} \bmod p$$

如果没有通过验证，则 P_j 产生对 P_i 的控告。若多于 $t-1$ 个成员产生对 P_i 的控告，则认为它是不诚实的。

每个成员 P_j 更新其秘密分享 $x_j' = x_j + \sum_{i=1}^{n} Q_{i,j}$ 和 $y_j' = y_j + \sum_{i=1}^{n} Q_{i,j}$，并公开 $X_j = g^{x_j'}$ 和 $Y_j = g^{y_j'}$。

(7)代理指定协议($\mathcal{PD}$,$\mathcal{PP}$)

$\mathcal{PD}$：已知授权文件 w，包含了代理关系的描述，如原始签名人的身份信息、代理签名组规模和成员的身份信息、门限值 t，代理签名消息的限制(如类型、期限等)。令 $\mathcal{W} \subseteq \{1, 2, \cdots, n_w\}$为 $w[j] = 1$ 的序号 j 的集合。原始签名人 P_A 的标准签名私钥为$(K_{P_A,1}, K_{P_A,2})$，P_A 随机选择 $r_w \in \mathbb{Z}_p$，计算

$$\begin{aligned}\sigma_w &= (K_{P_A,1}(w'\prod_{j\in\mathcal{W}} w_j)^{r_w}, K_{P_A,2}, g^{r_w}) \\ &= (g_2^{a}(u'\prod_{i\in\mathcal{U}_{P_A}} u_i)^{r_{P_A}}(w'\prod_{j\in\mathcal{W}} w_j)^{r_w}, g^{r_{P_A}}, g^{r_w}) \\ &= (\sigma_{w1}, \sigma_{w2}, \sigma_{w3})\end{aligned}$$

原始签名人 P_A 发送(w, σ_w)给每个代理成员 P_k $(k = 1, 2, \cdots, n)$。

$\mathcal{PP}$：代理签名人 P_k 通过验证下面等式判断(w, σ_w)是否有效：

$$e(\sigma_{w1}, g) = e(g_1, g_2)e(u'\prod_{i\in\mathcal{U}_A} u_i, \sigma_{w2})e(w'\prod_{j\in\mathcal{W}} w_j, \sigma_{w3})$$

设每个代理签名人 P_k $(k = 1, 2, \cdots, n)$标准签名私钥为$(K_{P_k,1}, K_{P_k,2})$，若(w, σ_w)有效，P_k 计算 $\omega_k = \prod_{j=1, j\neq k}^{t} j/(j-k)$，

$$\begin{aligned}PK_k &= (\sigma_{w1} \cdot K_{P_k,1} \cdot (w'\prod_{j\in\mathcal{W}} w_j)^{x_k\omega_k}, \sigma_{w2}, K_{P_k,2}, g^{x_k\omega_k} \cdot \sigma_{w3}) \\ &= (g_2^{a}(u'\prod_{i\in\mathcal{U}_{P_A}} u_i)^{r_{P_A}}(w'\prod_{j\in\mathcal{W}} w_j)^{r_w} \cdot g_2^{a}(u'\prod_{i\in\mathcal{U}_{P_k}} u_i)^{r_{P_k}} \\ &\quad \cdot (w'\prod_{j\in\mathcal{W}} w_j)^{x_k\omega_k}, g^{r_{P_A}}, g^{r_{P_k}}, g^{r_w + x_k\omega_k}) \\ &= (PK_{k,1}, PK_{k,2}, PK_{k,3}, PK_{k,4})\end{aligned}$$

则 PK_k 为代理签名人 P_k 代表原始签名人 P_A 签名的子代理签名私钥。

(8)部分代理签名产生算法 $\mathcal{PPS}$

设 m 为要签名的消息，P_k $(k = 1, 2, \cdots, t)$代表原始签名人对消息 m 进行代理签名。令 $\mathcal{M} \subseteq \{1, 2, \cdots, n_m\}$为 $m[l] = 1$ 的序号 l 的集合，则 P_k 计算

$$\begin{aligned}p\sigma_k &= (PK_{k,1}(m'\prod_{l\in\mathcal{M}} m_l)^{y_k\omega_k}, PK_{k,2}, PK_{k,3}, PK_{k,4}, g^{y_k\omega_k}) \\ &= (g_2^{2a}(u'\prod_{i\in\mathcal{U}_{P_A}} u_i)^{r_{P_A}}(u'\prod_{i\in\mathcal{U}_{P_k}} u_i)^{r_{P_k}}(w'\prod_{j\in\mathcal{W}} w_j)^{r_w + x_k\omega_k} \\ &\quad (m'\prod_{l\in\mathcal{M}} m_l)^{y_k\omega_k}, g^{r_{P_A}}, g^{r_{P_k}}, g^{r_w}, g^{x_k\omega_k}, g^{y_k\omega_k}) \\ &= (V_k, R_{k,1}, R_{k,2}, R_{k,3}, R_{k,4}, R_{k,5})\end{aligned}$$

则 $p\sigma_k$ 为 P_k 产生的部分代理签名。

(9)部分代理签名验证算法 $\mathcal{PPV}$

部分代理签名 $p\sigma_k$ 是代理签名人 P_k 代表原始签名人 P_A 在消息 m 上的有效部分代理签名吗？如果下面的等式成立，则是，否则，不是。

$$\begin{aligned}e(V_k, g) &= e^2(g_1, g_2) \cdot e(u'\prod_{i\in U_A} u_i, R_{k,1}) \cdot e(u'\prod_{i\in\mathcal{U}_{P_k}} u_i, R_{k,2}) \cdot e(w'\prod_{j\in W} w_j, R_{k,3} \cdot R_{k,4}) \cdot e(m'\prod_{l\in\mathcal{M}} m_l, R_{k,5}) \\ &= e^2(g_1, g_2) \cdot e(u'\prod_{i\in U_A} u_i, R_{k,1}) \cdot e(u'\prod_{i\in\mathcal{U}_{P_k}} u_i, R_{k,2}) \cdot e(w'\prod_{j\in W} w_j, R_{k,3} \cdot (X_k)^{\omega_k}) e(m'\prod_{l\in\mathcal{M}} m_l, (Y_k)^{\omega_k})\end{aligned}$$

(10)代理签名产生算法 $\mathcal{PS}$

若部分代理签名 $p\sigma_k$ $(k = 1, 2, \cdots, t)$通过验证，则 t 个代理签名人中任何一人计算

$$\begin{aligned}p\sigma' &= (\prod_{k=1}^{t} V_k, \prod_{k=1}^{t} R_{k,1}, R_{1,2}, \cdots, R_{t,2}, \prod_{k=1}^{t} R_{k,3}, \prod_{k=1}^{t} R_{k,4}, \prod_{k=1}^{t} R_{k,5}) \\ &= (PV, PR_1, PR_{2,1}, PR_{2,2}, \cdots, PR_{2,t}, PR_3, PR_4, PR_5)\end{aligned}$$

最后输出代理签名 $p\sigma = (ID_A, ID_1, \cdots, ID_t, w, p\sigma')$。

(11)代理签名验证算法 $\mathcal{PV}$

代理签名 $p\sigma = (ID_A, ID_1, \cdots, ID_t, w, p\sigma')$是代理签名组$(P_1, \cdots, P_n)$代表原始签名人 P_A 在消息 m 上的有效门限代理签名吗？如果下面的等式都成立，则是，否则不是。

$$\begin{aligned}e(PV, g) &= e^{2t}(g_1, g_2) \cdot e(u'\prod_{i\in\mathcal{U}_A} u_i, PR_1) \cdot \prod_{k=1}^{t} e(u'\prod_{i\in\mathcal{U}_{P_k}} u_i, PR_{2,k}) \cdot e(w'\prod_{j\in\mathcal{W}} w_j, PR_3 \cdot PR_4) e(m'\prod_{l\in\mathcal{M}} m_l, PR_5) \\ &= e^{2t}(g_1, g_2) \cdot e(u'\prod_{i\in\mathcal{U}_A} u_i, PR_1) \cdot \prod_{k=1}^{t} e(u'\prod_{i\in\mathcal{U}_{P_k}} u_i, PR_{2,k}) e(w'\prod_{j\in\mathcal{W}} w_j, PR_3 \cdot R) e(m'\prod_{l\in\mathcal{M}} m_l, R')\end{aligned}$$

3.2 方案正确性

(1)门限秘密共享生成算法的正确性

$$\sum_{i=1}^{t} x_i\omega_i = f(0) = \sum_{i=1}^{n} f_i(0) = \sum_{i=1}^{n} s_i$$

(2)门限秘密共享更新算法的正确性

$$\sum_{j=1}^{t} x_j'\omega_j = \sum_{j=1}^{t}(x_j + \sum_{i=1}^{n} Q_{i,j})\omega_j = \sum_{j=1}^{t} x_j\omega_j + \sum_{j=1}^{t}\sum_{i=1}^{n} R_i(j)\omega_j = \sum_{j=1}^{t} x_j\omega_j + \sum_{i=1}^{n}\sum_{j=1}^{t} R_i(j)\omega_j = \sum_{j=1}^{n} s_j + \sum_{i=1}^{n} R_i(0) = \sum_{j=1}^{n} s_j$$

式中,ω_i,ω_j 为 Lagrange 系数。

(3)代理签名验证的正确性

$$\begin{aligned}
e(PV,g) &= e(g_2^{2a}(u' \prod_{i\in\mathcal{U}_{P_A}} u_i)^{r_{P_A}} \prod_{k=1}^{t}(u' \prod_{i\in\mathcal{U}_{P_k}} u_i)^{r_{P_k}} (w' \prod_{j\in\mathcal{W}} w_j)^{r_w + \sum_{k=1}^{t} x_k\omega_k} (m' \prod_{l\in\mathcal{M}} m_l)^{\sum_{k=1}^{t} y_k\omega_k}, g) \\
&= e(g_2^{2a}, g)e((u' \prod_{i\in\mathcal{U}_{P_A}} u_i)^{r_{P_A}}, g) \cdot e(\prod_{k=1}^{t}(u' \prod_{i\in\mathcal{U}_{P_k}} u_i)^{r_{P_k}}, g)\, e((w' \prod_{j\in\mathcal{W}} w_j)^{r_w}, g)\, e((w' \prod_{j\in\mathcal{W}} w_j)^{\sum_{k=1}^{t} x_k\omega_k}, g)e((m' \prod_{l\in\mathcal{M}} m_l)^{\sum_{k=1}^{t} y_k\omega_k}, g) \\
&= e^{2t}(g_1, g_2)e(u' \prod_{i\in\mathcal{U}_{P_A}} u_i, g^{r_{P_A}}) \prod_{k=1}^{t} e(u' \prod_{i\in\mathcal{U}_{P_k}} u_i, g^{r_{P_k}}) \cdot e(w' \prod_{j\in\mathcal{W}} w_j, g^{r_w})e(w' \prod_{j\in\mathcal{W}} w_j, g_k^{\sum_{k=1}^{t} x_k\omega_k})e(m' \prod_{l\in\mathcal{M}} m_l, g_k^{\sum_{k=1}^{t} y_k\omega_k}) \\
&= e^{2t}(g_1, g_2)e(u' \prod_{i\in\mathcal{U}_{P_A}} u_i, PR_1) \prod_{k=1}^{t} e(u' \prod_{i\in\mathcal{U}_{P_k}} u_i, PR_{2,k}) \cdot e(w' \prod_{j\in\mathcal{W}} w_j, PR_3 \cdot R)e(m' \prod_{l\in\mathcal{M}} m_l, R')
\end{aligned}$$

因此,方案是正确的。

3.3 方案安全性

(1)可区分性

在一个有效的代理签名中包含一个授权文件,而且在代理签名验证等式中授权文件、原始签名人和代理签名人必须同时出现,所以任何人都可以区别代理签名和标准签名。

(2)可验证性

在一个有效的代理签名中包含一个授权文件、原始签名人、参与签名的代理签名人,而且在代理签名验证等式中授权文件、原始签名人、参与签名的代理签名人必须同时出现,所以根据代理签名,验证人能确信原始签名人认可代理签名人所签的签名。

(3)强可识别性

在一个有效的代理签名中包含参与签名的代理签名人的身份,所以根据代理签名,任何人都可确定相应的代理签名人的身份。

(4)强不可否认性(Strong undeniability)

因为在一个有效的部分代理签名中包含授权文件、原始签名人、代理签名人,而且在部分代理签名验证等式中授权文件、原始签名人、代理签名人必须同时出现,所以根据部分代理签名,代理签名人不能否认他代表原始签名人所做的签名。同理,代理签名组也不能否认他们代表原始签名人所签的签名。

(5)强不可伪造性(Strong unforgeability)

定义 2 如果不存在运行时间至多为 t、优势至少为 ε 的敌手 A,并且用户私钥询问的次数最多为 q_e,标准签名询问的次数为 q_s,部分代理签名私钥询问的次数最多为 q_{pk},代理签名询问的次数最多为 q_{ps},则该方案是 $(t, q_e, q_s, q_{pk}, q_{ps}, \varepsilon)$-EU-IBPTPS-CMIA 安全的。

定理 1 在 CDH 困难问题的假设下,方案是 EU-IB-PTPS-CMIA 安全的。

证明:假设敌手 $\mathcal{A}$ 能以不可忽略的优势攻击上面的方案,则能构造算法 $\mathcal{B}$,$\mathcal{B}$ 可以利用 $\mathcal{A}$ 解决 CDH 问题,从而导致矛盾。

给定 $\mathcal{B}$ 一个 CDH 问题的实例 (g, g^a, g^b)。为了计算 g^{ab},$\mathcal{B}$ 模仿 $\mathcal{A}$ 的挑战者 $\mathcal{C}$,具体过程如下。

系统建立:$\mathcal{B}$ 设定 $l_u = 2(q_e + q_s + q_{pk} + q_{ps})$,$l_w = 2(q_{pk} + q_{ps})$,$l_m = 2(q_s + q_{ps})$,其中 q_e 是 $\mathcal{A}$ 私钥询问的次数,q_s 是 $\mathcal{A}$ 标准签名询问的次数,q_{pk} 是门限代理私钥询问的次数,q_{ps} 是代理签名询问的次数。选择整数 k_u,k_w 和 k_m,满足 $0 \leqslant k_u \leqslant n_u$,$0 \leqslant k_w \leqslant n_w$ 和 $0 \leqslant k_m \leqslant n_m$,并假定 $l_u(n_u+1) < p$,$l_w(n_w+1) < p$ 和 $l_m(n_m+1) < p$。$\mathcal{B}$ 选择 $x' \in_R \mathbb{Z}_{l_u}$ 及长度为 n_u 的向量 $X = (x_i)$,其中 $x_i \in_R \mathbb{Z}_{l_u}$;选择 $y' \in_R \mathbb{Z}_{l_w}$ 及长度为 n_w 的向量 $Y = (y_i)$,其中 $y_i \in_R \mathbb{Z}_{l_w}$;选择 $z' \in_R \mathbb{Z}_{bn}$ 及长度为 n_m 的向量 $Z = (z_j)$,其中。最后 $\mathcal{B}$ 选择 $a', b', c' \in_R \mathbb{Z}_p$,并选择长度为 n_u 的向量 $A = (a_i)$、长度为 n_w 的向量 $B = (b_j)$ 和长度为 n_m 的向量 $C = (c_k)$,其中 $a_i, b_j, c_k \in_R \mathbb{Z}_p$。

对于身份 ID、授权文件 W 和消息 M,定义以下几个函数:

$$F(ID) = x' + \sum_{i\in\mathcal{U}} x_i - l_u k_u, J(ID) = a' + \sum_{i\in\mathcal{U}} a_i$$
$$K(W) = y_j' + \sum_{j\in\mathcal{W}} y_j - l_w k_w, L(W) = b' + \sum_{j\in\mathcal{W}} b_j$$
$$Q(M) = z' + \sum_{k\in\mathcal{M}} z_k - l_m k_m, Q(M) = c' + \sum_{k\in\mathcal{M}} c_k$$

算法 $\mathcal{B}$ 构造上面方案中的公开参数如下:

$g_1 = g^a$,$g_2 = g^b$

$u' = g_2^{-l_u k_u + x'} g^{a'}$,$u_i = g_2^{x_i} g^{a_i}$,$1 \leqslant i \leqslant n_u$

$w' = g_2^{-l_w k_w + y'} g^{b'}$,$w_j = g_2^{y_j} g^{b_j}$,$1 \leqslant j \leqslant n_w$

$m' = g_2^{-l_m k_m + z'} g^{c'}$,$m_k = g_2^{z_k} g^{c_k}$,$1 \leqslant k \leqslant n_m$

可以看出,这些参数的分布与一个真正的挑战者产生的公开参数的分布是一样的。这样可以得到主密钥为 $g_2^a = g^{ab}$,同时下面的等式也是成立的:

$$u' \prod_{i\in\mathcal{U}} u_i = g_2^{F(ID)} g^{J(ID)}, w' \prod_{j\in\mathcal{W}} w_j = g_2^{K(W)} g^{L(W)},$$
$$m' \prod_{k\in\mathcal{M}} m_k = g_2^{Q(M)} g^{R(M)}$$

算法 $\mathcal{B}$ 将公开参数发送给敌手 $\mathcal{A}$。

询问:当敌手 $\mathcal{A}$ 发起私钥询问和签名询问时,算法 $\mathcal{B}$ 进行如下响应:

①用户私钥询问:对身份为 ID 的用户私钥询问。虽然 $\mathcal{B}$ 不知道主密钥,但是在假定 $F(ID) \neq 0 \bmod p$ 的情况下,$\mathcal{B}$ 也能够构造其私钥 d_{ID}。不失一般性,假设 $F(ID) \neq 0 \bmod p$,$\mathcal{B}$ 选取 $r_u \in \mathbb{Z}_p$ 并计算:

$$d_{ID} = (g_1^{-J(ID)/F(ID)} (u' \prod_{i\in\mathcal{U}} u_i)^{r_u}, g_1^{-1/F(ID)} g^{r_u}) = (d_{ID,1}, d_{ID,2})$$

令 $\tilde{r}_u = r_u - a/F(ID)$,可以验证 d_{ID} 是 ID 的有效密钥。

$$\begin{aligned}
d_{ID,1} &= g_1^{-J(ID)/F(ID)} (u' \prod_{i\in\mathcal{U}} u_i)^{r_u} = g_1^{-J(ID)/F(ID)} (g_2^{F(ID)} g^{J(ID)}) \\
&= g_2^a (g_2^{F(ID)} g^{J(ID)})^{r_u - a/F(ID)} = g_2^a (u' \prod_{i\in\mathcal{U}} u_i)^{\tilde{r}_u}
\end{aligned}$$

$$d_{ID,2} = g_1^{-1/F(ID)} g^{r_u} = g^{r_u - a/F(ID)} = g^{\tilde{r}_u}$$

如果 $F(ID) = 0 \bmod p$,上面的计算将无法进行,$\mathcal{B}$ 将失败退出。

②标准签名询问：考虑在身份 ID 下对消息 M 的标准签名询问。不失一般性，假设 $\mathcal{A}$ 没有询问身份 ID 的密钥。如果 $F(ID)\neq 0 \bmod l_u$，算法 $\mathcal{B}$ 首先构造 ID 的私钥，然后利用 $\mathcal{S}$ 算法产生 ID 在消息 M 上的标准签名；否则，如果 $Q(M)\neq 0 \bmod l_m$，算法 $\mathcal{B}$ 也可以像在私钥询问中构造私钥的方法那样构造一个标准签名，$\mathcal{B}$ 随机选择 $r_u, r_m \in Z_p$ 构造身份 ID 在消息 M 上的标准签名：

$$\begin{aligned}\sigma &= (g_1^{-R(M)/Q(M)}(u'\prod_{i\in\mathcal{U}}u_i)^{r_u}(m'\prod_{k\in\mathcal{M}}m_k)^{r_m}, g^{r_u}, g_1^{-1/Q(M)}g^{r_m})\\ &=(g_2^a(u'\prod_{i\in\mathcal{U}}u_i)^{r_u}(m'\prod_{k\in\mathcal{M}}m_k)^{\tilde{r}_m}, g^{r_u}, g^{\tilde{r}_m})\end{aligned}$$

式中，$\tilde{r}_m = r_m - a/Q(M)$；否则如果 $Q(M)=0 \bmod l_m$，则 $\mathcal{B}$ 将失败退出。

③子代理签名私钥询问Ⅰ：$\mathcal{A}$ 提交原始签名人身份 ID_A 和代理签名人身份 $ID_i, ID_i \in \{ID_1,\cdots,ID_n\}$，$\mathcal{A}$ 扮演原始签名人角色 P_A，首先运行算法 $\mathcal{PD}$ 产生授权证书

$$\sigma_w=(g_2^a(u'\prod_{i\in\mathcal{U}_A}u_i)^{r_a}(w'\prod_{j\in\mathcal{W}}w_j)^{r_w}, g^{r_a}, g^{r_w})$$

并将(W,σ_W)发送给 $\mathcal{B}$。$\mathcal{B}$ 验证 σ_W 的有效性，如果 σ_W 有效，则 $\mathcal{B}$ 以下面的方法构造子代理签名私钥。首先运行 $\mathcal{PH}$ 协议生成所有代理人的秘密分享(x_i, y_i)；然后判断如果 $F(ID_i)\neq 0 \bmod l_u$，则 $\mathcal{B}$ 先构造 ID_i 的标准签名私钥，然后运行 $\mathcal{PP}$ 算法产生子代理签名私钥；否则，在假设 $l_w(n_w+1)<p$ 下蕴含 $K(W)\neq 0 \bmod p$，则 $\mathcal{B}$ 随机选择 $r_i \in Z_p$，计算子代理签名私钥

$$\begin{aligned}PK_i &= (g_2^a(u'\prod_{i\in\mathcal{U}_A}u_i)^{r_a}(u'\prod_{i\in\mathcal{U}_i}u_i)^{r_i}g_1^{-L(W)/K(W)}(w'\prod_{j\in\mathcal{W}}w_j)^{r_w+x_i\omega_i}, g^{r_a}, g^{r_i}, g_1^{-1/K(W)}g^{r_w+x_i\omega_i})\\ &=(g_2^{2a}(u'\prod_{i\in\mathcal{U}_A}u_i)^{r_a}(u'\prod_{i\in\mathcal{U}_i}u_i)^{r_i}(w'\prod_{j\in\mathcal{W}}w_j)^{r_w+r_w''}, g^{r_a}, g^{r_i}, g^{r_w+r_w''})\end{aligned}$$

式中，$r_w''=x_i\omega_i - a/K(W)$；否则 $\mathcal{B}$ 失败退出。

④子代理签名私钥询问Ⅱ：$\mathcal{A}$ 提交原始签名人身份 ID_A 和代理签名人身份 $ID_i, ID_i \in \{ID_1,\cdots,ID_n\}$，$\mathcal{A}$ 扮演代理签名人 ID_i，如果 $F(ID_A)\neq 0 \bmod l_u$，则 $\mathcal{B}$ 先构造 ID_A 的私钥，然后运行算法 $\mathcal{PD}$ 产生授权证书 σ_W；否则，如果 $K(W)\neq 0 \bmod l_w$，则 $\mathcal{B}$ 随机选择 $r_a, r_w \in Z_p$，计算

$$\begin{aligned}\sigma_W &= (g_1^{-L(W)/K(W)}(u'\prod_{i\in\mathcal{U}_A}u_i)^{r_a}(w'\prod_{j\in\mathcal{W}}w_j)^{r_w}, g^{r_a}, g_1^{-1/K(W)}g^{r_w})\\ &=(g_2^a(u'\prod_{i\in\mathcal{U}_A}u_i)^{r_a}(w'\prod_{j\in\mathcal{W}}w_j)^{r_w'}, g^{r_a}, g^{r'_w})\end{aligned}$$

式中，$r_w'=r_w - a/K(W)$；否则 $\mathcal{B}$ 失败退出。

⑤部分代理签名询问：$\mathcal{A}$ 给出原始签名人身份 ID_A 和代理签名人身份 $ID_i, ID_i \in \{ID_1,\cdots,ID_n\}$ 和消息 M，$\mathcal{A}$ 扮演原始签名人角色 P_A。$\mathcal{A}$ 提交部分代理签名询问$\langle ID_A, ID_i, W, M\rangle$，询问 ID_i 代表 ID_A 在 M 上的部分代理签名。$\mathcal{A}$ 运行算法 $\mathcal{PD}$ 产生授权证书

$$\sigma_w=(g_2^a(u'\prod_{i\in\mathcal{U}_A}u_i)^{r_a}(w'\prod_{j\in\mathcal{W}}w_j)^{r_w}, g^{r_a}, g^{r_w})$$

并将(W,σ_W)发送给 $\mathcal{B}$。$\mathcal{B}$ 验证 σ_W 的有效性。如果 σ_W 有效，则 $\mathcal{B}$ 以下面的方法构造部分代理签名。首先运行 $\mathcal{PH}$ 协议生成所有代理人的秘密分享(x_i, y_i)；如果 $F(ID)\neq 0 \bmod l_u \vee K(W)\neq 0 \bmod l_w$，则 $\mathcal{B}$ 先构造 ID_i 代表 ID_A 在 M 上的子代理签名私钥，然后运行 $\mathcal{PP}$ 算法产生部分代理签名；否则，如果 $Q(M)\neq 0 \bmod l_m$，则 $\mathcal{B}$ 随机选择 $r_i \in Z_p$，并计算

$$\begin{aligned}p\sigma_i &= (g_2^a(u'\prod_{i\in\mathcal{U}_A}u_i)^{r_a}(u'\prod_{i\in\mathcal{U}_i}u_i)^{r_i}(w'\prod_{j\in\mathcal{W}}w_j)^{r_w+x_i\omega_i}\\ &\quad g_1^{-R(W)/Q(W)}(m'\prod_{k\in\mathcal{M}}m_k)^{y_i\omega_i}, g^{r_a}, g^{r_i}, g^{r_w}, g^{x_i\omega_i},\\ &\quad g_1^{-1/Q(W)}g^{y_i\omega})\\ &=(g_2^{2a}(u'\prod_{i\in\mathcal{U}_A}u_i)^{r_a}(u'\prod_{i\in\mathcal{U}_i}u_i)^{r_i}(w'\prod_{j\in\mathcal{W}}w_j)^{r_w+x_i\omega_i}(m'\\ &\quad \prod_{k\in\mathcal{M}}m_k)^{r_m}, g^{r_a}, g^{r_i}, g^{r_w}, g^{x_i\omega_i}, g^{r_m})\end{aligned}$$

式中，$r_m'=y_i\omega_i - a/Q(W)$；否则 $\mathcal{B}$ 失败退出。

⑥代理签名询问：$\mathcal{A}$ 给出原始签名人身份 ID_A 和代理签名组$\{ID_1,\cdots,ID_n\}$和消息 M，$\mathcal{A}$ 扮演原始签名人角色 P_A。$\mathcal{A}$ 提交代理签名询问$\langle ID_A, (ID_1,\cdots,ID_n), W, M\rangle$，询问$\{ID_1,\cdots,ID_n\}$代表 ID_A 在 M 上的代理签名。$\mathcal{B}$ 用部分代理签名的方法产生各个代理人的部分签名，然后运行 $\mathcal{PS}$ 算法产生代理签名。$\mathcal{B}$ 失败退出的情况和部分代理签名询问一样。

伪造：如果 $\mathcal{B}$ 能够回答 $\mathcal{A}$ 所有的询问，即 $\mathcal{B}$ 没有失败退出，且 $\mathcal{A}$ 能以不可忽略的概率 ϵ 输出一个有效的签名伪造，则这个伪造可以是下面 4 种情形之一：

①在身份 ID^* 和消息 M^* 下的有效伪造标准签名 $\sigma^*=(\sigma_1^*, \sigma_2^*, \sigma_3^*)$。$\mathcal{A}$ 没有提交 ID^* 私钥询问和(ID^*, M^*)标准签名询问。如果 $F(ID^*)=0 \bmod p$ 且 $Q(M^*)=0 \bmod p$，则 $\mathcal{B}$ 计算并输出

$$\begin{aligned}\frac{\sigma_1^*}{(\sigma_2^*)^{J(ID^*)}(\sigma_3^*)^{R(M^*)}} &= \frac{g_2^a(u'\prod_{i\in\mathcal{U}}u_i)^{r_u}(m'\prod_{k\in\mathcal{M}}m_k)^{r_m}}{g^{J(ID^*)r_u}g^{R(M^*)r_m}}\\ &= \frac{g_2^a(g_2^{F(ID^*)}g^{J(ID^*)})^{r_u}(g_2^{Q(M^*)}g^{R(M^*)})^{r_m}}{g^{J(ID^*)r_u}g^{R(M^*)r_m}}\\ &= g_2^a = g^{ab}\end{aligned}$$

上述输出就是 CDH 问题实例的解；否则 $\mathcal{B}$ 失败退出。

②在身份 ID_A^*，ID_k^*，授权证书 W^* 和消息 M^* 下的有效伪造部分代理签名 $p\sigma_k^*$。这是敌手 $\mathcal{A}$ 在已知原始签名人 ID_A^* 的私钥而不知道代理签名人 ID_k^* 的私钥时对部分代理签名的伪造。如果 $F(ID_k^*)\neq 0 \bmod p$ 或 $K(W^*)\neq 0 \bmod p$ 或 $Q(M^*)\neq 0 \bmod p$，则算法 $\mathcal{B}$ 失败退出。如果 $F(ID_k^*)=0 \bmod p, K(W^*)=0 \bmod p$ 且 $Q(M^*)=0 \bmod p$，则 $\mathcal{B}$ 计算

$$\begin{aligned}&\frac{V_k^*}{d_{A,1}^*(R_{k,2}^*)^{J(ID_k^*)}(R_{k,3}^*R_{k,4}^*)^{L(W^*)}(R_{k,5}^*)^{R(M^*)}}\\ &= \frac{g_2^{2a}(u'\prod_{i\in\mathcal{U}_A}u_i)^{r_a}(u'\prod_{i\in\mathcal{U}_{ID_k}}u_i)^{r_k}(w'\prod_{j\in\mathcal{W}}w_j)^{r_w+r_w'}(m'\prod_{l\in\mathcal{M}}m_l)^{r_m}}{g_2^a(u'\prod_{i\in\mathcal{U}_A}u_i)^{r_a}g^{J(ID_k^*)r_k}g^{L(W^*)(r_w+r'_w)}g^{R(M^*)r_m}}\\ &= g_2^a = g^{ab}\end{aligned}$$

上述输出就是 CDH 问题实例的解。

③在身份 ID_A^*，ID_k^*，授权证书 W^* 和消息 M^* 下的有效伪造部分代理签名 $p\sigma_k^*$。这是敌手 $\mathcal{A}$ 在已知代理签名人 ID_k^* 的私钥而不知道原始签名人 ID_A^* 的私钥时对部分代理签名的伪造。如果 $F(ID_A^*)\neq 0 \bmod p$ 或 $K(W^*)\neq 0 \bmod p$ 或 $Q(M^*)\neq 0 \bmod p$，则算法 $\mathcal{B}$ 失败退出。

如果 $F(ID_A^*)=0 \bmod p$ 且 $K(W^*)=0 \bmod p$ 且 $Q(M^*)=0 \bmod p$，则 $\mathcal{B}$ 计算并输出

$$\begin{aligned}&\frac{V_k^*}{(R_{k,1}^*)^{J(ID_A^*)}d_{k,1}^*(R_{k,3}^*R_{k,4}^*)^{L(W^*)}(R_{k,5}^*)^{R(M^*)}}\\ &= \frac{g_2^{2a}(u'\prod_{i\in\mathcal{U}_A}u_i)^{r_a}(u'\prod_{i\in\mathcal{U}_{ID_k}}u_i)^{r_k}(w'\prod_{j\in\mathcal{W}}w_j)^{r_w+r_w'}(m'\prod_{l\in\mathcal{M}}m_l)^{r_m}}{g^{J(ID_A^*)r_a}g_2^a(u'\prod_{i\in\mathcal{W}_{ID_k}}u_i)^{r_k}g^{L(W^*)(r_w+r_w')}g^{R(M^*)r_m}}\\ &= g_2^a = g^{ab}\end{aligned}$$

上述输出就是 CDH 问题实例的解。

④在身份 ID_A^*、代理组(ID_1^*,…,ID_n^*)、授权证书 W^* 和消息 M^* 下的有效伪造代理签名 $p\sigma^*$。不失一般性,假设敌手只询问了 $t-1$ 个代理人的部分代理签名 $p\sigma_i^*$ ($1\leqslant i\leqslant t-1$),根据代理签名产生的方法可知其必须伪造第 t 代理人的部分代理签名 $p\sigma_t^*$,那么按上面分析 $\mathcal{B}$ 可能计算出 CDH 问题实例的解。

因此,如果存在一个敌手 $\mathcal{A}$ 可以不可忽略的概率伪造一个有效的签名,那么就存在一个算法 $\mathcal{B}$ 能以不可忽略的概率解决 CDH 问题,而这与 CDH 问题是一个困难问题相矛盾,故方案是 EU-IBPTPS-CMIA 安全的。

上面描述了算法 $\mathcal{B}$ 的模拟过程。下面分析一下算法解决 CDH 问题实例的概率以及它的运行时间。

概率分析:如果满足下面几个条件,则算法 $\mathcal{B}$ 将不会以失败退出而结束:

①所有的用户私钥询问都满足 $F(ID)\neq 0 \bmod l_u$;

②所有的标准签名询问都满足 $F(ID)\neq 0 \bmod l_u$ 或 $Q(M)\neq 0 \bmod l_m$;

③所有的门限代理私钥询问Ⅰ都满足 $F(ID_i)\neq 0 \bmod l_u$ 或 $K(W)\neq 0 \bmod l_w$;

④所有的门限代理私钥询问Ⅱ都满足 $F(ID_A)\neq 0 \bmod l_u$ 或 $K(W)\neq 0 \bmod l_w$;

⑤所有的部分代理签名询问都满足 $F(ID_i)\neq 0 \bmod l_u$ 或 $K(W)\neq 0 \bmod l_w$ 或 $Q(M)\neq 0 \bmod l_m$;

⑥在伪造阶段满足 $F(ID_i^*)=0 \bmod p$ 和 $K(W^*)=0 \bmod p$ 和 $Q(M^*)=0 \bmod p$。

为了简化对模拟者的分析,我们只考虑在这个事件上的一个子集的概率。例如把标准签名询问(ID,M)分为包括 $ID=ID_k^*$ 和包括 $ID\neq ID_k^*$ 的询问,对于包括 $ID\neq ID_k^*$ 的询问只考虑 $F(ID)\neq 0 \bmod l_u$ 的情况,而忽略 $F(ID)=0 \bmod l_u$ 且 $Q(M)\neq 0 \bmod l_m$ 的情况。

设在用户私钥询问以及不包括身份 ID_k^* 的标准签名询问、子代理签名私钥询问和代理签名询问中的身份为 ID_1,…,ID_{q_I},在包括身份 ID_k^* 而不包括授权文件 W^* 的子代理签名私钥询问和代理签名询问中的授权文件为 W_1,…,W_{q_W},在包括身份 ID_k^* 的标准签名询问以及包括身份 ID_k^* 和授权文件 W^* 的代理签名询问中消息为 M_1,…,M_{q_M}。显然有 $q_I\leqslant q_e+q_s+q_{pk}+q_{ps}$,$q_W\leqslant q_{pk}+q_{ps}$,$q_M\leqslant q_s+q_{ps}$。又定义事件 A_i,A^*,B_j,B^*,C_l,C^* 如下:

$A_i: F(ID_i)\neq 0 \bmod l_u$,$A^*: F(ID_k^*)\neq 0 \bmod p$,

$B_j: K(W_j)\neq 0 \bmod l_w$,$B^*: K(W^*)\neq 0 \bmod p$,

$C_l: Q(M_l)\neq 0 \bmod l_m$,$C^*: Q(M^*)\neq 0 \bmod p$。

根据上面的分析,算法没有失败退出的概率为

$$\Pr[\neg abort]\geqslant \Pr[\bigcap_{i=1}^{q_I} A_i \wedge A^* \wedge \bigcap_{j=1}^{q_W} B_j \wedge B^* \bigcap_{l=1}^{q_M} C_l \wedge C^*]$$

可以看出,事件 $\bigcap_{i=1}^{q_I} A_i \wedge A^*$,$\bigcap_{j=1}^{q_W} B_j \wedge B^*$ 和 $\bigcap_{l=1}^{q_M} C_l \wedge C^*$ 是相互独立的。

由假设 $l_u(n_u+1)<p$ 可知,$F(ID_k^*)=0 \bmod p$ 蕴涵 $F(ID_k^*)=0 \bmod l_u$,可得

$$\begin{aligned}\Pr[A^*] &= \Pr[F(ID_k^*)=0 \bmod p \wedge F(ID_k^*)=0 \bmod l_u]\\ &= \Pr[F(ID_k^*)=0 \bmod l_u]\Pr[F(ID_k^*)=0 \bmod p \mid F(ID_k^*)=0 \bmod l_u]\\ &= \frac{1}{l_u(n_u+1)}\end{aligned}$$

另外对任意 i,事件 A_i 和 A^* 相互独立,且 $\Pr[\neg A_i \mid A^*]=\frac{1}{l_u}$,则

$$\Pr[\bigcap_{i=1}^{q_I} A_i \mid A^*]=1-\Pr[\bigcup_{i=1}^{q_I} \neg A_i \mid A^*]\geqslant 1-\sum_{i=1}^{q_I}\Pr[\neg A_i \mid A^*]=1-\frac{q_I}{l_u}$$

所以可以得到

$$\begin{aligned}\Pr[\bigcap_{i=1}^{q_I} A_i \wedge A^*] &= \Pr[A^*]Pr[\bigcap_{i=1}^{q_I} A_i \mid A^*]\\ &\geqslant \frac{1}{l_u(n_u+1)}(1-\frac{q_I}{l_u})\\ &\geqslant \frac{1}{l_u(n_u+1)}(1-\frac{q_e+q_s+q_{pk}+q_{ps}}{l_u})\end{aligned}$$

由设定 $l_u=2(q_e+q_s+q_{pk}+q_{ps})$,可得

$$\Pr[\bigcap_{i=1}^{q_I} A_i \wedge A^*]\geqslant \frac{1}{4(q_e+q_s+q_{pk}+q_{ps})(n_u+1)}$$

同理可得

$$\Pr[\bigcap_{j=1}^{q_W} B_j \wedge B^*]\geqslant \frac{1}{4(q_{pk}+q_{ps})(n_w+1)}$$

$$\Pr[\bigcap_{l=1}^{q_M} C_l \wedge C^*]\geqslant \frac{1}{4(q_s+q_{ps})(n_m+1)}$$

因此,我们有

$$\Pr[\neg abort]\geqslant \Pr[\bigcap_{i=1}^{q_I} A_i \wedge A^* \wedge \bigcap_{j=1}^{q_W} B_j \wedge B^* \bigcap_{l=1}^{q_M} C_l \wedge C^*]\geqslant \frac{1}{4(q_e+q_s+q_{pk}+q_{ps})(n_u+1)}\cdot\frac{1}{4(q_{pk}+q_{ps})(n_w+1)}\cdot\frac{1}{4(q_s+q_{ps})(n_m+1)}=\frac{1}{64(q_e+q_s+q_{pk}+q_{ps})(q_{pk}+q_{ps})(q_s+q_{ps})(n_u+1)(n_w+1)(n_m+1)}$$

时间复杂度分析:算法的时间复杂度是由用户私钥询问、标准签名询问、门限代理私钥询问和代理签名询问中的乘法运算和指数运算次数决定的。由于在用户私钥询问、标准签名询问、子代理签名私钥询问和代理签名询问中的乘法运算次数分别是 $O(n_u)$,$O(n_u+n_w)$,$O(n_u+n_w)$,$O(n_u+n_w+n_m)$,指数运算次数均为 $O(1)$,因此算法 $\mathcal{B}$ 的时间复杂度为

$$t+O((q_e n_u+q_s(n_u+n_m)+q_{pk}(n_u+n_w)+q_{ps}(n_u+n_w+n_m))\rho+(q_e+q_s+q_{pk}+q_{ps})\tau)$$

3.4 方案效率

本方案的公开参数 $params$ 主要由 $n_u+n_w+n_m+6$ 个群 G 的元素组成,用户普通签名私钥 K_{ID} 为 2 个群 G 的元素,门限代理签名私钥为 4 个群 G 的元素,部分代理签名 $p\sigma_k$ 为 6 个群 G 的元素,代理签名 $p\sigma$ 为 $t+5$ 个群 G 的元素。生成部分代理签名时,代理签名者进行最多 n_m+1 次群 G 的乘法运算和 2 次群 G 的指数运算;代理签名进行 $5t$ 次群 G 的乘法运算。在部分代理签名验证时,验证者进行最多 $2n_u+n_m+n_w+1$次群 G 的乘法运算、5 次双线性对运算和 6 次群 G_1 的乘法运算;在门限代理签名验证时,验证者进行$(t+1)n_u+n_m+n_w+1$次群 G 的乘法运算、$t+4$ 次双线性对运算和 $t+3$ 次群 G_1 的乘法运算和 1 次群 G_1 的指数运算。

结束语 门限代理签名是一种重要的具有特殊性质的签名形式,现有的基于身份的门限代理签名方案的安全性都是在随机模型下证明的。本文在标准模型下设计了一个基于身

份的(t,n)动态门限代理签名方案。相对于一般的基于身份的门限代理签名方案，本文方案具有更高的动态属性。同时，利用CDH问题的困难性证明了本文方案在标准模型下的安全性，因此相对于随机预言模型下可证安全的方案来说，本文方案具有更高的安全性。

参考文献

[1] Mambo M, Usuda K, Okamoto E. Proxy signature for delegating signing operation [C]// Proceedings of the 3rd ACM Conference on Computer and Communications Security. New York: ACM, 1996: 48-57

[2] Zhang K. Threshold proxy signature schemes[C]// Proceedings of Information Security Workshop (ISW97). LNCS 1396, Springer-Verlag, 1997: 282-290

[3] Kim S, Park S, Won D. Proxy Signatures, Revisited [C]// Proceedingsof Information and Communications Security (ICICS 97). LNCS 1334. Springer-Verlag, 1997: 223-232

[4] Shamir A. How to share a secret[J]. Communications of the ACM, 1979, 22(11): 612-613

[5] Sun H M. An efficient nonrepudiable threshold proxy signature scheme with known signers[J]. Computer Communicaion, 1997, 22(8): 717-722

[6] 李继国，曹珍富. 一个改进的门限代理签名方案[J]. 计算机研究与发展，2002，39(11)：1513-1518

[7] Hwang M S, Lu J L, Lin L C. A practical(t,n) threshold proxy signature scheme based on the RSA cryptosystem[J]. IEEE Trans. on Knowledge and Data Engineering, 2003, 15(6): 1552-1560

[8] 蒋瀚，徐秋亮，周永彬. 基于RSA密码体制的门限代理签名[J]. 计算机学报，2007，30(2)：241-247

[9] Shamir A. Identity-based cryptosystems and signature schemes [C]// Blakley G, Chaum D, eds. Proceedings of Crypto 1984. New York: Springer-Verlag, 1984: 47-53

[10] Boneh D, Franklin M. Identity-based encryption from the Weil pairing[C]// Kilian J, ed. Proceedings of Crypto 2001. London: Springer-Verlag, 2001: 213-229

[11] Xu J, Zhang Z F, Feng D G. Identity Based Threshold Proxy Signature[EB/OL]. http://eprint.iacr.org/2004/250/

[12] Bao H Y, Cao Z F, Wang S B. Identity-based Threshold Proxy Signature Scheme with Known Signers [C]// Proceedings of Theory and Applications of Models of Computation. LNCS 3959. Springer-Verlag, 2006: 538-546

[13] 鲁荣波，何大可，王常吉. 对一种基于身份的已知签名人的门限代理签名方案的分析[J]. 电子与信息学报，2008，30(1)：100-103

[14] Paterson K G, Schuldt J C N. Efficient identity-based signatures secure in the standard model[C]// Proceedings of ACISP 2006. Berlin: Springer-Verlag, 2006: 207-222

(上接第9页)

[37] Becker C, Buijssen S H M, Wobker H, et al. FEAST: Development of HPC technologies for FEM applications[C]// Münster G, Wolf D, Kremer M, eds. High Performance Computing in Science and Engineering. Berlin: Springer, 2008

[38] Goddeke D, Strzodka R, Mohd-Yusof J, et al. Exploring weak scalability for FEM calculations on a GPU-enhanced cluster[J]. Parallel Computing, 2007(33): 685-699

[39] Pastor L, Orero J L B. An Efficiency and Scalability Model for Heterogeneous Clusters[C]// Proceedings of the 2001 IEEE International Conference on Cluster Computing. Newport Beach: IEEE, 2001

[40] Zhe F, Feng Q, Kaufman A, et al. GPU cluster for high performance computing[C]// SC2004. Washington: IEEE, 2004

[41] Ogawa S, Aoki T. GPU computing for 2-dimensional incompressible-flow simulation based on multigrid method [C]// Transactions of the Japan Society for Computational Engineering and Science. 2009: 20090021

[42] Nukada A, Matsuoka S. Auto-tuning 3-D FFT Library for CUDA GPUs[C]// SC2009. Portland: ACM, 2009

[43] Matsuoka S. Petascaling Commodity onto Exascale: GPUs as Multithreaded Massively-parallel Vector Processors-the Only Road to Exascale [C]// IEEE Cluster Computing Conference 2009. New Orleans: IEEE, 2009

[44] Matsuoka S, Aoki T, Endo T, et al. GPU accelerated computing-from hype to mainstream, the rebirth of vector computing[J]. Journal of Physics: Conference Series, 2009, 180(1): 012043

[45] 葛震. GPU加速PQMRCGSTAB算法研究[D]. 长沙：国防科学技术大学，2009

[46] 吴强. GPU加速高速粒子碰撞模拟[D]. 长沙：国防科学技术大学，2009

[47] Fang Xu-dong, Tang Yu-hua, Wang Gui-bin, et al. Optimizing stencil application on multi-thread GPU architecture using stream programming model[C]// Muller-Schloer C, Karl W, Yehia S, eds. ARCS. LNCS 5974. 2010: 234-245

[48] Ma An-guo, Cai Jing, Cheng Yu, et al. Performance Optimization Strategies of High Performance Computing on GPU[C]// Dou Y, Gruber R, Joller J, eds. APPT. LNCS 5737. 2009: 150-164

[49] Chen Fei-guo, Ge Wei, Guo Li, et al. Multi-scale HPC system for multi-scale discrete simulation—Development and application of a supercomputer with 1 Petaflops peak performance in single precision[J]. Particuology, 2009, 7: 332-335

[50] Asanovic K, Bodik R, Catanzaro B, et al. The Landscape of Parallel Computing Research: A View from Berkeley[R]. California: Electrical Engineering and Computer Sciences University of California at Berkeley, 2006

[51] Bell G. Ultracomputers: a teraflop before its time[J]. Communication of the ACM, 1992, 35(8): 26-47

[52] Gupta A, Kumar V. Scalability of Parallel Algorithms for Matrix Multiplication[C]// 1993 International Conference on Parallel Processing. New York: IEEE, 1993: 115-123

[53] Sun Xian-he, Rover D T. Scalability of Parallel Algorithm- Machine Combinations[J]. IEEE Transactions on Parallel and Distributed Systems, 1994, 5(6): 599-613

[54] Kumar V, Gupta A. Analysis of scalability of parallel algorithms and architectures: A survey[C]// International Conference on Supercomputing. Cologne: ACM, 1991: 396-405