具有不同代理群的多代理多签名方案

刘丹妮[1,2] 王兴伟[1] 黄 敏[1]

(东北大学信息科学与工程学院 沈阳 110004)[1] (大连东软信息学院计算机科学与技术系 大连 116023)[2]

摘 要 目前在大多数多代理多签名(multi-proxy multi-signature,MPMS)方案中,所有的原始签名人指定同一个代理群生成代理签名。然而在很多实际应用中,每个原始签名人经常需要在自己所在的组织中选择代理群,从而导致各个原始签名人需要选择不同的代理群,代表自己生成代理签名。现有的多代理多签名方案中还没有考虑这方面的需求。提出一个可以指定不同代理群的MPMS方案。在该方案中,可以指定一个多人的验证群合作验证最终签名的合法性。为证明方案的安全性,对现有的安全模型进行改进,在计算Diffie-Hellman假设的基础上,验证所提方案是安全的。与已有方案相比,本方案更加安全和有效。

关键词 密码学,多代理多签名,代理群

中图法分类号 TP393 **文献标识码** A

Provably Secure Multi-proxy Multi-signature Scheme with Different Proxy Groups

LIU Dan-ni[1,2] WANG Xing-wei[1] HUANG Min[1]

(School of Information Science & Engineering, Northeastern University, Shenyang 110004, China)[1]

(Department of Computer Science & Technology, Dalian Neusoft Institute of Information, Dalian 116023, China)[2]

Abstract In most of the existing multi-proxy multi-signature (MPMS) schemes, the same proxy group is delegated the proxy right to sign by all the original members. Nevertheless, in many practical applications, original signer often demands to designate the proxy group in his own organization which is different from others'. It is seldom considered in the MPMS schemes. In this paper, we proposed a MPMS scheme with different proxy groups. Furthermore, in our scheme, when the final proxy signature is being authenticated, a group of specified verifiers have the access. To proved the safety of the new scheme, we improved a security model to testify that the new one is secure based on the computational Diffie-Hellman assumption. Compared with the previous scheme, the new one offers tighter safety and better computational efficiency.

Keywords Cryptography, Multi-proxy multi-signature (MPMS), Proxy group

多重签名的概念是由Itakura等人[1]首次提出的。在多重签名方案中,多个签名人合作对消息进行签名,验证者验证该签名是否由所有成员合作生成。

2002年,Tzeng等人[2]提出了多代理多签名(multi-proxy multi-signature,MPMS)方案。在方案中,一群原始签名人可以将代理权限授予一群代理签名人。在现实生活中,有很多MPMS方案的应用。例如,在一个大的楼群中,建造者和业主之间存在一些矛盾。所有的业主想要通过网络委托几个律师作为代理。于是一个律师群获得了所有业主的授权,代表业主行使权力[3]。MPMS方案引起了许多学者的关注。

Tzeng等人提出了多个验证者共同参与验证的不可否认门限多代理多签名方案[4]。方案中,原始签名群中的部分人可以将签名权限授予一群指定的代理签名人。但是,Bao等人[5]指出文献[4]的方案不具有代理保护和签名不可伪造的特性。2009年,Hsu等人[6]指出文献[4]的方案不能阻止内部攻击。文献[5,6]分别对文献[4]中的方案进行了改进。Mashhadi[7]进一步地对Hsu等人的方案进行了改进,使其能够阻止框架攻击。

文献[4-7]的方案中,原始签名群的所有成员将代理权限授予同一个代理群。实际应用中经常需要多个原始签名人分别指定不同的代理群,将代理权限授予自己指定的代理群。针对这种需求,文中提出原始签名人可以指定不同代理群的多代理多签名方案。

目前的多代理多签名方案大多采用一个随机的验证者验证最终签名的有效性[8]。正如文献[4]中提到的,实际应用中经常需要指定几个验证者一起验证最终代理签名的有效性。针对这种需求,文中提出的多代理多签名方案具有多人合作验证最终签名的特性。

在文献[9-11]的基础上,构造了一个多代理多签名方案。方案中,每个原始签名人都可以指定自己的代理群,各个代理

到稿日期:2013-07-17 返修日期:2013-10-28 本文受国家杰出青年科学基金项目(61225012),国家自然科学基金项目(61070162,71071028,70931001),高等学校博士学科点专项科研基金优先发展领域课题(20120042130003),高等学校博士学科点专项科研基金课题(20100042110025,20110042110024),工信部物联网发展专项资金项目,中央高校基本科研业务费专项资金项目(N110204003)资助。

刘丹妮(1975-),女,博士生,主要研究方向为信息安全,E-mail:liudanni@neusoft.edu.cn;王兴伟(1968-),男,教授,博士生导师,主要研究方向为网络安全;黄 敏(1968-),女,教授,博士生导师,主要研究方向为生产计划及存储理论。

群可以不相同；由多人组成的验证群验证最终代理签名是否合法。

1 提出的多代理多签名方案

p 是一个大素数；G 和 G_T 是阶为 p 的群，$|G|=|G_T|=p$；g 是 G 的生成元。双线性对表示为 $e:G\times G\rightarrow G_T$。

1.1 系统初始化

在系统初始化(SystemInit)阶段，定义系统需要的参数。$G_o=(O_1,O_2,\cdots,O_l)$ 是包含有 l 个成员的原始签名人集合；$U_j=\{P_{j,1},P_{j,2},\cdots,P_{j,n_j}\}$ 是含有 n_j 个成员的代理签名人集合；$O_j(j=1,2,\cdots,l)$ 为 U_j 中成员分配代理权限；$G_v=(V_1,V_2,\cdots,V_m)$ 是含有 m 个验证者的集合。SEM(security mediator)是在线可信服务器[4]。W 是含有 n 个字符的字符串，记录 G_o、G_v 和 $U_j(j=1,2,\cdots,l)$ 中每个成员的身份信息。待签信息 M 是一个长度为 n 的字符串。选择随机数 $u',u_1,u_2,\cdots,u_n,v,\in G$，令 $u=(u_1,\cdots,u_i,\cdots,u_n)$。公开参数$(G,G_T,p,g,e,u',u,v)$。$H:\{0,1\}^*\in G$ 是哈希函数。

1.2 密钥生成

密钥生成(SecretGen)阶段，每个 $O_j(1\leqslant j\leqslant l)\in G_o$ 随机选择 $ox_j,oy_j\in Z_p^*$，设其私钥为 $Osk_j=(ox_j,oy_j)$，并计算其公钥为 $Opk_j=(OX_j,OY_j)=(g^{ox_j},g^{oy_j})$。同样地，每个 $P_{j,i}(1\leqslant i\leqslant n_j)\in U_j(j=1,\cdots,l)$ 的私钥和公钥分别为 $Psk_{j,i}=(px_{j,i},py_{j,i})$ 和 $Ppk_{j,i}=(PX_{j,i},PY_{j,i})=(g^{px_{j,i}},g^{py_{j,i}})$。每个验证者 $V_z(1\leqslant z\leqslant m)\in G_v$ 设定其私钥和公钥为 $Vsk_z=(vx_z,vy_z)$ 和 $Vpk_z=(VX_z,VY_z)=(g^{vx_z},g^{vy_z})$。

1.3 代理份额产生

代理份额产生(DelegationGen)阶段，G_o 中每个成员 O_j 将签名权限分配给自己的代理群 $U_j=\{P_{j,1},P_{j,2},\cdots,P_{j,n_j}\}$，使其代表 O_j 行使签名权利。每个 O_j 只对自己的代理群授权。各个原始签名人的代理群可以不相同。每个 O_j 选择随机数 $r_j\in Z_p^*$，生成 $\sigma_{W_j}=(\sigma_{W_{j1}},\sigma_{W_{j2}})$，其中，$\sigma_{W_{j1}}=g^{ox_joy_j}(u'\prod_{i=1}^{n}u_i^{w_i})^{r_j}$，$\sigma_{W_{j2}}=g^{r_j}$。$O_j$ 将代理份额(W,σ_{W_j},U_j)发送给代理群 U_j 和 SEM。

1.4 生成代理人的签名

生成代理人的签名(PSGofProxySigner)阶段，每个代理人 $P_{j,i}$ 选择随机数 $k_i\in Z_p^*$，并且计算 $h_i=H(M\parallel W,\sigma_{W_{j2}},g^{k_i})$。$P_{j,i}$ 代表 O_j 生成代理签名 $\sigma_{j,i}=(\sigma_{j,i1},\sigma_{j,i2},\sigma_{j,i3})$，其中，$\sigma_{j,i1}=\sigma_{W_{j1}}g^{px_{j,i}py_{j,i}}(Mv)^{h_ik_i}$，$\sigma_{j,i2}=\sigma_{W_{j2}}$，$\sigma_{j,i3}=g^{h_ik_i}$。

$P_{j,i}$ 将 $\sigma_{j,i}$ 和(W,σ_{W_j},U_j)发送给指定的接收人 c_j。c_j 负责收集 U_j 中所有人的代理签名。

生成群代理签名(PSGofProxyGroup)：指定的接收人 c_j 合成 U_j 中所有成员的代理签名，生成代理群 U_j 的代理签名 $\sigma_j=(\sigma_{j1},\sigma_{j2},\sigma_{j3})$，其中，

$$\sigma_{j1}=\prod_{i=1}^{n_j}\sigma_{j,i1},\sigma_{j2}=\prod_{i=1}^{n_j}\sigma_{j,i2},\sigma_{j3}=\prod_{i=1}^{n_j}\sigma_{j,i3}$$

$\sigma_j=(\sigma_{j1},\sigma_{j2},\sigma_{j3})$被发送给指定的接收者 C。

1.5 生成最终签名

生成最终签名(MPMSGen)阶段，收集者 C 生成最终的 MPMS $\sigma=(\sigma_1,\sigma_2,\sigma_3)$，其中，$\sigma_1=\prod_{j=1}^{l}\sigma_{j1}$，$\sigma_2=\prod_{j=1}^{l}\sigma_{j2}\cdot\prod_{z=1}^{m}VX_z$，$\sigma_3=\prod_{j=1}^{l}\sigma_{j3}\cdot\prod_{z=1}^{m}VY_z$。

1.6 共享验证

共享验证(SharedVer)阶段，收到签名 $\sigma=(\sigma_1,\sigma_2,\sigma_3)$，验证群 G_v 中的所有成员执行下面的操作：

每个验证者 $V_z(1\leqslant z\leqslant m)\in G_v$ 计算验证密钥 $\sigma_{vz}=(\sigma_{vz1},\sigma_{vz2})$，其中，$\sigma_{vz1}=(u'\prod_{i=1}^{n}u_i^{w_i})^{vx_z}$，$\sigma_{vz2}=(Mv)^{vy_z}$，$z=1,2,\cdots,m$。并将验证密钥发送给指定的接收者，该接收者是 G_v 中可信任的成员。接收者合成所有的验证密钥，生成 $\sigma_v=(\sigma_{v1},\sigma_{v2})$，其中，$\sigma_{v1}=\prod_{z=1}^{m}\sigma_{vz1}$，$\sigma_{v2}=\prod_{z=1}^{m}\sigma_{vz2}$。只有当式(1)成立时，验证群才会接受签名 $\sigma=(\sigma_1,\sigma_2,\sigma_3)$。

$$e(\sigma_1\cdot\sigma_{v1}\cdot\sigma_{v2},g)=\prod_{j=1}^{l}\prod_{i=1}^{n_j}(e(OX_j,OY_j)e(PX_{j,i},PY_{j,i}))\cdot e(u'\prod_{i=1}^{n}u_i^{w_i},\sigma_2)e(Mv,\sigma_3)\quad(1)$$

2 正确性分析

收到最终签名 $\sigma=(\sigma_1,\sigma_2,\sigma_3)$后，如果该签名能通过验证组 G_v 采用式(1)的验证，则该签名是有效的。

首先计算 $e(\sigma_1,g)$：

$$\begin{aligned}e(\sigma_1,g)&=\prod_{j=1}^{l}e(\sigma_{j1},g)\\&=\prod_{j=1}^{l}\prod_{i=1}^{n_j}(e(OX_j,OY_j)e(PX_{j,i},PY_{j,i}))\cdot e(u'\prod_{i=1}^{n}u_i^{w_i},\prod_{j=1}^{l}\sigma_{j2})e(Mv,\prod_{j=1}^{l}\sigma_{j3})\end{aligned}$$

然后计算 $e(\sigma_{v1},g)$：

$$\begin{aligned}e(\sigma_{v1},g)&=e(\prod_{z=1}^{m}\sigma_{vz1},g)\\&=e(\prod_{z=1}^{m}(u'\prod_{i=1}^{n}u_i^{w_i})^{vx_z},g)=e(u'\prod_{i=1}^{n}u_i^{w_i},g^{\sum_{z=1}^{m}vx_z})\\&=e(u'\prod_{i=1}^{n}u_i^{w_i},\prod_{z=1}^{m}VX_z)\end{aligned}$$

再次，计算 $e(\sigma_{v2},g)$：

$$\begin{aligned}e(\sigma_{v2},g)&=e(\prod_{z=1}^{m}\sigma_{vz2},g)=e(\prod_{z=1}^{m}(Mv)^{vy_z},g)\\&=e(v'\prod_{k\in\mathcal{M}}v_k,g^{\sum_{z=1}^{m}vy_z})=e(Mv,\prod_{z=1}^{m}VY_z)\end{aligned}$$

由上述 3 个式子可以推出：

$$\begin{aligned}&e(\sigma_1\cdot\sigma_{v1}\cdot\sigma_{v2},g)\\&=e(\sigma_1,g)\cdot e(\sigma_{v1},g)\cdot e(\sigma_{v2},g)\\&=\prod_{j=1}^{l}\prod_{i=1}^{n_j}(e(OX_j,OY_j)e(PX_{j,i},PY_{j,i}))\cdot e(u'\prod_{i=1}^{n}u_i^{w_i},\sigma_2)e(Mv,\sigma_3)\end{aligned}$$

综上所述，式(1)得证。

3 安全性分析

本方案的安全性分析建立在计算 Diffie-Hellman 假设的基础上。采用仿真的方法分析该方案是安全的。仿真过程是一个模拟敌手试图伪造签名的过程。在假设敌手能够伪造签名的前提下，得出 CDH(Computational Diffie-Hellman)问题被解决的结论。而事实上，CDH 问题是无法解决的，从而得出敌手能够伪造签名的假设是错误的。

文献[12]中给出了一个较好的安全模型，敌手的分类很清晰。为了使该安全模型适用于本方案，文中将敌手分了 3 类：

(1)第一类敌手 $\mathcal{A}_1$ 获取了所有原始签名人和代理签名人的公钥；

(2)第二类敌手 $\mathcal{A}_2$ 获取了所有原始签名人和代理签名人的公钥,此外还捕获了所有代理签名人和除 O_1 以外所有 $l-1$ 个原始签名人的私钥;

(3)第三类敌手 $\mathcal{A}_3$ 获取了所有原始签名人和代理签名人的公钥,此外还捕获了所有原始签名人和除 $P_{1,1}$ 以外所有代理签名人的私钥。

可知,如果该方案能够抵御第二类和第三类敌手的伪造攻击,则该方案也能抵御第一类敌手的伪造攻击。

3.1 抵制敌手 $\mathcal{A}_2$ 的攻击

定理 1 假设敌手 $\mathcal{A}_2$ 能够在时间 T 内,以 ε 的概率伪造签名,并且对 DelegationGen、PSGofProxy-Signer、PSGofProxy-Group 和 MPMSGen 等阶段发出请求的次数不超过 q_{DG}、q_{PS}、q_{PG} 和 q_{MS}。上述情况称为 $\mathcal{A}_2$ 能够对方案进行 $(T,q_{DG},q_{PS},q_{PG},q_{MS},\varepsilon)$-攻击。如果该情况能够发生,那么就可以构造出算法 $\mathcal{B}$,$\mathcal{B}$ 能对 CDH 问题进行 (T',ε')-攻击,其中,

$$\varepsilon' \geqslant \frac{1}{4(n+1)(q_{DG}+q_{PS}+q_{PG}+q_{MS})}\varepsilon,$$

$$T' = T+(2n+6+3l\cdot q_{DG}+3l\cdot q_{PS})T_e+(n+3+2l\cdot q_{DG}+(5l-1)\cdot q_{PS}+3\prod_{j=1}^{l}(n_j-1)\cdot q_{PG}+3(l-1)\cdot q_{MS})T_m$$

可以把 $\mathcal{A}_2$ 看作 $\mathcal{B}$ 的子程序。

证明:p 是大素数,G 是阶为 p 的 GDH 群。假设 $\mathcal{B}$ 接到一个群 G 的 CDH 问题实例 (g,g^a,g^b),要计算出 g^{ab}。$\mathcal{B}$ 作为 $\mathcal{A}_2$ 的挑战者,并且把 $\mathcal{A}_2$ 看作 $\mathcal{B}$ 的子程序。当 $\mathcal{A}_2$ 发出请求时,$\mathcal{B}$ 通过以下方式响应 $\mathcal{A}_2$。

系统初始化:$\mathcal{B}$ 设定 $l=2(q_{DG}+q_{PS}+q_{PG}+q_{MS})$。$\mathcal{B}$ 在 0 到 n 的范围内选择随机数 k。假设 $\mathcal{B}$ 获取了 q_{DG},q_{PS},q_{PG},q_{MS} 和 n 的值,其中 $l(n+1)<p$。然后 $\mathcal{B}$ 选择了随机数 x' 和一个 n 维向量 $\vec{x}=(x_i)(i\in[1,n])$,其中 $x',x_i\in Z_l$。$\mathcal{B}$ 随机选择 $y'\in Z_p$ 和 n 维向量 $\vec{y}=(y_i)(i\in[1,n])$,其中 $y_i\in Z_p$。$W=(w_1,w_2,\cdots,w_n)$ 是一个协议,并且由所有 $w_i=1$ 的元素生成集合 $\mathcal{W}$。$\mathcal{B}$ 设置 O_1 的公钥为 $Opk_1=(OX_1,OY_1)=(g^a,g^b)$。$g^a$ 和 g^b 都作为 CDH 问题的输入参数。

为了进一步分析,定义下面公式[11]:

$$F(W)=(p-lk)+x'+\sum_{j\in\mathcal{W}}x_j,\ J(W)=y'+\sum_{j\in\mathcal{W}}y_j$$

$$K(W)=\begin{cases}0, & x'+\sum_{j\in\mathcal{W}}x_i=0(\bmod\ l)\\ 1, & \text{otherwise}\end{cases}$$

设定公共参数 $u'=OY_1^{(p-lk)+x'}\cdot g^{y'}$,$u_j=OY_1^{x_j}\cdot g^{y_j}$ $(1\leqslant j\leqslant n)$。

因此,$u'\prod_{j\in\mathcal{W}}u_i^{w_i}=OY_1^{F(W)}g^{J(W)}$。

DelegationGen 请求:

(1)原始签名人 O_1 发出请求。$\mathcal{B}$ 选择随机数 $r_1\in Z_p^*$。如果 $K(W)\neq 0$,意味着 $F(W)\neq 0(\bmod\ p)$[13],$\mathcal{B}$ 通过计算 $\sigma_{W1}=(\sigma_{W_{11}},\sigma_{W_{12}})$ 生成 O_1 的代理授权,其中,$\sigma_{W_{11}}=OX_1^{\frac{I(W)}{F(W)}}(u'\prod_{i=1}^{n}u_i^{w_i})^{r_1}$,$\sigma_{W_{12}}=OX_1^{\frac{-1}{F(W)}}g^{r_1}$。

令 $\tilde{r}_1=r_1-\frac{a}{F(W)}$,可得出

$$\sigma_{W_{11}}=OX_1^{\frac{I(W)}{F(W)}}(u'\prod_{i=1}^{n}u_i^{w_i})^{r_1}$$

$$=OY_1^{a}(OY_1^{F(W)}g^{J(W)})^{\tilde{r}_1}=OY_1^{a}(u'\prod_{i=1}^{n}u_i^{w_i})^{\tilde{r}_1}$$

$$\sigma_{W_{12}}=OX_1^{\frac{-1}{F(W)}}g^{r_1}=g^{r_1-\frac{a}{F(W)}}=g^{\tilde{r}_1}。$$

如果 $K(W)=0$,仿真过程结束,$\mathcal{B}$ 返回错误提示信息。

(2)其他原始签名人发出请求。$\mathcal{B}$ 计算其他原始签名人的代理授权:

$$\sigma_{W_j}=(\sigma_{W_{j1}},\sigma_{W_{j2}}),\ \sigma_{W_{j1}}=g^{\alpha x_j oy_j}(u'\prod_{i=1}^{n}u_i^{w_i})^{r_j},$$

$$\sigma_{W_{j2}}=g^{r_j}\ (j=2,\cdots,l)。$$

PSGofProxySigner 请求:

(1)O_1 的 PSGofProxySigner 请求。如果 $K(W)=0$,仿真过程结束,$\mathcal{B}$ 返回错误信息。当 $K(W)\neq 0$ 时,由于 $\mathcal{B}$ 获得了 U_1 中成员的密钥,于是执行算法 PSGofProxySigner 得到 U_1 中每个代理人的 $\sigma_{1,i}=(\sigma_{1,i1},\sigma_{1,i2},\sigma_{1,i3})$。其中,$\sigma_{1,i1}=\sigma_{W_{11}}g^{\alpha x_{1,i}y_{1,i}}(Mv)^{h_ik_i}$,$\sigma_{j,i2}=\sigma_{W_{12}}$,$\sigma_{j,i3}=g^{h_ik_i}$。

(2)其他原始签名人的 PSGofProxySigner 请求。与 O_1 的 PSGofProxySigner 请求相似,$\mathcal{B}$ 能计算出 U_j 中其他代理签名人的签名信息 $\sigma_{j,i}=(\sigma_{j,i1},\sigma_{j,i2},\sigma_{j,i3})$:

$$\sigma_{j,i1}=\sigma_{W_{j1}}g^{\alpha x_{j,i}y_{j,i}}(Mv)^{h_ik_i},\ \sigma_{j,i2}=\sigma_{W_{j2}},$$

$$\sigma_{j,i3}=g^{h_ik_i}\ (j=2,\cdots,l;i=1,2,\cdots,n_j)。$$

PSGofProxyGroup 请求:

(1)O_1 的 PSGofProxyGroup 请求。如果 $K(W)=0$,仿真结束,$\mathcal{B}$ 返回错误提示信息。否则,如果 $K(W)\neq 0$,$\mathcal{B}$ 可以计算出代理群 U_1 的签名 $\sigma_1=(\sigma_{11},\sigma_{12},\sigma_{13})$:

$$\sigma_{11}=\prod_{i=1}^{n_1}\sigma_{1,i1}=(OX_1^{\frac{I(W)}{F(W)}}(u'\prod_{i=1}^{n}u_i^{w_i})^{r_1})^{n_1}\cdot g^{\sum_{i=1}^{n_1}\alpha x_{1,i}y_{1,i}}(Mv)^{\sum_{i=1}^{n_1}h_ik_i}$$

$$\sigma_{12}=\prod_{i=1}^{n_1}\sigma_{1,i2}=(OX_1^{\frac{-1}{F(W)}}g^{r_1})^{n_1}$$

$$\sigma_{13}=\prod_{i=1}^{n_1}\sigma_{1,i3}=g^{\sum_{i=1}^{n_1}h_ik_i}$$

(2)其他原始签名人的 PSGofProxyGroup 请求。与 O_1 的 PSGofProxyGroup 请求相似,$\mathcal{B}$ 能计算出其他代理群 U_j $(j=2,\cdots,l)$ 的代理签名 $\sigma_j=(\sigma_{j1},\sigma_{j2},\sigma_{j3})$,

$$\sigma_{j1}=\prod_{i=2}^{n_j}\sigma_{j,i1},\ \sigma_{j2}=\prod_{i=2}^{n_j}\sigma_{j,i2},\ \sigma_{j3}=\prod_{i=2}^{n_j}\sigma_{j,i3}$$

MPMSGen 请求:

最后攻击者 $\mathcal{A}_2$ 输出了多代理多签名 MPMS 为 $\sigma=(\sigma_1,\sigma_2,\sigma_3)$。

$$\sigma_1=\prod_{j=1}^{l}\sigma_{j1}=((OY_1^{a}(u'\prod_{i=1}^{n}u_i^{w_i})^{\tilde{r}_1})^{n_1}\cdot g^{\sum_{i=1}^{n_1}\alpha x_{1,i}y_{1,i}}(Mv)^{\sum_{i=1}^{n_1}h_ik_i}\cdot\prod_{j=2}^{l}\sigma_{j1}$$

$$\sigma_2=\prod_{j=1}^{l}\sigma_{j2}\cdot\prod_{z=1}^{m}VX_z=g^{\tilde{r}_1}\cdot\prod_{j=2}^{l}\sigma_{j2}\cdot\prod_{z=1}^{m}VX_z$$

$$\sigma_3=\prod_{j=1}^{l}\sigma_{j3}\cdot\prod_{z=1}^{m}VY_z=g^{\sum_{i=1}^{n_1}h_ik_i}\cdot\prod_{j=2}^{l}\sigma_{j3}\cdot\prod_{z=1}^{m}VY_z$$

下面的证明过程表明签名 σ 是正确的。首先计算 $e(\sigma_1,g)$:

$$e(\sigma_1,g)=\prod_{j=1}^{l}e(\sigma_{j1},g)$$

$$=e(OY_1^{a},g)\prod_{i=1}^{n_1}e(PX_{1,i},PY_{1,i})e(u'\prod_{i=1}^{n}u_i^{w_i},\prod_{j=1}^{l}\prod_{i=1}^{n_j}g^{r_j})\cdot\prod_{j=2}^{l}\prod_{i=1}^{n_j}(e(OX_j,OY_j)e(PX_{j,i},PY_{j,i}))e(Mv,\prod_{j=1}^{l}\prod_{i=1}^{n_j}g^{h_ik_i})$$

计算 $e(\sigma_{v1},g)$：

$$e(\sigma_{v1},g)=e(\prod_{z=1}^{m}\sigma_{wz1},g)=e(\prod_{z=1}^{m}(u'\prod_{i=1}^{n}u_i^{w_i})^{\omega_z},g)$$

$$=e(u'\prod_{i=1}^{n}u_i^{w_i},g^{\sum_{z=1}^{m}\omega_z})=e(u'\prod_{i=1}^{n}v_i^{w_i},\prod_{z=1}^{m}VX_z)$$

计算 $e(\sigma_{v2},g)$：

$$e(\sigma_{v2},g)=e(\prod_{z=1}^{m}\sigma_{wz2},g)=e(\prod_{z=1}^{m}(Mv)^{vy_z},g)$$

$$=e(Mv,g^{\sum_{z=1}^{m}vy_z})=e(Mv,\prod_{z=1}^{m}VY_z)$$

所以，

$$e(\sigma_1\cdot\sigma_{v1}\cdot\sigma_{v2},g)=\prod_{j=1}^{l}\prod_{i=1}^{n_j}(e(OX_j,OY_j)e(PX_{j,i},PY_{j,i}))\cdot$$

$$e(u'\prod_{i=1}^{n}u_i^{w_i},\sigma_2)e(Mv,\sigma_3)$$

伪造签名：假设仿真过程不会结束，$M^*\in\{0,1\}^*$，$W^*=(w_1^*,w_2^*,\cdots,w_n^*)$，则 $\mathscr{A}_2$ 以最低 ε 的概率生成伪造签名 $\sigma^*=(\sigma_1^*,\sigma_2^*,\sigma_3^*)$，具体步骤如下：

(1)$\mathscr{A}_2$ 没有发出过关于协议 W^* 的 DelegationGen 请求；

(2)$\mathscr{A}_2$ 没有发出过关于消息 M^* 和协议 W^* 的 PSGofProxySigner 请求；

如果 $F(W^*)\neq 0(\mathrm{mod}\ p)$，那么伪造过程结束。否则，如果 $F(W^*)=0(\mathrm{mod}\ p)$，那么 $u'\prod_{j\in\mathscr{W}}u_i^{w_i^*}=g^{J(W^*)}$。$\mathscr{B}$ 计算得出：

$$e(\sigma_1^*,g)=e(OY_1^a,g)\prod_{i=1}^{n_1}e(PX_{1,i},PY_{1,i})\prod_{j=2}^{l}\prod_{i=1}^{n_j}(e(OX_j,OY_j)$$

$$e(PX_{j,i},PY_{j,i}))\cdot e(u'\prod_{i=1}^{n}u_i^{w_i^*},\prod_{j=1}^{l}\prod_{i=1}^{n_j}g^{r_j})e(M^*v,$$

$$\prod_{j=1}^{l}\prod_{i=1}^{n_j}g^{h_ik_i})$$

$$\Rightarrow\sigma_1^*=g^{ab}\cdot(M^*v)^{\sum_{j=1}^{l}\sum_{i=1}^{n_j}(h_ik_i)}\cdot$$

$$g^{\sum_{i=1}^{n_1}(px_{1,i}+py_{1,i})+\sum_{j=2}^{l}\sum_{i=1}^{n_j}(ax_j+ay_j+px_{j,i}+py_{j,i})+J(W^*)\cdot\sum_{j=1}^{l}n_jr_j}$$

因此，$\mathscr{B}$ 可以计算出

$$g^{ab}=\sigma_1^*\cdot(M^*v)^{-\sum_{j=1}^{l}\sum_{i=1}^{n_j}(h_ik_i)}\cdot$$

$$g^{-(\sum_{i=1}^{n_1}(px_{1,i}+py_{1,i})+\sum_{j=2}^{l}\sum_{i=1}^{n_j}(ax_j+ay_j+px_{j,i}+py_{j,i})+J(W^*)\cdot\sum_{j=1}^{l}n_jr_j)}$$

于是，证明过程开始时提到的 CDH 问题解决了。证明结束。

在 Waters 方法的启发下[13]，下面给出解决 CDH 问题的概率分析。

假设 $W_1,W_2,\cdots,W_{q_w}$ 是上述请求过程中用到的协议，但是其中不包含 W^*；有 $q_w\leqslant q_{DG}+q_{PS}+q_{PG}+q_{MS}$。

需要满足下述条件：

A_i：在 DelegationGen、PSGofProxySigner、PSGofProxyGroup 和 MPMSGen 请求阶段，$K(W_i)\neq 0(\mathrm{mod}\ l)$，$i=1,2,\cdots,q_w$；

B：在 MPMSGen 请求阶段，$F(W^*)=0(\mathrm{mod}\ p)$。

因此，$\mathscr{B}$ 成功的概率为：

$$\Pr(\bigcap_{i=1}^{q_w}A_i\cap B)$$

$$=\Pr(B)\cdot\Pr(\bigcap_{i=1}^{q_w}A_i\mid B)=\Pr(B)\cdot(1-Pr(\bigcup_{i=1}^{q_w}\neg A_i\mid B))$$

$$\geqslant\Pr(B)\cdot(1-\sum_{i=1}^{q_w}\Pr(\bigcup_{i=1}^{q_w}\neg A_i\mid B))$$

$$=(\Pr(F(W^*)=0\ \mathrm{mod}\ p)\cdot(1-\sum_{i=1}^{q_w}\Pr(\bigcup_{i=1}^{q_w}\neg(K(W_i)\neq 0\ \mathrm{mod}\ l)\mid F(W^*)=0\ \mathrm{mod}\ p))$$

$$=\frac{1}{l}\cdot\frac{1}{n+1}\cdot(1-\frac{q_w}{l})\geqslant\frac{1}{(n+1)l}(1-\frac{q_{DG}+q_{PS}+q_{PG}+q_{MS}}{l})$$

$$=\frac{1}{4(n+1)(q_{DG}+q_{PS}+q_{PG}+q_{MS})}$$

假设 $\mathscr{A}_2$ 赢得这次对弈的概率不超过 ε，那么 CDH 问题被解决的概率为 $\varepsilon'\geqslant\frac{1}{4(n+1)(q_{DG}+q_{PS}+q_{PG}+q_{MS})}\varepsilon$。

$\mathscr{B}$ 时间消耗的计算量取决于各请求阶段的模幂和模乘运算。DelegatinGen 阶段、PSGofProxy-Signer 阶段、PSGofProxyGroup 阶段和 MPMSGen 阶段的请求次数分别为 q_{DG}、q_{PS}、q_{PG} 和 q_{MS}。初始化阶段需要 G 群上的 $2n+6$ 次模幂运算和 $n+3$ 次模乘运算。每次的 DelegatinGen 请求需要 $3l$ 次模幂运算和 $2l$ 次的模乘运算。每次的 PSGofProxySigner 请求需要 $3l$ 次的模幂运算和 $5l-1$ 次的模乘运算。PSGofProxyGroup 和 MPMSGen 请求分别需要 $3\prod_{j=1}^{l}(n_j-1)$ 和 $3l+2m-5$ 的模乘运算。因此整个仿真过程大约需要的时间为

$$T'=T+(2n+6+3l\cdot q_{DG}+3l\cdot q_{PS})T_e+(n+3+2l\cdot q_{DG}+(5l-1)\cdot q_{PS}+3\prod_{j=1}^{l}(n_j-1)\cdot q_{PG}+3(l-1)\cdot q_{MS})T_m$$

3.2 抵制敌手 $\mathscr{A}_3$ 的攻击

定理 2 假设敌手 $\mathscr{A}_3$ 能够在时间 T 内，以 ε 的概率伪造签名，并且对 PSGofProxySigner、PSGofProxyGroup 和 MPMSGen 等阶段发出请求的次数不超过 q_{PS}、q_{PG} 和 q_{MS}。上述情况称为 $\mathscr{A}_3$ 能够对方案进行 $(T,0,q_{PS},q_{PG},q_{MS},\varepsilon)$-攻击。如果该情况能够发生，那么就可以构造出算法 $\mathscr{B}$，$\mathscr{B}$ 能对 CDH 问题进行 (T',ε')-攻击，可以把 $\mathscr{A}_3$ 看作 $\mathscr{B}$ 的子程序。其中，

$$\varepsilon'\geqslant\frac{\varepsilon}{2n(q_{PS}+q_{PG}+q_{MS})},$$

$$T'=T+(2n+6+\prod_{j=1}^{l}(3n_j-1)\cdot q_{PS})T_e+(n+3+\prod_{j=1}^{l}5n_j\cdot q_{PS}+3(n_j-1)q_{PG}+(3l+2m-5)q_{MS})T_m$$

证明：证明过程与定理 1 有很多相似之处。限于篇幅，重点叙述二者区别之处。首先，$\mathscr{A}_3$ 拥有 G_0 和 $U_j(j=1,2,\cdots,l)$ 中所有成员的公钥；此外，它获取了除 $P_{1,1}$ 以外所有成员的私钥，$P_{1,1}$ 是 O_1 代理群中成员。所以 $\mathscr{A}_3$ 不需要发出 DelegationGen 请求，而且能够重构代理权限。再者，在初始化阶段 $P_{1,1}$ 的公钥就被设置为 $Ppk_1=(OX_1,OY_1)=(g^a,g^b)$，其中 g^a 和 g^b 是 CDH 问题的输入参数。

4 效率分析

目前，几乎所有的 MPMS 方案都是基于离散对数问题或大数因子问题。在密码学应用中，双线性对具有很好的特性，所以本方案应用了双线性对。文献[11]的方案也是基于双线性对的，下面对本方案和文献[11]的方案在效率方面进行比较和分析。

(下转第 163 页)

过身份认证"关口"之后在集群中的操作一直都受到监控，其安全性大大增强。

3)提高了集群访问控制机制的灵活性。LT 模型根据实时记录下来的用户操作行为计算用户信任值，一旦用户行为对集群安全性有危害，在信任值计算算法中促使信任值降低，直到信任值低于集群所设定的信任值阈值，不再为其提供服务，这样就可以灵活有效地中止用户继续其有害行为。

参 考 文 献

[1] 刘玮，王丽宏. 云计算应用及其安全问题研究[J]. 计算机研究与发展，2012，49：186-191

[2] 云计算百科. 什么是云计算平台？云计算平台有哪些？[EB/OL]. http://www.cloudwhy.com/ mingci/2011/0317/128.html，2012-06-12

[3] 韩伟，张福生，胡志勇. 基于 Hadoop 云计算平台下 DDoS 攻击防御研究[D]. 太原：太原科技大学，2011，07

[4] Hadoop. [EB/OL]. http://hadoop.apache.org/，2012-06-12

[5] Nutch. [EB/OL]. http://nutch.apache.org/，2012-06-12

[6] White T. Hadoop: The Definitive Guide(2nd edition)[M]. 2009-05

[7] it168.com. 浅谈 Hadoop 系统架构与海量数据分析[EB/OL]. http://wenku.it168.com/d_ 00076703.shtml，2012-06-12

[8] Becherer A. Attacking Kerberos and the New Hadoop Security Design[EB/OL]. http://www.ipma-wa.com/prof_dev/2011/HadoopSecurityDesign_201104_AndrewBecherer.pdf，2012-06-13

[9] Yahoo. Scaling Hadoop to 4000 nodes atYahoo! [EB/OL]. http://developer.yahoo.com/blogs/hadoop/scaling-hadoop-4000-nodes-yahoo-410.html，2008-09-30

[10] O'Malley O，Zhang Kan，Radia S. Hadoop Security Design [EB/OL]. http:/www.valleytalk.org/wp-content/uploads/2013/03/hadoop-security-design.pdf，2009-10

[11] Hadoop Releases [EB/OL]. http://hadoop.apache.org/common/releases.html，2012-06-14

[12] Yahoo，Hadoop 0.20.S Virtual Machine Appliance[EB/OL]. http://developer.yahoo.com/blogs/hadoop/hadoop-0-20-virtual-machine-appliance-460.html，2010-06-29

[13] Cloudera. CDH3 Security Guide [EB/OL]. https://ccp.cloudera.com/display/CDHDOC/CDH3+Security+Guide

[14] Chang Bao-rong，Tsai H F. Access Security on Cloud Computing Implemented in Hadoop System[C]//IEEE 2011 Fifth International Conference on Genetic and Evolutionary Computing. 2010，27：77-80

(上接第 136 页)

模幂和模乘运算决定了方案的效率。令 T_e 表示一次模幂运算的时间消耗；T_m 表示一次模乘运算的时间消耗。

两个方案进行比较的信息如表 1 所列。第一行表示最终签名 MPMS 的维数。第二行表示方案中所需要的系统参数的个数。ProxySigner 行和 MPMS 行分别表示每个代理人生成个人签名和最终生成 MPMS 的时间消耗。S. V. 行表示方案是否具有共享验证功能。文献[11]的方案需要 $2n+4$ 个系统参数，而文中方案只需要 $n+4$ 个系统参数。因为在 PSG-ProxySigner 阶段和 MPMSGen 阶段，时间消耗分别减少到 $3T_e+5T_m$ 和 $(3l-3)T_m$，所以新方案效率更高一些。新方案还具有共享验证的功能。总之，所提方案具有更好的可执行性。

表 1 时间消耗的比较

方案	文献[11]的方案	新方案
最终签名的维数	3G	3G
系统参数	2n+4	n+4
ProxySigner	$9T_e+(2n+8)\ T_m$	$3T_e+5T_m$
MPMS	$(3l-2)\ T_m$	$(3l-3)\ T_m$
S. V.	No	Yes

结束语 文中提出的多代理多签名方案中生成的最终签名可以由一组验证人合作验证其合法性。方案中每个原始签名人可以指定自己的代理群，各个代理群可以各不相同。本方案的安全性建立在计算 Diffie-Hellman 假设的基础上，通过分析可知本方案是安全和有效的。

参 考 文 献

[1] Itakura K，Nakamura K. A public-key cryptosystem suitable for digital multisignatures[J]. NEC RES DEVELOP，1983(71)：1-8

[2] Tzeng S，Yang C，Hwang M. A new multi-proxy multi-signature scheme[J]. 2002，130-138

[3] Hwang S J，Chen C C. New multi-proxy multi-signature schemes [J]. Applied Mathematics and Computation，2004，147(1)：57-67

[4] Tzeng S F，Yang C Y，Hwang M S. A nonrepudiable threshold multi-proxy multi-signature scheme with shared verification[J]. Future Generation Computer Systems，2004，20(5)：887-893

[5] Bao H Y，Cao Z F，Wang S B. Improvement on Tzeng et al.'s nonrepudiable threshold multi-proxy multi-signature scheme with shared verification[J]. Applied Mathematics and Computation，2005，169(2)：1419-1430

[6] Hsu C L，Tsai K Y，Tsai P L. Cryptanalysis and improvement of nonrepudiable threshold multi-proxy multi-signature scheme with shared verification[J]. Inf Sci，2007，177(2)：543-549

[7] Mashhadi S. Analysis of frame attack on Hsu et al.'s non-repudiable threshold multi-proxy multi-signature scheme with shared verification[C]//Scientia Iranica. 2012：1-6

[8] 霍亮，杨柳，李明祥. 基于身份的多重代理签名的安全模型[J]. 计算机科学，2012，39(6A)：41-43

[9] Kang B Y，Boyd C，Dawson E. A novel nonrepudiable threshold multi-proxy multi-signature scheme with shared verification[J]. Computers & Electrical Engineering，2009，35(1)：9-17

[10] Sun Y，Xu C X，Yu Y，et al. Improvement of a proxy multi-signature scheme without random oracles[J]. COMPUT COMMUN，2011，34(3)：257-263

[11] Liu Z H，Hu Y P，Zhang X S，et al. Provably secure multi-proxy signature scheme with revocation in the standard model [J]. COMPUT COMMUN，2011，34(3)：494-501

[12] Sun Y，Xu C X，Yu Y，et al. Improvement of a proxy multi-signature scheme without random oracles[J]. COMPUT COMMUN，2011，34(3)：257-263

[13] Waters B. Efficient identity-based encryption without random oracles[M]// Advances in Cryptology-EUROCRYPT Springer Berlin Heidelberg，2005：114-127