基于子分组的身份基多重签名方案

田陈, 王志伟

引用本文

田陈, 王志伟. 基于子分组的身份基多重签名方案[J]. 计算机科学, 2022, 49(12): 346-352.

TIAN Chen, WANG Zhi-wei. Robust Subgroup ID-based Multi-signature Scheme[J]. Computer Science, 2022, 49(12): 346-352.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.211200101

基于子分组的身份基多重签名方案

田　陈[1]　王志伟[1,2,3]
1 南京邮电大学计算机学院　南京 210023
2 先进密码技术与系统安全四川省重点实验室　成都 610225
3 江苏省大数据安全与智能处理重点实验室　南京 210023
(1020041318@njupt.edu.cn)

摘　要　目前应用于共识机制场景下的多重签名方案默认签名者为诚实实体,因此在恶意节点存在时无法保证签名安全有效。为了结合身份基密码体制与多重签名的优势,并提高多重签名在共识机制应用中对抗场景下的鲁棒性,文中提出了一种基于子分组的身份基多重签名方案。该签名方案中由不固定的随机子分组合作生成代表整个群组的多重签名,并且在签名聚合前须验证所有子分组签名的有效性。该方案生成多重签名所需的双线性对运算与子分组成员数量有关,以一定的效率为代价提升了方案的安全性;定义了基于子分组的身份基多重签名的鲁棒性,并给出了对该方案的相应证明;在随机预言模型下,证明了所提方案在适应性选择消息攻击下是不可伪造的,其安全性基于 CDH 问题的困难假设。

关键词:身份基签名;多重签名;计算 DH 问题;随机预言模型;分叉引理
中图法分类号　TP309

Robust Subgroup ID-based Multi-signature Scheme

TIAN Chen[1] and WANG Zhi-wei[1,2,3]
1 School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China
2 Advanced Cryptography and System Security Key Laboratory of Sichuan Province, Chengdu 610225, China
3 Jiangsu Key Laboratory of Big Data Security & Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing 210023, China

Abstract　The existing multi-signature scheme applied in the consensus mechanism scenario defaults that the signers are honest entities, so the security and validity of the signature could not be guaranteed when malicious nodes existed. In order to improve the robustness of multi-signature in the typical adversarial scenarios in consensus protocols, this paper proposes an ID-based multi-signature scheme based on the advantages of the ID-based cryptography system. In this signature scheme, non-fixed subgroup generates randomly cooperated to generate multi-signatures representing the entire group, and the validity of all subgroup signatures must be verified before signature aggregation. The bilinear pairings required by this scheme to generate multi-signatures are related to the number of subgroup members, which improve the security of the scheme at the cost of certain efficiency. This paper introduces a notion of robustness for robust subgroup ID-based multi-signatures, and the corresponding proof of the proposed scheme is given. Furthermore, under the random oracle model, relying on the hardness of the computational Diffie-Hellman(CDH) problem, the scheme is proved is proved to be unforgeable under adaptive selection message attack. In addition, theoretical analysis and prototype implementation of the signature scheme are carried out, and the experimental results are compared with the performance of relevant signature schemes.

Keywords　ID-based signature, Multi-signatures, Computational Diffie-Hellman(CDH) problem, Random oracle model, Forking lemma

常见的数字签名算法一般是签名者对某一消息的单次签名,而在现实应用场景中有时需要多个签名者同时签署同一消息。多重签名的概念由 Itakura 等[1]提出,指多个签名者合作产生对某一消息的签名,其签名长度与签名者数量无关。在区块链[2]等需要协同合作的应用场景中,使用多重签名算法是兼顾安全与实施效率的一种解决方案。目前,多重签名

到稿日期:2021-12-08　返修日期:2022-05-31
基金项目:先进密码技术与系统安全四川省重点实验室开放课题资助项目(SKLACSS-202114);国家自然科学基金(61672016)
This work was supported by the Open Fund of Advanced Cryptography and System Security Key Laboratory of Sichuan Province(SKLACSS-202114) and National Natural Science Foundation of China(61672016).
通信作者:王志伟(zhwwang@njupt.edu.cn)

被越来越多地应用在分布式账本等场景[3-6]中，其优点在于减少了区块的存储消耗、正确性验证时间以及已达成共识的验证时间。

在分布式计算机领域，共识机制要求在存在恶意或是故障节点的情况下，诚实节点仍然能就一致性达成共识[7]。然而，现有的非交互式多重签名方案[8-9]并不会在产生聚合签名之前进行认证或是进行有效个体签名的筛选，因此无法满足共识协议的特殊安全需求。一旦参与签名的个体中有“Byzantine 叛徒”[10]，协议的安全性就无法保证。基于子分组的多重签名[11]则是非常适合在共识协议场景下应用的一种变体，该方案允许任意合法子分组中的成员代表群产生签名，即对需要签名的消息 m 来说，真实参与签名者属于群组，但签名者子分组 J 是不确定的。2019 年，Elrond 等将 BLS 签名与 bitmap 相结合[12]，提出了基于子分组的多重签名方案，然而该方案中使用的签名无法抵御流氓密钥攻击（The Rogue Public-Key Attack），并且文中也缺乏完善的安全性分析。Pixel 签名[4]是一种基于分层身份基加密（Hierarchical Identity-based Encryption）的前向安全多重签名方案，该方案利用密钥更新的方法来提供前向安全性，但是如果用它来记录对一个区块的共识，则其安全模型只涉及 n-of-n 多重签名，而不包括 m-of-n 的多重签名。2020 年，Galindo 等给出了多重签名方案的鲁棒性定义[13]，构造了一个多重签名方案，并对其安全性进行了证明。Galindo 等的方案中使用了子分组来参与多重签名的生成，能够提供 m-of-n 的多重签名，但其子分组成员签名方案并非身份基签名方案，需要较大的密钥管理开销。

基于文献[13]，本文提出了一种基于子分组的身份基多重签名方案，改进了原方案的签名算法，由用户的身份派生公钥，使方案在应用部署时的密钥管理得到简化[14]，提高了实际的应用效率；定义并证明了方案的鲁棒性，同时在随机预言模型下，证明了方案在 CDH 困难性假设下是不可伪造的。

1　基础知识

1.1　双线性对

设 q 是大素数，G 为加法循环群，G_T 为乘法循环群，阶均为 q，其中 $g\in G$ 是生成元。假设在群 G，G_T 中离散对数问题难解，双线性映射 $e:G\times G\to G_T$ 满足以下性质：

（1）双线性性。设 $a,b\in Z_q^*$，$u,v\in G$，$e(u^a,v^b)=e(u,v^{ab})=e(u,v)^{ab}$。

（2）非退化性。存在 $u,v\in G$，使得 $e(u,v)\neq 1$。

（3）可计算性。对于所有的 $u,v\in G$，存在有效算法能够计算 $e(u,v)$。

1.2　相关困难问题

定义 1（CDH 问题）　在群 G 上，$g\in G$ 是生成元，已知 $\langle g^\alpha,g^\beta\rangle$，计算输出 $g^{\alpha\beta}$ 的值，其中 $\alpha,\beta\in Z_q^*$。

1.3　扩展的分叉引理

为了简化安全性证明，Pointcheval 等[15]提出了包含概率预言机的分叉引理（The Forking Lemma），并在文中对数字签名与盲签名方案的安全性进行了证明。Bagherzandi 等为了证明其在文献[16]中提出的基于 KV 模型的多重签名方案的安全性，将分叉引理进行了扩展，扩展的分叉引理（Generalized Forking Lemma）也能应用于其他类型签名的安全性证明，以下介绍扩展的分叉引理。

规定 $\boldsymbol{f}=(\rho,h_1,\cdots,h_{q_H})$ 是执行算法 $\mathcal{A}$ 涉及到的随机性向量，其中 ρ 是 $\mathcal{A}$ 的随机磁带，h_i 用来回答 $\mathcal{A}$ 的第 i 次哈希询问，询问哈希值的最大次数为 q_H。假设 Ω 是 $\boldsymbol{f}$ 所属的向量空间，规定 $\boldsymbol{f}'|_{j_i}=(\rho,h_1,\cdots,h_{i-1})$。此外，$in\xleftarrow{\$}\mathcal{IG}$ 是输入生成器 $\mathcal{IG}$ 输出的相关参数。算法流程 $\mathcal{A}(in,\boldsymbol{f})$ 执行完毕，输出结果 $(J,\{out_j\}_{j\in J})$，其中 J 是 $\{1,2,\cdots,q_H\}$ 的非空子集，$\{out_j\}_{j\in J}$ 是其他输出的集合。若输出的 J 为空集，则算法流程 $\mathcal{A}(in,\boldsymbol{f})$ 失败。用 p 表示算法流程执行成功，即输出合法结果的概率。扩展的分叉引理算法 $\mathcal{GF}_{\mathcal{A}}$ 如算法 1 所示。

算法 1　扩展的分叉引理算法 $\mathcal{GF}_{\mathcal{A}}$

输入：$in\xleftarrow{\$}\mathcal{IG}$，$\mathrm{f}=(\rho,\mathrm{h}_1,\cdots,\mathrm{h}_{\mathrm{q_H}})\xleftarrow{\$}\omega$

1. $(\mathrm{J},\{\mathrm{out_j}\}_{\mathrm{j}\in\mathrm{J}})\leftarrow\mathcal{A}(\mathrm{in},\mathrm{f})$ if $\mathrm{J}=\emptyset$，then output fail

2. 令 $\mathrm{J}=\{\mathrm{j}_1,\cdots,\mathrm{j}_n\}$ 且 $\mathrm{j}_1\leqslant\cdots\leqslant\mathrm{j}_n$

   for i=1，…，n do：

   $\mathrm{succ_i}\leftarrow 0$；$\mathrm{k_i}\leftarrow 0$；

   $\mathrm{k_{max}}\leftarrow 8\mathrm{nq_H}/\varepsilon\cdot\ln(8\mathrm{n}/\epsilon)$

3. 不断重复执行以下操作直到 $\mathrm{succ_i}=1$ or $\mathrm{k_i}>\mathrm{k_{max}}$：

   $\mathrm{f}''\xleftarrow{\$}\Omega$ 满足 $\mathrm{f}'|_{\mathrm{j_i}}=\mathrm{f}''|_{\mathrm{j_i}}$

   $\mathrm{f}''=(\rho,\mathrm{h}_1,\cdots,\mathrm{h}_{\mathrm{j_i}-1},\mathrm{h}''_{\mathrm{j_i}},\cdots,\mathrm{h}''_{\mathrm{q_H}})$

   $(\mathrm{J}'',\{\mathrm{out_j}''\}_{\mathrm{j}\in\mathrm{J}''})\leftarrow\mathcal{A}(\mathrm{in},\mathrm{f}'')$

4. if $\mathrm{h}''_{\mathrm{j_i}}\neq\mathrm{h}_{\mathrm{j_i}}$ and $\mathrm{J}''=\emptyset,\mathrm{j_i}\in\mathrm{J}''$

   then $\mathrm{out}'_{\mathrm{j_i}}\leftarrow\mathrm{out}''_{\mathrm{j_i}}$；$\mathrm{succ_i}\leftarrow 1$

   if i∈(1，2，…，n)时，均满足 $\mathrm{succ_i}=1$

   then output$(\mathrm{J},\{\mathrm{out_j}\}_{\mathrm{j}\in\mathrm{J}},\{\mathrm{out_j}''\}_{\mathrm{j}\in\mathrm{J}''})$

   else output fail

引理 1（扩展的分叉引理）　假设 $\mathcal{IG}$ 是随机输入生成器，q_H 是算法 $\mathcal{A}$ 哈希询问的最大次数，算法 $\mathcal{A}$ 的运行时间为 τ，并能以 ϵ 的概率执行成功。若 $q>8nq_H/\epsilon$，那么算法流程 $\mathcal{GF}_{\mathcal{A}}(in)$ 就能在不超过 $\tau\cdot 8n^2q_H/\epsilon\cdot\ln(8n/\epsilon)$ 的时间内，以不低于 $\epsilon/8$ 的概率成功输出结果。

2　方案定义及安全模型

2.1　方案定义

基于子分组的身份基多重签名方案的参与者主要有：群管理员 KGC、n 个群成员 $U=\{u_1,u_2\cdots,u_n\}$（对应的身份列表为 $ID_G=\{ID_1,ID_2\cdots,ID_n\}$）和签名的收集者 C。该方案一般包含 8 种算法：系统建立算法、公钥生成算法、私钥生成算法、群公钥生成算法、聚合公钥生成算法、群成员签名算法、聚合签名算法、多重签名验证算法。

（1）系统建立算法（Setup）：KGC 输入安全参数 k，输出系统主私钥 x、主公钥 $y=g^x$ 以及系统参数 $Params$。

（2）公钥生成算法（Set-Public-Key）：群成员 u_i 输入身份 ID_i，输出对应公钥 pk_i。

（3）私钥生成算法（Set-Private-Key）：KGC 输入群成员 u_i 的身份信息 ID_i、系统参数 $Params$ 和主私钥 x，输出群成员

u_i 的私钥 d_i，并将 d_i 发送给群成员 u_i。

(4)群公钥生成算法(Set-Group)：KGC 输入群成员身份集合 ID_G，输出固定群的唯一标识群标签 $gtag$ 和群公钥 gpk。

(5)聚合公钥生成算法(Key-Aggregate)：由签名的收集者 C 执行，输入签名者集合 J 和群成员公钥集合 $\mathcal{PK}$，输出群聚合公钥 apk。

(6)群成员签名算法(Sign)：由签名者集合中的群成员执行。输入包含签名消息 m、群标签 $gtag$、群成员 u_i 的私钥 d_i 以及系统参数 $Params$，输出各个群成员 u_i 对 m 的签名 S_i，并将 S_i 发送回签名的收集者。

(7)聚合签名算法(Combine)：由签名的收集者 C 执行。输入系统参数 $Params$、群成员公钥集合 $\mathcal{PK}$、群聚合公钥 apk 以及收到的群成员签名集合 $\{S_i\}_{i\in J}$，输出聚合后的对消息 m 的多重签名 σ。

(8)多重签名验证算法(Multi-Verify)：输入群公钥 gpk、签名者集合 J、消息 m 以及对应的多重签名 σ，若签名正确则输出“真”，否则输出“假”。

2.2 安全模型

基于子分组的身份基多重签名方案的安全性要求主要为正确性、鲁棒性和不可伪造性。

(1)正确性。按照正确的签名步骤计算出的多重签名能够通过签名验证。

(2)鲁棒性。攻击者伪装成群成员生成的不合法签名不能被聚合为多重签名。

本文设计了一个包含 3 阶段的游戏来定义多重签名的鲁棒性。

1)初始阶段(Setup)：产生系统参数 $Params$，选定挑战主公钥 y 与挑战身份 ID^*，并将 $(Params, ID^*)$ 发送给攻击者 $\mathcal{A}$。

2)签名询问阶段(Queries)：攻击者 $\mathcal{A}$ 得到预言机 $\mathcal{O}$ 的询问权，在任意群标签 $gtag$ 下，$\mathcal{A}$ 可以询问与身份 ID 对应的对任意消息 m 的签名。

3)输出阶段(Output)：攻击者 $\mathcal{A}$ 输出伪造的签名列表 $\varepsilon=\{S_i\}_{i\in J}$ 以及对应的群公钥 gpk、签名者集合 J 和消息 m^*，其中 $J\subseteq U$。

规定若攻击者 $\mathcal{A}$ 的伪造输出满足以下 3 点条件，则认为攻击者 $\mathcal{A}$ 成功破坏了鲁棒性：

1) $gpk\neq\perp$ 且 $gpk=\text{Set-Group}(ID_G)$；

2) $pk^*=pk_k$ 且 $k\in U$ 而 $k\notin J$；

3) Multi-Verify $(m^*, J, \sigma^*, gpk)=0$，攻击者 $\mathcal{A}$ 以 ID^* 为身份询问 $(m^*, gtag^*)$ 的签名得到 S^*。其中，J 是签名者集合，σ^* 是由 $\varepsilon\cup\{S^*\}$ 聚合生成的多重签名，即 $\sigma^*=\text{Combine}(m^*, \varepsilon\cup\{S^*\}, gpk)$ 且保证 $J\backslash\{k\}\neq\varnothing$。

(3)不可伪造性。攻击者伪造出一个能够通过验证的多重签名在计算上是不可行的。

本文设计了一个包含 3 阶段的游戏来定义多重签名的不可伪造性。

1)初始阶段(Setup)：产生系统参数 $Params$，选定挑战主公钥 y 与挑战身份 ID^*，并将 $(Params, ID^*)$ 发送给攻击者 $\mathcal{A}$。

2)签名询问阶段(Queries)：攻击者 $\mathcal{A}$ 得到预言机 $\mathcal{O}$ 的询问权，在任意群标签 $gtag$ 下，$\mathcal{A}$ 可以询问与身份 ID 对应的对任意消息 m 的签名。

3)输出阶段(Output)：攻击者 $\mathcal{A}$ 输出伪造的签名 σ^* 以及对应的群公钥 gpk、签名者集合 J 和消息 m^*。

规定若攻击者 $\mathcal{A}$ 的伪造输出满足以下 3 点条件，则认为攻击者 $\mathcal{A}$ 伪造成功：

1) $gpk\neq\perp$ 且 $gpk=\text{Set-Group}(ID_G)$；

2) $pk^*=pk_k$ 且 $k\in U\cap J$；

3)攻击者 $\mathcal{A}$ 不能直接询问 $(m^*, gtag^*)$ 的签名，而伪造的签名 (J, σ^*) 能被验证为有效，即 Multi-Verify $(m^*, J, \sigma^*, gpk)=1$。

若算法 $\mathcal{A}$ 的运行时间不超过 τ，询问签名的次数不超过 q_S，询问随机预言机的次数不超过 q_H 且伪造成功的概率 $p\geqslant\epsilon$，则称 $\mathcal{A}$ 是签名的 $(\tau, q_S, q_H, \epsilon)$ 伪造者。反之，如果这样的 $\mathcal{A}$ 不存在，则认为签名是 $(\tau, q_S, q_H, \epsilon)$ 不可伪造的。

3 基于子分组的身份基多重签名方案

为了提高多重签名在共识机制应用对抗场景下的鲁棒性，本文提出的方案是基于子分组设计的，允许任意合法子分组中的成员代表群产生签名，即对需要签名的消息 m 来说，真实参与签名者属于群组，但签名者子分组 J 并不是确定的。同时，为了减小对公钥证书的需要，使得密钥管理更加高效，本文基于 sakai 身份基签名[17]构建了子分组多重签名方案。

下文给出了基于子分组的身份基多重签名方案，这些方案由 8 个多项式时间算法组成。

(1)系统建立算法(Setup)。KGC 输入安全参数 k，输出系统主私钥 x、主公钥 $y=g^x$ 以及系统参数 $Params$。给定安全参数 k，群管理员 KGC 执行以下步骤：

1)通过 $Gen(k)$ 生成 (q, g, G, G_T)，其中 (G, G_T) 为素数阶 q 的双线性群对，双线性映射为 $e: G\times G\rightarrow G_T$，生成元 $g\in G$；

2)选择 $x\in Z_q^*$，计算主公钥 $y=g^x$；

3)选择安全散列函数 $H_1: \{0,1\}^*\rightarrow G$，$H_2: \{0,1\}^*\rightarrow Z_q^*$，$H_3: \{0,1\}^*\rightarrow G$，$H_4: \{0,1\}^*\rightarrow Z_q^*$，发布系统参数 $Params=\{G, G_T, e, q, g, H_1, H_2, H_3, H_4\}$，消息空间为 $m\in\{0,1\}^*$，主私钥为 x，主公钥为 $y=g^x$。

(2)公钥生成算法(Set-Public-Key)。群成员 u_i 计算 $pk_i=H_1(ID_i)$，产生对应公钥 pk_i。

(3)私钥生成算法(Set-Private-Key)。KGC 计算 $d_i=H_1(ID_i)^x$，产生群成员 u_i 的私钥 d_i，并将 d_i 发送给群成员 u_i。

(4)群公钥生成算法(Set-Group)。输入群成员身份集合 ID_G，KGC 计算 $gtag=H_4(ID_G)$ 与 $gpk=(gtag, ID_G)$，产生群标签 $gtag$ 与群公钥 gpk。

(5)聚合公钥生成算法(Key-Aggregate)。输入签名者集合 J 和群公钥集合 $\mathcal{PK}$，签名收集者 C，依次计算 $a_j=H_2(ID_j, J, ID_G)$，再计算 $apk=\prod_{j\in J} pk_j^{a_j}$，产生聚合公钥 apk。

(6)群成员签名算法(Sign)。假设签名发起人 S 发送签名消息 m 给群管理员 KGC，群管理员 KGC 确定签名者集合为 $J=\{ID_1,ID_2,\cdots,ID_n\}$，并向群成员发送$(m,J)$。每个属于签名者集合中的群成员 u_j 收到(m,J)后，执行以下具体步骤：

1)选取随机数 $r_j\in Z_q^*$，计算 $S_{2j}=g^{r_j}$；

2)计算 $H_3(gtag,m)$，再用私钥 d_j 计算 $S_{1j}=H_3(gtag,m)^{r_j}\cdot d_j$；

3)输出群成员 u_j 对 m 的签名 $S_j=(S_{1j},S_{2j})$，并将签名 S_j 发送给签名收集者 C。

(7)聚合签名算法(Combine)。签名收集者 C 接收所有参与签名的群成员的签名消息 $S_j=(S_{1j},S_{2j})$，执行以下步骤聚合签名。

1)依次验证接收到的签名消息 S_j 是否合法，若等式 $e(g,S_{1j})=e(y,pk_j)\cdot e(S_{2j},H_3(gtag,m))$成立，则签名消息 S_j 正确，否则签名消息是伪造的。若所有签名消息均合法，则执行下一步聚合，否则退出聚合签名算法，重新签名；

2)依次计算 $a_j=H_3(ID_j,J,ID_G)$；

3)计算 $\sigma_1=\prod\limits_{j\in J}S_{1j}^{a_j}$ 与 $\sigma_2=\prod\limits_{j\in J}S_{2j}^{a_j}$，产生聚合签名 $\sigma=(\sigma_1,\sigma_2)$。

(8)多重签名验证算法(Multi-Verify)。验证者输入群公钥 gpk、签名者集合 J 以及群公钥集合 $\mathcal{PK}$、多重签名 $\sigma=(\sigma_1,\sigma_2)$，执行以下验证过程。

1)调用聚合公钥生成算法，计算聚合公钥 $apk=\prod\limits_{j\in J}pk_j^{a_j}$；

2)验证等式是否成立：$e(g,\sigma_1)=e(y,apk)\cdot e(\sigma_2,H_3(gtag,m))$。

4　安全性分析

4.1　正确性

定理 1　双线性映射 $e:G\times G\to G_T$ 具有可计算性，则该基于子分组的身份基多重签名方案是正确的。

证明：假如多重签名是按照上述签名算法计算得到的，则必有以下两类等式成立。

(1)群组固定且群标签为 $gtag$ 的情况下，每个参与签名的群成员 u_i 对消息 m 的签名 $S_i=(S_{1i},S_{2i})$满足验证等式：

$$\begin{aligned}e(g,S_{1i})&=e(g,H_3(gtag,m)^{r_i}\cdot d_i)\\&=e(g,H_3(gtag,m)^{r_i})\cdot e(g,d_i)\\&=e(g^{r_i},H_3(gtag,m))\cdot e(g^x,pk_i)\\&=e(S_{2i},H_3(gtag,m))\cdot e(y,pk_i)\\&=e(y,pk_i)\cdot e(S_{2i},H_3(gtag,m))\end{aligned}$$

(2)多重签名 $\sigma=(\sigma_1,\sigma_2)$满足验证等式：

$$\begin{aligned}e(g,\sigma_1)&=e(g,\prod_{j\in J}S_{1j}^{a_j})\\&=e(g,\prod_{j\in J}H_3(gtag,m)^{a_j\cdot r_j})\cdot e(g,\prod_{j\in J}d_j^{a_j})\\&=e(g^{\sum\limits_{j\in J}a_j\cdot r_j},H_3(gtag,m))\cdot e(g,\prod_{j\in J}pk_j^{x\cdot a_j})\\&=e(\prod_{j\in J}(g^{r_j})^{a_j},H_3(gtag,m))\cdot e(g^x,\prod_{j\in J}pk_j^{a_j})\\&=e(\sigma_2,H_3(gtag,m))\cdot e(g^x,apk)\\&=e(y,apk)\cdot e(\sigma_2,H_3(gtag,m))\end{aligned}$$

4.2　鲁棒性

定理 2　基于子分组的身份基多重签名方案具有鲁棒性，并依赖于 sakai 签名的安全性。

证明：假设分布式共识场景中存在攻击者 $\mathcal{A}(\tau,q_S,q_H,\epsilon)$能够破坏多重签名方案的鲁棒性，即使用基于子分组身份基多重签名方案生成的代表共识的多重签名可能为非法签名。那么根据 2.2 节定义的鲁棒性模型，攻击者最后输出$(m^*,gpk=(gtag^*,ID_G=\{ID_i\}_{i\in U}),\varepsilon=\{S_i\}_{i\in J})$，且挑战公钥 pk^* 在公钥集合中被表示为 pk_k。若输出结果为“成功”，即方案的聚合签名算法生成了非法签名 σ^*，$e(g,\sigma_1{}^*)=e(y,apk)\cdot e(\sigma_2{}^*,H_3(gtag^*,m^*))$验证等式不成立。因此，参与聚合的 $\varepsilon=\{S_i\}_{i\in J}$ 中一定存在 $i\in J$ 使 S_i 为非法签名，则 $e(g,S_{1i})\neq e(y,pk_i)\cdot e(S_{2i},H_3((gtag^*,m^*))$验证等式不成立。分析可能得到此结果的情形：

(1)若 $i=k$，即签名 S_k 非法。由于 $S_{1k}=H_3((gtag^*,m^*)^{r_k}\cdot d^*$，$S_{2k}=g^{r_k}$ 且 $d^*=pk_k^x$，与签名 S_k 非法矛盾，因此该情形不成立。

(2)若 $i\neq k$，即签名 S_i 非法。根据方案的聚合签名算法的步骤，进行聚合前对所有签名 S_i 的正确性进行验证，与签名 S_i 非法矛盾，因此该情形不成立。

分布式共识场景下，参与共识的诚实实体与“Byzantine 叛徒”均包含在群组中，它们的公钥与用户身份关联，私钥由 KGC 参与生成。群组中任意成员，包括签名收集者 C，能够轻易验证群成员参与共识所生成的子签名的合法性。

KGC 选择任意子分组代表群组参与共识的多重签名生成，当签名收集者 C 收到子分组中成员的子签名时，首先单独验证各子签名的合法性，若含有非法签名，则终止共识，重新选择子分组生成多重签名。在保证所有子签名均合法时，再进行聚合，这样可以有效抵御试图破坏多重签名鲁棒性的“Byzantine 攻击”。

综上所述，在分布式共识场景中不存在攻击者 $\mathcal{A}(\tau,q_S,q_H,\epsilon)$能够破坏多重签名方案的鲁棒性，即方案是鲁棒的。

4.3　不可伪造性

假定 CDH 问题是困难的，我们利用扩展的分叉引理来证明该方案在适应性选择消息下的伪造攻击是安全的。

定理 3　在随机预言模型下，基于子分组的身份基多重签名方案具有不可伪造性，并依赖于 CDH 问题。

证明：$\mathcal{A}$ 是攻击者算法，$\mathcal{B}$ 是以 $\mathcal{A}$ 为子程序的另一种算法，$\mathcal{F}$ 是 CDH 问题的挑战者。H_1,H_2,H_3,H_4 是随机预言机，$\mathcal{B}$ 给定$(G,G_T,q,g,g^\alpha,g^\beta)$，其中 $G=\langle g\rangle$ 为阶为素数 q 的循环群，$\alpha,\beta\xleftarrow{\$}Z_q^*$。挑战者 $\mathcal{F}$ 的目标是利用扩展的分叉引理，运行算法 $\mathcal{GF}_{\mathcal{B}}$ 解决 CDH 问题，即计算出 $g^{\alpha\beta}$。

算法 $\mathcal{B}$ 使用 $\mathcal{A}$ 作为其子程序，设定 $y=g^\alpha$ 为挑战主公钥，则 α 为系统主私钥。$\mathcal{B}$ 设定挑战身份 ID^*，同时 $\mathcal{B}$ 需要回答 $\mathcal{A}$ 的签名与 Hash 询问，规定挑战身份 ID^* 在询问中得到的对应公钥被称为挑战公钥 pk^*。选择系统参数 $Params=\{G,G_T,e,q,g,y,H_1,H_2,H_3,H_4\}$，发送系统参数给 $\mathcal{A}$。以下定义 $\mathcal{B}$ 回答 $\mathcal{A}$ 询问的规则：

(1) $\mathcal{B}$ 回答 $\mathcal{A}$ 有关 H_2 的询问参考随机向量 $\boldsymbol{f}=(\rho, c_1, \cdots, c_{q_H})$。

(2)回答 $\boldsymbol{H}_1$:$\mathcal{B}$ 保持一个列表 $\mathcal{L}_{H_1}=\{z,c,x,h\}$,初始为 $\emptyset$。$\mathcal{A}$ 询问 z 对应的 Hash 值,若 $(z,c,x,h)\in\mathcal{L}_{H_1}$,则输出 h 作为回答;否则先确定随机值 $x\in\{0,1\}$,再选择随机数 $c\xleftarrow{\$}\mathbb{Z}_q$,若 $x=0$,则令 $h=g^c$,若 $x=1$,则令 $h=g^{\beta c}$,每次回答均更新 $\mathcal{L}_{H_1}=\mathcal{L}_{H_1}\cup\{(z,c,x,h)\}$。

(3)回答 *Extract* 查询:$\mathcal{A}$ 询问 ω 对应的私钥时,先调用 $\boldsymbol{H}_1$ 预言机查看 $\mathcal{L}_{H_1}$ 中的(z,c,x,h)。若 $x=0$,即 $h=g^c$,则返回 $d_{ID}=\omega^c$ 作为私钥;若 $x=1$,则返回$\perp$。

(4)回答 $\boldsymbol{H}_2$:$\mathcal{B}$ 保持一个列表 $\mathcal{L}_{H_2}=\{z,c\}$,初始为 $\emptyset$。$\mathcal{A}$ 第 i 次询问 z 对应的 Hash 值,若 $(z,c)\in\mathcal{L}_{H_2}$,则输出 c 作为回答;否则根据 z 的内容来决定如何回应 $\mathcal{A}$。若 $z=(ID,J,ID_G)$且$ID^*\in J$,当 $ID=ID^*$ 时,回答 $H_2(ID,J,ID_G)=c_i$;否则回答 $H_2(ID_j,J,ID_G)=d_j$,其中 $d_j\xleftarrow{\$}\mathbb{Z}_q$。若不属于以上情况,选择随机数 $d\xleftarrow{\$}\mathbb{Z}_q$ 作为回答。每次回答后更新列表 $\mathcal{L}_{H_2}\cup\{(z,c)\}$。

(5)回答 $\boldsymbol{H}_3$:$\mathcal{B}$ 保持一个列表 $\mathcal{L}_{H_3}=\{z,\lambda,H\}$,初始为 $\emptyset$。$\mathcal{A}$第 i 次询问 z 对应的 Hash 值,若$(z,\lambda,H)\in\mathcal{L}_{H_3}$,则输出 H 作为回答;否则选择随机数 $\lambda\xleftarrow{\$}\mathbb{Z}_q$,计算 $H=g^\lambda$ 作为回答;同样每次回答后更新列表 $\mathcal{L}_{H_3}\cup\{(z,\lambda,H)\}$。

(6)回答 $\boldsymbol{H}_4$:$\mathcal{B}$ 保持一个列表 $\mathcal{L}_{H_4}=\{z,v\}$,初始为 $\emptyset$。$\mathcal{A}$ 询问 z 对应的 hash 值,若$(z,v)\in\mathcal{L}_4$,则输出 v 作为回答;否则选择随机数 $v\xleftarrow{\$}\mathbb{Z}_q$ 作为回答;同样每次回答后更新列表 $\mathcal{L}_{H_4}\cup\{(z,v)\}$。

(7)回答 $sign(\cdot,\boldsymbol{sk}^*,\boldsymbol{pk}^*,\cdot)$:$\mathcal{A}$ 询问 z 对应的签名时,先调用 $\boldsymbol{H}_3$ 预言机查看 $\mathcal{L}_{H_3}$ 中的(z,λ,H)。若$(z,\lambda,H)\in\mathcal{L}_{H_3}$,则返回 $\perp$;否则根据 z 的内容来决定如何回应 $\mathcal{A}$。若$z=(ID,gtag,m)$且$ID^*\in J$,当 $ID=ID^*$ 时,返回 $\perp$;否则查找列表 $\mathcal{L}_{H_1}$,获取 ID 对应的公钥 h,选择随机数 $\delta\xleftarrow{\$}\mathbb{Z}_q$,返回 $U=y^\delta$,$V=y^\beta$,即 $S=(U,V)$作为签名,再计算$(g^\beta-h)^{-\delta}$作为 H,令 $\lambda=\perp$,并增加(z,λ,H)到列表 $\mathcal{L}_{H_3}$ 中。

最终,伪造者 $\mathcal{A}$ 会返回包含 n 个群成员的签名者集合 $J=\{ID_1,ID_2,\cdots,ID_n\}$,群成员公钥集合 $\mathcal{PK}=\{pk_1,pk_2,\cdots,pk_n\}$,伪造的签名 σ^* 以及对应的消息 m^* 与群公钥 $gpk=(gtag^*,\mathcal{PK})$。伪造者 $\mathcal{A}$不能直接询问$(m^*,gtag^*)$的签名,而伪造的签名(J,σ^*)能被验证为有效。

规定若列表 $\mathcal{L}_{H_1}=\{z,c,x,h\}$中挑战身份 ID^* 对应的 $x=0$,则终止算法 $\mathcal{B}$。由于 x 是随机选择的,因此 $\mathcal{B}$ 不终止的概率为 1/2。设 k 是 pk^* 在 $\mathcal{PK}$ 中的下标,即 $pk^*=pk_k$;j_f 是 $H_2(ID^*,J,ID_G)$ 在 $\boldsymbol{f}$ 中的下标,即 $H_2(ID^*,J,ID_G)=c_{j_f}$;$a_j=H_2(ID_j,J,ID_G)$。因此,最后 $\mathcal{B}$ 的输出表示为$(\{j_f\},\{(\sigma^*,ID_G,J,apk,\{a_j\}_{j\in J})\})$,$\mathcal{B}$ 成功输出的概率为 $\epsilon/2$。

挑战者 $\mathcal{F}$ 运行算法 $\mathcal{GF}_{\mathcal{B}}$来求解 CDH 问题,根据广义分叉引理算法设置,运行 $\mathcal{GF}_{\mathcal{B}}$的输出结果为$(\{j_f\},\{out\},\{out'\})$。前后两次运行 $\mathcal{GF}_{\mathcal{B}}$使用的随机向量 $\boldsymbol{f}$ 与 $\boldsymbol{f}'$虽不同,但仍满足 $\boldsymbol{f}|_{j_f}=\boldsymbol{f}'|_{j_f}$。输出结果中 $out=(\sigma,ID_G,J,apk,\{a_j\}_{j\in J})$而 $out'=(\sigma',ID_G',J',apk',\{a_j'\}_{j\in J'})$。具体地,$\sigma=(\sigma_1,\sigma_2)$而 $\sigma'=(\sigma_1',\sigma_2')$。

前后两次运行 $\mathcal{GF}_{\mathcal{B}}$的分叉设置为 $a_k=c_{j_f}$ 与 $a_k'=c'_{j_f}$,即 $a_k\neq a_k'$。而签名者群组是固定的,即 $ID_G=ID_G'$且 $J=J'$。因此,除 a_k 外其他 $j\in J$ 均满足 $a_j=a_j'$,根据 $apk=\prod_{j\in J}pk_j^{a_j}$ 可得 $apk/apk'=(pk^*)^{(a_k-a_k')}$。

算法 $\mathcal{GF}_{\mathcal{B}}$输出的签名 σ 与 σ'均为合法签名,因此有以下验证等式成立:

$$e(g,\sigma_1)=e(y,apk)\cdot e(\sigma_2,H_3(gtag,m))$$

$$e(g,\sigma_1')=e(y,apk')\cdot e(\sigma_2',H_3(gtag,m))$$

根据对称双线性映射性质,有:

$$\begin{aligned}e(g,\sigma_1/\sigma_1')&=e(y,apk/apk')\cdot e(\sigma_2/\sigma_2',H_3(gtag,m))\\&=e(g^\alpha,(pk^*)^{(a_k-a_k')})\cdot e(\sigma_2/\sigma_2',g^{\lambda^*})\\&=e(g^\alpha,(g^{\beta c})^{(a_k-a_k')})\cdot e(\sigma_2/\sigma_2',g^{\lambda^*})\end{aligned}$$

可得 $e(g,\sigma_1\cdot\sigma_2'/\sigma_1'\cdot\sigma_2)^{\lambda^*}=e(g,g^{\alpha\beta})^{(a_k-a_k')\cdot c}$,即$(\sigma_1\cdot\sigma_2'/\sigma_1'\cdot\sigma_2)^{\lambda^*}=g^{\alpha\beta(a_k-a_k')\cdot c}$。

最终,挑战者 $\mathcal{F}$ 能据此成功计算出 CDH 困难问题的解,即:

$$g^{\alpha\beta}=(\sigma_1\cdot\sigma_2'/\sigma_1'\cdot\sigma_2)^{-\lambda^*(a_k-a_k')\cdot c}$$

在多项式时间下 CDH 问题是困难的,会出现矛盾,且在 4.3 节中证明了在分布式共识场景下该多签名方案具有鲁棒性,因此证明中假定的伪造者 $\mathcal{A}(\tau,q_S,q_H,\varepsilon)$不存在。该基于子分组的身份基多重签名方案是不可伪造的。

5 效率分析

本节从计算开销与实验仿真两方面来评估基于子分组的身份基多重签名方案,并将结果与文献[18]中的 sakai 签名方案以及文献[13]中的 RSMSP 签名方案进行比较。

5.1 计算代价分析

表 1 列出了 3 种方案需要的各类运算以及对应的运算次数。其中,n 表示多重签名方案的子分组大小,SM 表示群乘法运算,BP 表示双线性对运算。分析表 1 中的数据,与文献[17]中的 sakai 单签名方案相比,本文方案的多重签名大大减小了签名尺寸,节约了存储消耗与签名验证的时间;但在本文方案设计的多重签名聚合过程中需要对成员签名进行验证,因此额外增加了签名的计算代价。由于使用身份基签名算法,本文方案的多重签名尺寸是 RSMSP 方案[13]的两倍,签名与多重签名验证的计算代价也略高于 RSMSP 方案;在保证与 RSMSP 方案相当的安全性的同时,本文方案以不多的额外的计算代价,简化了实际应用中的公钥密码管理。

表 1 运算量对比

Table 1 Calculation comparison

Scheme	Sign	Multi-verification	Size
Sakai Scheme	nSM	$3nBP+nSM$	$2n\lvert G_1\rvert$
RSMSP Scheme	$2nBP$	$2BP$	$\lvert G_1\rvert$
This Scheme	$3nBP+2nSM$	$3BP+SM$	$\lvert G_1\rvert$

5.2 实验仿真分析

实验仿真使用 JPBC(Java Pairting Based Cry ptography

Library)密码库[18]来实现多重签名方案并统计运行时间,最终得出的实验结果如图1—图3所示。本次实验的运行环境如下:CPU为Intel i7-7700HQ;内存为8 GB;操作系统使用Windows 10家庭版。

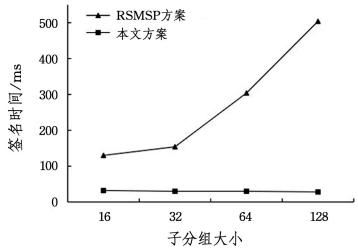

图1　签名时间开销对比

Fig. 1　Comparison of signature time overhead

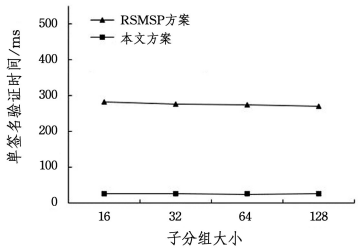

图2　单签名验证时间开销对比

Fig. 2　Comparison of time cost for single signature verification

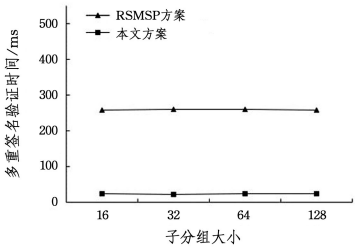

图3　多重签名验证时间开销对比

Fig. 3　Comparison of time cost for multi-signature verification

尽管本文方案的理论计算开销比RSMSP方案略高,但根据图1—图3中实验仿真的结果来看,由于本文方案使用的双线性对是对称双线性对,而RSMSP方案使用的是非对称双线性对,因此在签名与验证的时间开销方面,本文方案更优。此外,在实验仿真中,本文方案的签名时间开销固定,与子分组大小无关;而RSMSP方案的签名时间开销受到子分组大小的影响。

结束语　为了结合身份基密码体制与多重签名的优势,并提高多重签名在共识机制应用对抗场景下的鲁棒性,本文提出了基于子分组的身份基多重签名方案,该方案具有鲁棒性并且在随机预言模型下证明了它是不可伪造的。本文提出的多重签名方案以文献[17]中的sakai签名方案为基础,与sakai签名方案相比,本文方案降低了多重签名存储消耗与验证时间。与文献[13]的方案相比,尽管本文方案的多重签名尺寸增加,但在实验仿真中其效率仍具有优势。本文提出的多重签名方案所使用的身份基签名算法能够简化密钥管理,弱化对公钥证书的需求,能够为区块链共识机制应用场景[19]提供更高效安全的签名,是一个很有应用前景的方案。下一步,我们将以本文的方案为基础,试图构造基于子分组的有序多重签名[20]、基于子分组多重签名的共识协议方案[21]等,并探索方案与具体应用场景更实际的结合[22]。

参考文献

[1] ITAKURA K,NAKAMURA K. A public-key cryptosystem suitable for digital multisignatures[J]. NEC Research and Development,1983,71(71):474-480.

[2] NAKAMOTO S. Bitcoin:A Peer-to-Peer Electronic Cash System [EB/OL]. [2021-11-15]. https://bitcoin.org/bitcoin.pdf.

[3] MAXWELL G,POELSTRA A,SEURIN Y,et al. Simple Schnorr multi-signatures with applications to Bitcoin[J]. Designs Codes and Cryptography,2019,87(9):2139-2164.

[4] DRIJVERS M,GORBUNOV S,NEVEN G,et al. Pixel:Multi-signatures for Consensus[C]// 29th USENIX Security Symposium(USENIX Security 20). 2020:2093-2110.

[5] XU C D,WANG H Q. Sequential multi-signature scheme based on blockchain [J]. Journal of Nanjing University of Posts and Telecommunications (Natural Science Edition), 2021,41(2):85-94.

[6] BONEH D,DRIJVERS M,NEVEN G. Compact Multi-signatures for Smaller Blockchains[C]// International Conference on the Theory and Application of Cryptology and Information Security. Cham:Springer,2018:435-464.

[7] TAN M S,YANG J,DING L,et al. Review of Consensus Mechanism of Blockchain [J]. Computer Engineering,2020,46(12):1-11.

[8] YU H,FU S,LIU Y,et al. Certificateless Broadcast Multisignature Scheme Based on MPKC[J]. IEEE Access,2020,8:12146-12153.

[9] GABIZON A,GURKAN K,JOVANOVIC P,et al. Plumo:Towards Scalable, Interoperable Blockchains Using Ultra Light Validation Systems[C]// The 3rd ZK Proof Workshop. 2020.

[10] SHI E. Streamlined Blockchains:A Simple and Elegant Approach(A Tutorial and Survey) [C]// International Conference on the Theory and Application of Cryptology and Information Security. Cham:Springer,2019:3-17.

[11] BOLDYREVA A. Threshold Signatures , Multisignatures and Blind Signatures Based on the Gap-Diffie-Hellman-Group Signature Scheme [C]// International Workshop on Public Key Cryptography. Berlin:Springer,2003:31-46.

[12] TEAM E. Elrond:A highly scalable public blockchain via adaptive state shardingand secure proof of stake[EB/OL]. https://elrond.com/assets/files/elrond-whitepaper.pdf.

[13] GALINDO D,LIU J. Robust Subgroup Multi-Signatures for Consensus [C]// Cryptographers' Track at the RSA Conference. Cham:Springer,2022:537-561.

[14] SHAMIR A. Identity-Based Cryptosyst ems and Signature Schemes[J]. Lecture Notes in Computer Science,1985,196(1):47-53.

[15] POINTCHEVAL D,STERN J. Security arguments for digital

signatures and blind signatures[J]. Journal of Cryptology, 2000, 13(3): 361-396.
[16] BAGHERZANDI A, CHEON J H, JARECKI S, et al. Multisignatures Secure under the Discrete Logarithm Assumption and a Generalized Forking Lemma[C]// Proceedings of The 15th ACM Conference on Computer and Communications Security (CCS'08). 2008: 449-458.
[17] SAKAI R, OHGISHI K, KASAHARA M. Cryptosystems based on pairing[C]// The 2000 Symposium on Cryptography and Information Security. 2000: 354-368.
[18] DE CARO A, IOVINO V. jPBC: Java pairing based cryptography[C]// 2011 IEEE Symposium on Computers and Communications(ISCC). IEEE, 2011: 850-855.
[19] ZHAI R, CHEN X B. Research on Blockchain Consensus Mechanism[J]. Frontiers of Data & Computing, 2021, 3(3): 86-94.
[20] YANAI N, CHIDA E, MAMBO M, et al. A CDH-based Ordered Multisignature Scheme Provably Secure without Random Oracles[J]. Journal of Information Processing, 2014, 22(2): 366-375.
[21] YUAN C, XU M X, SI X M. Optimization Scheme of Consensus Algorithm Based on Aggregation Signature [J]. Computer Science, 2018, 45(2): 53-56, 83.
[22] WANG Z W. An Identity-Based Data Aggregation Protocol for the Smart Grid[J]. IEEE Transactions on Industrial Informatics, 2017, 13(5): 2428-2435.

TIAN Chen, born in 1998, postgraduate. Her main research interests include muti-signature and blockchain consensus mechanism.

WANG Zhi-wei, born in 1976, Ph. D, professor. His main research interests include applied cryptography, security and privacy in mobile and wireless systems, clouding computing and fog/edge computing.

(责任编辑:喻藜)