ACJT 群签名方案中成员撤消的高效实现*

陈泽文[1], 王继林[2,3], 黄继武[1+], 王育民[3], 黄达人[1]

[1](中山大学 信息科学与技术学院,广东 广州 510275)

[2](浙江财经学院 信息学院,浙江 杭州 310012)

[3](西安电子科技大学 ISN 重点国家实验室,陕西 西安 710071)

An Efficient Revocation Algorithm in ACJT Group Signature

CHEN Ze-Wen[1], WANG Ji-Lin[2,3], HUANG Ji-Wu[1+], WANG Yu-Min[3], HUANG Da-Ren[1]

[1](School of Informational Science and Technology, Sun Yat-Sen University, Guangzhou 510275, China)

[2](Department of Information, Zhejiang University of Finance and Economics, Hangzhou 310012, China)

[3](National Key Laboratory of ISN, Xidian University, Xi'an 710071, China)

+ Corresponding author: E-mail: isshjw@zsu.edu.cn, http://sist.zsu.edu.cn/graduate/bodao/huangjiwu.htm

Received 2003-09-05; Accepted 2004-02-11

Chen ZW, Wang JL, Huang JW, Wang YM, Huang DR. An efficient revocation algorithm in ACJT group signature. *Journal of Software*, 2005,16(1):151−157. http://www.jos.org.cn/1000-9825/16/151.htm

Abstract: The problem of secure and efficient revocation of membership without incurring big costs has been considered, but no satisfactory solution was reported. This paper proposes a new revocation method of membership based on the ACJT group scheme. The solution is efficient in that it only needs one multiplication to update the public key for the group manager to exclude one group member, and the signing and verifying procedure is independent of the number of the current and excluded group members. To the best of our knowledge, the signing and verifying procedure in the existing revocation schemes is dependent on the number of either the current or the excluded group members, and thus the group manager needs a heavy computation load to update the public key.

Key words: group signature; revocation; ACJT scheme; co-prime; zero-knowledge proof

摘 要: 成员撤消问题是设计群签名方案中的一个难题,到目前为止尚无满意的解决办法.在 ACJT 群签名方案的基础上,提出了新的成员撤消方法.在新方案中,管理员在撤消一个成员时仅需要一次乘法运算来更新群公钥,签名和验证算法的计算量均独立于目前群成员个数和被撤消的成员个数,因而算法是高效的.以前的具有撤消成员功能的群签名方案,签名和验证算法的计算量要么依赖当前的群成员个数,要么依赖被撤消的群成员个数,而且群公钥的更新或者成员密钥的更新往往需要多次指数运算.

关键词: 群签名;撤消;ACJT 方案;互素;零知识证明

* Supported by the National Natural Science Foundation of China under Grant Nos.60133020, 69975011, 60172067 (国家自然科学基金); the Natural Science Foundation of Guangdong Province of China under Grant No.04205407 (广东省自然科学基金)

作者简介: **陈泽文**(1975−),男,福建惠安人,博士,主要研究领域为信息安全;**王继林**(1965−),男,博士,副教授,主要研究领域为电子商务安全;**黄继武**(1962−),男,博士,教授,博士生导师,主要研究领域为多媒体信息安全;**王育民**(1936−),男,教授,博士生导师,主要研究领域为信息论,编码,密码学;**黄达人**(1945−),男,教授,博士生导师,主要研究领域为小波理论及应用.

中图法分类号: TP309 文献标识码: A

1 群签名中成员撤消问题的研究状况

在一个群签名方案中,一个群体中的任意一个成员可以以匿名的方式代表整个群体对消息进行签名,并且在有争议的时候,群管理者可以确定签名者的身份.群签名方案有很多的应用场合,比如电子现金系统[1]、投票协议[2]等.然而群签名要真正在实际中得到应用,还有一些问题需要解决.其中很重要的一个问题就是如何高效地撤消群成员.

Bresson 和 Stern[3]基于 Camenisch 和 Stadler[4]的群签名提出了第 1 个具有撤消成员功能的群签名方案,但该方案要求签名的长度线性依赖于被撤消的成员个数,而且文献[4]中的群签名方案后来被发现存在安全性问题.Song[5]基于 ACJT 群签名[6]提出了两个值得注意的撤消方案,因为除了能实现撤消功能以外,方案还具有前向安全性,而且签名长度是定长的,但其验证算法仍然线性依赖被撤消的成员个数.Ateniese 和 Tsudik[7]提出了另外一个签名长度独立于被撤消的成员个数的撤消方案,但与 Song 方案一样,其验证算法也线性依赖被撤消的成员个数,而且其使用的双重离散对数方法,严重地增加了签名和验证的计算量.Kim 等人也提出了一种撤消方案[8],但已经被证明存在安全性问题.Camenisch 和 Lysyanskaya[9]给出了一个动态累加器,该累加器允许动态加入和删除数据,并利用该累加器实现了 ACJT 群签名中的成员撤消问题以及 Camenisch 和 Lysyanskaya[10]的匿名 credential 系统中的成员撤消问题.改进后的验证算法,在计算量上仅需增加一个小于 2 的常数因子.虽然改进后的方案有很多优点,但还存在下述不足:(1) 每撤消一个成员,剩余的群成员必须更新自己的证据,而这仍然是线性依赖于被撤消的成员个数;(2) 证明一个承诺值在累加器中的计算量很大且复杂;(3) 验证者要定期查看群公钥;(4) 群管理员每次撤消成员要经过很多次指数运算来更新群公钥.

因此,到目前为止,以前具有撤消成员功能的群签名中签名和验证算法的计算量要么依赖于当前的群成员个数,要么依赖于被撤消的成员个数,这种线性依赖给群签名方案造成了下述一个或多个负担:(1) 群管理员需要做很多次指数运算来更新群公钥;(2) 群成员证明自己的成员证书不在被撤消者集合中的计算量随撤消人数的增加而增加;(3) 验证者收到签名后验证签名者不是已撤消的成员的计算量随撤消人数的增加而增加.

本文基于 ACJT 群签名方案,提出一个新的撤消方法.与以前的方案相比,新方案更为高效:(1) 签名长度固定;(2) 撤消一个成员,群管理者只需做一次乘法运算来更新群公钥,而以前方案需要做多次的指数运算;(3) 签名和验证算法均独立于目前的成员个数和撤消的成员个数.

本文第 2 节给出预备知识.第 3 节描述 ACJT 群签名方案.第 4 节给出互素的零知识证明并应用于新的撤消方案.新方案的安全性分析在第 5 节给出.最后是结论.

2 预备知识

设 $G=\langle g\rangle$是有限循环群,其阶$\#G$未知,但阶的最大二进制长度 l_G 是公开的.$y\in G$ 对 g 的离散对数是满足 $y=g^x$ 的整数 $x\in Z$.有一个无碰撞的 hash 函数 $H:\{0,1\}^*\rightarrow\{0,1\}^k$,该函数把任意长度的二进制串映射为一个 k-bit 的 hash 值.

2.1 困难性假设

假设 1(强 RSA 假设). 存在一个概率算法 T,使得对所有的概率多项式时间的算法 A、所有的多项式 $p(\cdot)$ 和所有充分大的 l_g:

$$Pr[z=u^e\,|\,(G,z):=T(1^{l_g}),(u,e>1):=A(G,z)]<1/p(l_g).$$

假设 2(Diffie-Hellman 假设). 不存在这样的概率多项式时间算法,该算法能以不可忽略的概率区分出分布 D 和 R,这里,$D=(g,g^x,g^y,g^z)$, $x,y,z\in_R Z_{\#G}$, $R=(g,g^x,g^y,g^{xy})$, $x,y\in_R Z_{\#G}$.

2.2 知识签名

很多群签名方案使用了知识签名这一工具,它允许一方在不泄露任何有用信息的情况下证明他知道一个

秘密值,这种工具本质上是知识的零知识证明或最小泄露证明,其表示形式是基于 Schnorr 签名[11].

离散对数的知识签名.给定 $y\in G$,一对 $(c,s)\in\{0,1\}^k\times\pm\{0,1\}^{\varepsilon(l_G+k)+1}$ 称为 y 对 g 关于消息 m 的知识签名,如果它满足 $c=H(m\|y\|g\|g^s y^c)$.

若知道了私钥 x,这样的签名可计算如下:

随机地选择 $r\in\pm\{0,1\}^{\varepsilon(l_G+k)}$,计算 $c=H(m\|y\|g\|g^r)$ 和 $s:=r-cx$.

两个离散对数相等的知识签名.一对 $(c,s)\in\{0,1\}^k\times\pm\{0,1\}^{\varepsilon(l_G+k)+1}$ 称为是两个离散对数 $y_1=g^x$ 和 $y_2=h^x$ 的知识对消息 m 的签名,如果它满足方程 $c=H(m\|y_1\|y_2\|g\|h\|g^s y_1^c\|h^s y_2^c)$.

若知道私钥 x,该签名可计算如下:

随机选取 $r\in\pm\{0,1\}^{\varepsilon(l_G+k)}$,计算 $c=H(m\|y_1\|y_2\|g\|h\|g^r\|h^r)$ 和 $s:=r-cx$.

离散对数在一个给定区间的知识签名.一对 $(c,s)\in\{0,1\}^k\times\pm\{0,1\}^{\varepsilon(l_G+k)+1}$ 称为是 y 关于底 g 的离散对数在区间 $[X-2^l,X+2^l]$ 对消息 m 的签名,如果它满足 $c=H(m\|y\|g\|g^{s-cX}y^c)$.

若知道 $x\in[X-2^l,X+2^l]$,该签名可以这样计算:

选择随机数 $r\in\pm\{0,1\}^{\varepsilon(l_G+k)}$,计算 $c=H(m\|y\|g\|g^r)$ 和 $s:-r-c(x-X)$.

3 ACJT 方案概述

本节我们将概述 ACJT 方案[6].在强 RSA 和 DDH 假设下,该方案被证明是安全的,且能抗击联合攻击.

设 $\varepsilon>1,k$ 和 l_p 为安全参数,定义两个整数区间 $\Delta=[2^{\lambda_1}-2^{\lambda_2},2^{\lambda_1}+2^{\lambda_2}]$, $\Gamma=[2^{\gamma_1}-2^{\gamma_2},2^{\gamma_1}+2^{\gamma_2}]$.这里,$\lambda_1>\varepsilon(\lambda_2+k)+2$, $\lambda_2>4l_p$, $\gamma_1>\varepsilon(\gamma_2+k)+2$, $\gamma_2>\lambda_1+2$.

1) 初始化

a) GM 随机地秘密选择 l_p 比特的素数 p',q' 使得 $p=2p'+1$, $q=2q'+1$ 为素数.令 $n=pq$.

b) GM 随机选择 $a,a_0,g,h\in_R QR(n)$.

c) GM 随机地秘密选择 $x\in Z_{p'q'}$,令 $y=g^x \bmod n$.

d) 群公钥为 $Y=(n,a,a_0,y,g,h)$.

e) GM 的私钥为 $S=(p',q',x)$.

2) 成员加入

假定每个成员与管理员间的信道是安全的.其加入过程如下:

a) P_i 产生一个秘密值 $\tilde{x}_i\in[0,2^{\lambda_1}]$,一个随机整数 $\tilde{r}\in[0,n^2]$,计算 $C_1=g^{\tilde{x}_i}h^{\tilde{r}}$,把 C_1 发送给 GM,并证明 C_1 的正确性.

b) GM 检查 $C_1\in QR(n)$.若是 GM 随机选择 α_i 和 $\beta_i\in[0,2^{\lambda_1}]$ 送 (α_i,β_i) 给 P_i.

c) P_i 计算 $x_i=2^{\lambda_1}+(\alpha_i\tilde{x}_i+\beta_i \bmod 2^{2^{\lambda_1}})$,把 $C_2=a^{x_i} \bmod n$ 送 GM. P_i 向 GM 证明:

I. C_2 对 a 的离散对数在 Δ 内;

II. 知道 u,v,w 使得(1) u 在 $[-2^{\lambda_1},2^{\lambda_1}]$ 内,(2) u 等于 $C_2/a^{2^{\lambda_1}}$ 对 a 的离散对数,(3) $C_1^{\alpha_i}g^{\beta_i}$ 等于 $g^u(g^{2^{\lambda_2}})^v h^w$.

d) GM 检查 $C_2\in QR(n)$,如果是且上述证明正确,则 GM 选择一个素数 $e_i\in\Gamma$,计算 $A_i=(C_2a_0)^{1/e_i}$,并发送给 P_i 成员证书$[A_i,e_i]$.

e) P_i 验证 $a^{x_i}a_0=A_i^{e_i} \bmod n$.

3) 签名

a) 产生一个随机数 $w\in_R\{0,1\}^{2l_p}$,计算 $T_1=A_i y^w \bmod n$, $T_2=g^w \bmod n$, $T_3=g^{e_i}h^w \bmod n$.

b) 随机选择 $r_1\in\pm\{0,1\}^{\varepsilon(\gamma_2+k)}$, $r_2\in\pm\{0,1\}^{\varepsilon(\lambda_2+k)}$, $r_3\in\pm\{0,1\}^{\varepsilon(\gamma_1+2l_p+k+1)}$ 和 $r_4\in\pm\{0,1\}^{\varepsilon(2l_p+k)}$,计算

$$d_1=T_1^{r_1}/(a^{r_2}y^{r_3}),\ d_2=T_2^{r_1}/g^{r_3},\ d_3=g^{r_4},\ d_4=g^{r_1}h^{r_4}$$

$$c=H(g\|h\|y\|a_0\|a\|T_1\|T_2\|T_3\|d_1\|d_2\|d_3\|d_4\|m)$$

$$s_1 = r_1 - c(e_i - 2^{\gamma_1}), s_2 = r_2 - c(x_i - 2^{\lambda_1}), s_3 = r_3 - ce_i w, s_4 = r_4 - cw$$

c) 输出$(c,s_1,s_2,s_3,s_4,T_1,T_2,T_3)$

4) 验证

a) 计算:

$$c' = H(g \| h \| y \| a_0 \| a \| T_1 \| T_2 \| T_3 \| a_0^c T_1^{s_1 - c2^{\gamma_1}} / (a^{s_2 - c2^{\lambda_1}} y^{s_3}) \bmod n \|$$
$$T_2^{s_1 - c2^{\gamma_1}} / g^{s_3} \bmod n \| T_2^c g^{s_4} \bmod n \| T_3^c g^{s_1 - c2^{\gamma_1}} h^{s_4} \bmod n \| m)$$

b) 当且仅当 $c = c'$, $s_1 \in \pm\{0,1\}^{\varepsilon(\lambda_2+k)+1}$, $s_2 \in \pm\{0,1\}^{\varepsilon(\lambda_2+k)+1}$, $s_3 \in \pm\{0,1\}^{\varepsilon(\lambda_1+2\ell_p+k+1)+1}$, $s_4 \in \pm\{0,1\}^{\varepsilon(2\ell_p+k)+1}$ 时接受签名.

5) 打开

a) 通过验证算法验证签名的有效性.

b) 通过自己的私钥,管理者可以计算 $A_i = T_1 / T_2^x$,恢复 A_i.

c) 证明 $\log_g^y = \log_{T_2}^{(T_1 / A_i \bmod n)}$.

4 成员撤消的实现

在这一节里,在不影响签名的安全性的条件下,我们给出上述群签名方案的一个成员撤消办法.加入成员撤消功能后,群管理员要发布一个所有相应撤消成员证书(A_j,x_j,e_j)中 e_j 的乘积.任何成员签名时要给出自己的 e_i 与该乘积互素的零知识证明.一个成员被撤消,GM 仅需要一次乘法来更新群公钥,签名和验证的计算量均独立于目前的成员个数和被撤消的成员个数.

4.1 互素性证明

设 $n = pq$,其中 $p = 2p' + 1$, $q = 2q' + 1$ 为素数,并且素数 p', q' 为 l_p 比特.令 $\tilde{g}, \tilde{h} \in_R QR(n)$.假设有一个公开的数 E,证明者不知道 $p'q'$ 和 $\tilde{h}$ 关于 $\tilde{g}$ 的离散对数.若证明者想在不泄露秘密 e 的情况下证明 e 与 E 互素,则他可按照如下步骤达到目的:

证明者:

I) 利用扩展 GCD 算法计算 a 和 b,使得 $ae+bE=1$;

II) 任意选择整数 $w_1, w_2, w_3 \in \pm\{0,1\}^{2l_p}$;

III) 计算 $y_1 = \tilde{g}^{w_1}, y_2 = \tilde{g}^{w_2}, y_3 = \tilde{g}^{w_3}, y_4 = \tilde{h}^{-(aw_1+w_2+w_3)}, T_1 = \tilde{g}^e \tilde{h}^{w_1}, T_2 = T_1^a \tilde{h}^{w_2}, T_3 = (\tilde{g}^E)^b \tilde{h}^{w_3}$;

IV) 任意选择整数 $r_1, r_2, r_3, r_4, r_5, r_6, r_7 \in \pm\{0,1\}^{\varepsilon(2l_p+k)}$;

V) 计算 $R_1 = \tilde{g}^{r_1}, R_2 = \tilde{g}^{r_2}, R_3 = \tilde{g}^{r_3}, R_4 = \tilde{h}^{r_4}, R_5 = \tilde{g}^{r_5} \tilde{h}^{r_1}, R_6 = T_1^{r_6} \tilde{h}^{r_2}, R_7 = (\tilde{g}^E)^{r_7} \tilde{h}^{r_3}$;

VI) 计算 $c = H(\tilde{g} \| \tilde{h} \| E \| y_1 \| y_2 \| y_3 \| y_4 \| T_1 \| T_2 \| T_3 \| R_1 \| R_2 \| R_3 \| R_4 \| R_5 \| R_6 \| R_7)$;

VII) 计算 $s_1=r_1-cw_1, s_2=r_2-cw_2, s_3=r_3-cw_3, s_4=r_4+c(aw_1+w_2+w_3), s_5=r_5-ce, s_6=r_6-ca, s_7=r_7-cb$;

VIII) 公布$(y_1,y_2,y_3,y_4,T_1,T_2,T_3,c,s_1,s_2,s_3,s_4,s_5,s_6,s_7)$.

验证者(任何人):

I) 计算

$$c' = H(\tilde{g} \| \tilde{h} \| E \| y_1 \| y_2 \| y_3 \| y_4 \| T_1 \| T_2 \| T_3 \| \tilde{g}^{s_1} y_1^c \| \tilde{g}^{s_2} y_2^c \| \tilde{g}^{s_3} y_3^c \|$$
$$\tilde{h}^{s_4} y_4^c \| \tilde{g}^{s_5} \tilde{h}^{s_1} T_1^c \| T_1^{s_6} \tilde{h}^{s_2} T_2^c \| (\tilde{g}^E)^{s_7} \tilde{h}^{s_3} T_3^c).$$

II) 验证 $c' \stackrel{?}{=} c$.如果不等,停止,验证失败.

III) 若 $\tilde{g} = T_2 T_3 y_4$,则 e 与 E 互素.

我们说验证者不能得到关于 e 的任何有用信息,因为验证者只获得$(y_1,y_2,y_3,y_4,T_1,T_2,T_3,c,s_1,s_2,s_3,s_4,s_5,s_6,s_7)$,若验证者想获得 e,a,b,那么他就必须能解决离散对数.

利用 Camenisch 和 Stadler[4]有关离散对数知识签名的表示形式,上述证明可表述为

$$PK\{(\alpha,\beta,\gamma,\theta,\tau,\xi,\zeta): y_1 = \tilde{g}^{\theta} \wedge y_2 = \tilde{g}^{\tau} \wedge y_3 = \tilde{g}^{\xi} \wedge y_4 = \tilde{h}^{\zeta} \wedge T_1 = \tilde{g}^{\alpha} \tilde{h}^{\theta} \wedge T_2 = T_1^{\beta} \tilde{h}^{\tau} \wedge T_3 = (\tilde{g}^E)^{\gamma} \tilde{h}^{\xi} \wedge \tilde{g} = T_2 T_3 y_4\}.$$

引理 1. 在强 RSA 假设下, $PK\{(\alpha,\beta,\gamma,\theta,\tau,\xi,\zeta): y_1=\tilde{g}^{\theta}\wedge y_2=\tilde{g}^{\tau}\wedge y_3=\tilde{g}^{\xi}\wedge y_4=\tilde{h}^{\zeta}\wedge T_1=\tilde{g}^{\alpha}\tilde{h}^{\theta}\wedge T_2=T_1^{\beta}\tilde{h}^{\tau}\wedge T_3=(\tilde{g}^{E})^{\gamma}\tilde{h}^{\xi}\wedge\tilde{g}=T_2T_3y_4\}$ 仅能由某个使用与 E 互素的数的人给出.

证明:设 $d=\gcd(\alpha,E)$ 且 $d\neq1$,则 $d>1$.若 A 是能给出上述 PK 协议证明的攻击者,那么 A 能攻破强 RSA 假设,即能找到 z 和 $d>1$,使 $\tilde{g}=z^d$.

从 $y_4=\tilde{h}^{\zeta}, T_1=\tilde{g}^{\alpha}\tilde{h}^{\theta}, T_2=T_1^{\beta}\tilde{h}^{\tau}, T_3=(\tilde{g}^{E})^{\gamma}\tilde{h}^{\xi}$ 以及 $\tilde{g}=T_2T_3y_4$, A 能得到 $\tilde{g}=T_2T_3y_4=\tilde{g}^{\alpha\beta+E\gamma}\tilde{h}^{\beta\theta+\tau+\xi+\zeta}$.因为证明者不知道 $\tilde{h}$ 关于 $\tilde{g}$ 的离散对数,在离散对数假设下,必然有 $\tilde{g}=\tilde{g}^{\alpha\beta+E\gamma}$.即 $\tilde{g}=(\tilde{g}^{(\alpha\beta/d+E\gamma/d)})^{d}$,令 $z=\tilde{g}^{\alpha\beta/d+E\gamma/d}$,那么 $\tilde{g}=z^d$.即 z 是 $\tilde{g}$ 的 d 次方根,与强 RSA 假设矛盾.这就说明引理的正确性. □

定理 1. 在强 RSA 假设下,相应上述 PK 协议的交互协议是关于证明者知道一个数 α 与 E 互素的(诚实验证者)统计零知识证明.

证明:很容易看出,上述交互协议是统计零知识[12]的.

为了说明上述协议是一个知识证明,我们说明若有两个可接受数组,那么知识提取因子(knowledge extractor)可以恢复出一个与 E 互素的 α.令

$$(y_1,y_2,y_3,y_4,T_1,T_2,T_3,c,s_1,s_2,s_3,s_4,s_5,s_6,s_7)$$

和

$$(y_1,y_2,y_3,y_4,T_1,T_2,T_3,c',s_1',s_2',s_3',s_4',s_5',s_6',s_7')$$

为两个可接受数组.

因为 $R_1=\tilde{g}^{s_1}y_1^{c}=\tilde{g}^{s_1'}y_1^{c'}$,我们可以得到 $y_1^{c-c'}=\tilde{g}^{s_1'-s_1}$.让 $d_1=\gcd(c-c',s_1'-s_1)$,通过扩展的 GCD 算法,知道存在 u_1,v_1,使得 $u_1(c-c')+v_1(s_1'-s_1)=d_1$,因而 $\tilde{g}=\tilde{g}^{(u_1(c-c')+v_1(s_1'-s_1))/d_1}=(\tilde{g}^{u_1}y_1^{v_1})^{(c-c')/d_1}$.若 $d_1<c-c'$,那么 $\tilde{g}^{u_1}y_1^{v_1}$ 是 $\tilde{g}$ 的 $(c-c')/d_1$ 方根,这与强 RSA 的假设相矛盾,所以 $d_1=c-c'$,即可以计算整数 $\vartheta=(s_1'-s_1)/(c-c')$,使得 $y_1=\tilde{g}^{\theta}$.

$R_5=\tilde{g}^{s_5}\tilde{h}^{s_1}T_1^{c}=\tilde{g}^{s_5'}\tilde{h}^{s_1'}T_1^{c'}$ 可以重新写为 $\tilde{g}^{s_5'-s_5}\tilde{h}^{s_1'-s_1}=T_1^{c-c'}$. 因为 $\vartheta=(s_1'-s_1)/(c-c')$,我们可以得到 $\tilde{g}^{s_5'-s_5}=(T_1\tilde{h}^{-\theta})^{c-c'}$.类似上面的证明,可以得到 $\alpha=(s_5'-s_5)/(c-c')$,使得 $T_1=\tilde{g}^{\alpha}\tilde{h}^{\theta}$.

同样地,从 $R_2=\tilde{g}^{s_2}y_2^{c}=\tilde{g}^{s_2'}y_2^{c'}$ 和 $R_6=T_1^{s_6}\tilde{h}^{s_2}T_2^{c}=T_1^{s_6'}\tilde{h}^{s_2'}T_2^{c'}$,我们可以得到 β,τ,使得 $T_2=T_1^{\beta}\tilde{h}^{\tau}=\tilde{g}^{\alpha\beta}\tilde{h}^{\beta\theta+\tau}$.从 $R_3=\tilde{g}^{s_3}y_3^{c}=\tilde{g}^{s_3'}y_3^{c'}$ 和 $R_7=(\tilde{g}^{E})^{s_7}\tilde{h}^{s_3}T_3^{c}=(\tilde{g}^{E})^{s_7'}\tilde{h}^{s_3'}T_3^{c'}$,我们也可以获得 γ,ξ,使 $T_3=(\tilde{g}^{E})^{\gamma}\tilde{h}^{\xi}$.

因此从 $\tilde{g}=T_2T_3y_4$ 得到 $\tilde{g}=\tilde{g}^{\alpha\beta+E\gamma}\tilde{h}^{\zeta}$,通过引理 1 的证明,知道 α 跟 E 互素.这就给出了证明. □

4.2 ACJT群签名方案中成员撤消的实现

下面我们利用上述协议来构建 ACJT 中的成员撤消方案.

群签名方案由建立、加入、撤消、签名、验证和打开几个过程组成.在本节中,建立、加入和打开过程与 ACJT 方案一样.因此我们下面只给出撤消、签名和验证过程.

撤消.假设 $E_{\text{delete}}:=\{G_1,G_2,\ldots,G_m\}$ 为现有撤消成员集合.群管理者计算 $E:=e_{G1}\ldots e_{Gm}$ 并公布.其中 e_{Gi} 是对应成员 G_i 证书的素数.当有一些成员要被撤消,令 E' 是欲被撤消成员的 e_{Gj} 的乘积,群管理者通过计算 $E:=EE'$ 来更新群公钥,并在一个公共的目录上公布最新的 E、当前的时间 t 和所有撤消成员的 e_{Gi}.

签名.对一个未被撤消的群成员来讲,假如他要对消息签名,那么首先他必须证明拥有成员证书(A_i,e_i,x_i),这可以通过 ACJT 方案得到.其次他必须通过公共的目录得到最新的 E 和时间 t,并证明成员证书(A_i,e_i,x_i)中的 e_i 是跟 E 互素,这可以通过定理 1 得到.即成员资格的证明可以通过下面获得:

i. 通过扩展的 GCD 算法可以计算出 a,b 使得 $ae_i+bE=1$.

ii. 任意选择随机数 $u,v,w\in_R\{0,1\}^{2l_p}$ 并计算:

iii. 计算 $T_1=A_iy^{u}, T_2=g^{u}, T_3=g^{e_i}h^{u}, T_4=g^{v}, T_5=T_3^{a}h^{v}, T_6=g^{w}, T_7=(g^{E})^{b}h^{w}, T_8=h^{-(au+v+w)}$.

iv. 生成:

$$PK1\{(\alpha,\beta,\delta,\varepsilon): a_0=T_1^{a}(1/a)^{\beta}(1/y)^{\delta}\wedge T_2=g^{\varepsilon}\wedge1=T_2^{a}(1/g)^{\delta}\wedge T_3=g^{\alpha}h^{\varepsilon}\wedge\alpha\in\Gamma\wedge\beta\in\Delta\},$$

$$PK2\{(\alpha,\eta,\gamma,\varepsilon,\tau,\xi,\zeta): T_2=g^{\varepsilon}\wedge T_4=g^{\tau}\wedge T_6=g^{\xi}\wedge T_8=h^{\zeta}\wedge T_3=g^{\alpha}h^{\varepsilon}\wedge T_5=T_3^{\eta}h^{\tau}\wedge T_7=(g^{E})^{\gamma}h^{\xi}\wedge g=T_5T_7T_8\wedge\alpha\in\Gamma\}.$$

数组$(t,T_1,T_2,T_3,T_4,T_5,T_6,T_7,T_8,PK1,PK2)$就是群成员的签名.

第 1 个协议 $PK1$ 与 ACJT 方案一样,是用来证明拥有成员证书(A_i,e_i,x_i).第 2 个协议 $PK2$ 是用来证明 $PK1$ 协议中的 e_i 确实跟 E 互素.需要说明的是,$PK2$ 与第 4.1 节稍微有点不同,主要是证明者需要把时间 t 加入到 hash 值当中,即在 VI)中计算 $c=H(t\|\tilde{g}\|\tilde{h}\|E\|y_1\|y_2\|y_3\|y_4\|T_1\|T_2\|T_3\|R_1\|R_2\|R_3\|R_4\|R_5\|R_6\|R_7)$,而不是原来那样.验证做相应的修改即可.

验证.验证者首先通过时间 t,找到相对应的 E,然后检查 $PK1$,$PK2$ 的正确性.

从上面的过程我们可以看到,我们的方案比以前的方案更为高效,因为群管理者只需做一次乘法运算就可以撤消一个成员,并且其他成员不需要更新自己的密钥.签名和验证过程是独立于现有的成员个数和撤消成员个数.

5 安全性分析

本节我们将证明,加入撤消功能后的方案满足群签名的所有安全性要求.

定理 2[6].假设发布的成员证书的个数 K 是多项式有界的,则在强 RSA 假定下,满足的成员证书(A_i,e_i,x_i)仅能由 GM 生成,其中 $x_i\in\Delta,e_i\in\Gamma$.

定理 3[6].在强 RSA 假设下,相应 $PK1$ 的交互式协议是关于(A_i,e_i,x_i)的统计零知识证明.

下面讨论所提出的群签名方案的安全性.

正确性.可以通过验证 $PK1$ 和 $PK2$ 的正确性来获得.

不可伪造性.如果一个用户既不是群成员,也不是一个撤消群成员,那么从定理 2 可以知道,他不可能产生(A_i,e_i,x_i),使得 $a_0a^{x_i}=A_i^{e_i}$.若他是一个撤消成员,即他拥有$(A_i,e_i,x_i),e_i|E$,则从定理 1 可以看出,他不可能证明 e_i 与 E 互素.因此只有群成员才可以代表群体签名.

可跟踪性.可以通过计算 $A_i=T_1/T_2^x$ 来确定实际签名者的身份.

抗攻击性.这可以从引理 1 和定理 2 得到.

抗陷害性.群成员和群管理者都不能代表其他成员进行签名.首先,从定理 2 可以知道群成员不能代表其他群成员进行签名.其次,群管理者不能从 a^{x_i} 得到一个群成员的密钥 x_i.

不可链接性.因为 $PK1$,$PK2$ 没有泄漏有用信息,$(T_1,T_2,T_3,T_4,T_5,T_6,T_7,T_8)$是无条件与随机数 u,v,w 绑定在一起的,因而要决定不同的签名来自同一个签名者在计算上是困难的.下面说明,即使当一个成员被撤消后,即他的 e_i 被公开,其他成员也不能把他跟他过去的签名链接在一块.假设$(T_1,T_2,T_3,T_4,T_5,T_6,T_7,T_8)$和$(T_1',T_2',T_3',T_4',T_5',T_6',T_7',T_8')$是两个签名,当 e_i 公开时,其他人可能知道 a 和 b,使 $ae_i+bE=1$.若他们想确定这两个签名是否为同一个成员产生,他们必须确定 $\log_y^{T_1/T_1'}$, $\log_g^{T_2/T_2'}$ 跟 $\log_g^{T_3/T_3'}$ 是否相等或者 $\log_h^{T_3T_3'^a/(T_3'T_3^a)}$ 跟 $\log_g^{T_4/T_4'}$ 是否相等或者 $\log_g^{T_6/T_6'}$ 跟 $\log_h^{T_7/T_7'}$ 是否相等.在 Diffie-Hellman 假设下,这是不可能的.

匿名性.给定一个合法的签名$(T_1,T_2,T_3,T_4,T_5,T_6,T_7,T_8,PK1,PK2)$,从定理 1 和定理 3 可以知道,$PK1$,$PK2$ 并没有泄漏信息.在离散对数假设下,可以知道从 $T_1,T_2,T_3,T_4,T_5,T_6,T_7,T_8$,获得$(A_i,e_i,x_i)$是困难的.当一个成员的 e_i 被泄漏时,类似不可链接性的分析,可以知道匿名性仍然满足.

可撤消性.对一个拥有证书(A_i,e_i)的被撤消的成员来说,他有两种可能的方式来证明其是合法用户:(1) 选择一个与 e_i 不同且与 E 互素的 e 来证明成员资格,根据定理 2,这是无法实现的.(2) 还用原来的 e_i 来证明成员资格,根据定理 1 和引理 1,这也是无法实现的.

6 结 论

我们在 ACJT 方案中增加了成员撤消功能,实现思想是,管理员发布被撤消成员的 e_i 的乘积,合法用户证明自己的 e_j 与该乘积互素.我们的实现办法比既有方案的效率要高,因为方案改进后,签名长度是固定的,GM 删除一个成员时仅需一次乘法,而且验证过程独立于当前成员个数和被删除的成员个数.与原始的不具有撤消功能的 ACJT 方案相比,我们加入撤消功能后的方案仅有下述变化:签名者在签名前必须查看群公钥,每次签名要增加 8 次指数运算,且给出一个与当前的 E 互素的知识证明;验证者要定期查看群公钥.所给的办法可以推广到 Camenisch 和 Lysyanskayas 的 credential 系统[10]的撤消实现中.

致谢 在此,我们向对本文的工作给予支持和建议的同行,尤其是西安电子科技大学的张键红、伍前红博士以及实验室的其他同学和老师表示感谢.

References:

[1] Lysyanskaya A, Ramzan Z. Group blind digital signatures: A scalable solution to electronic cash. In: Financial Cryptography (FC'98). LNCS 1465, Heidelberg: Springer-Verlag, 1998.184−197.

[2] Nakanishi T, Fujiwara T, Watanabe H. A linkable group signature and its application to a fair secret voting. Trans. IPS. Japan, 1999, 40(7):3085−3096.

[3] Bresson E, Stern J. Efficient revocation in group signature. In: Proc. of the PKC'01. LNCS 1992, Heidelberg: Springer-Verlag, 2001. 190−206.

[4] Camenish J, Stadler M. Efficient group signatures for large groups. In: Proc. of the CRYPTO'97. LNCS 1296, Heidelberg: Springer-Verlag, 1997. 410−424.

[5] Song D. Practical forward secure group signature schemes. In: Proc. of the 8th ACM Conf. on Computer and Communication Security (CCS 2001). ACM, 2001. 225−234.

[6] Ateniese G, Camenisch J, Joye M, Tsudik G. A practical and provably secure coalition-resistant group signature scheme. In: Advances in Cryptology- CRYPTO 2000. LNCS 1880, Heidelberg: Springer-Verlag, 2000. 255−270.

[7] Ateniese G, Tsudik G. Quasi-Efficient revocation of group signature. 2001. http://eprint.iacr.org/2001/101/

[8] Kim HJ, Lim JI, Lee DH. Efficient and secure member deletion in group signature schemes. In: Won D, ed. Proc. of the ICISC 2000. LNCS 2015, Heidelberg: Springer-Verlag, 2001. 150−161.

[9] Camenisch J, Lysyanskaya A. Dynamic accumulators and application to efficient revocation of anonymous credentials. In: Advances in Cryptology—CRYPTO 02. LNCS 2442, Heidelberg: Springer-Verlag, 2002. 61−77.

[10] Camenisch J, Lysyanskaya A. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. In: Advances in Cryptology—EUROCRYPT 01. LNCS 2045, Heidelberg: Springer-Verlag, 2001. 93−118.

[11] Schnorr CP. Efficient identification and signature for smart cards. In: Proc. of the Crypto'89. LNCS 435, Heidelberg: Springer-Verlag, 1990. 239−252.

[12] Camenisch J, Michels M. A group signature scheme based on an RSA-variant. Technical Report, RS-98-27, BRICS, University of Aarhus, 1999.