

带鲁棒性的子分组多重 Schnorr 签名方案

张振琦¹, 朱秋池¹, 王志伟^{1,2}

(1.南京邮电大学计算机学院, 江苏 南京 210023; 2.南京邮电大学江苏省大数据安全与智能处理重点实验室, 江苏 南京 210023)

摘要: 共识机制是区块链系统的核心技术, 但是目前的共识机制存在3个问题, 即共识效率低、可靠性和安全性低、计算复杂度高。针对这些问题, 提出了一种新的子分组多重 Schnorr 签名方案, 该方案既拥有 Schnorr 数字签名密码体制的计算复杂度低优势, 又拥有子分组多重签名的优势 (它可以从集合的全体成员中选择不定数量的成员组成子分组, 以代替群组产生多重签名, 由于子分组是不可预知的, 因此可以有效避免出现 Byzantine 叛徒, 提高了方案的安全性, 解决了共识机制存在的可靠性和安全性低、计算复杂度高的问题)。该方案引入了公共第三方 (PTP, public third party), PTP 由可自动公开执行的智能合约充当, 完全公开透明, 不仅可以抵御流氓密钥攻击, 还减少了签名过程的总通信轮次和时间开销, 解决了共识机制存在的共识效率低问题。同时, 详细证明了该方案具有鲁棒性, 可以提高共识机制的安全性; 基于离散对数假设, 该方案在随机预言模型下具有不可伪造性。理论分析和实验结果证明, 该方案拥有更小的公钥长度、私钥长度、单签名长度和多重签名长度, 拥有更少的通信轮次, 签名生成算法和验证签名算法的时间开销更小, 应用在共识机制上具有更优越的性能。

关键词: Schnorr 签名; 多重签名; 鲁棒性; 离散对数假设; 随机预言模型

中图分类号: TP303

文献标志码: A

DOI: 10.11959/j.issn.2096-109x.2024059

Robust Schnorr-based subgroup multi-signature scheme

ZHANG Zhenqi¹, ZHU Qiuchi¹, WANG Zhiwei^{1,2}

1. School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China

2. Jiangsu Key Laboratory of Big Data Security and Intelligent Processing,
Nanjing University of Posts and Telecommunications, Nanjing 210023, China

Abstract: The consensus mechanism has been considered as the core technology of blockchain systems. However, current consensus mechanisms have encountered three issues: low consensus efficiency, low reliability and security, and high computational complexity. To address these issues, a new Schnorr-based subgroup multi-signature scheme was proposed. This scheme retained the advantage of low computational complexity inherent in the Schnorr digital signature cryptosystem while incorporating the benefits of subgroup multi-signature. It allowed an indeterminate

收稿日期: 2023-12-15; 修回日期: 2024-07-28

通信作者: 王志伟, zhwwang@njupt.edu.cn

基金项目: 国家自然科学基金 (62372245)

Foundation Item: The National Natural Science Foundation of China (62372245)

引用格式: 张振琦, 朱秋池, 王志伟. 带鲁棒性的子分组多重 Schnorr 签名方案[J]. 网络与信息安全学报, 2024, 10(4): 132-142.

Citation Format: ZHANG Z Q, ZHU Q C, WANG Z W. Robust Schnorr-based subgroup multi-signature scheme[J]. Chinese Journal of Network and Information Security, 2024, 10(4): 132-142.

number of members from the entire set to form subgroups to generate multi-signatures, which replaced the group signature. The unpredictability of the subgroups effectively avoided the occurrence of Byzantine traitors, thus enhancing security and solving the problems of low reliability, security, and high computational complexity in consensus mechanisms. Additionally, a public third party was introduced, implemented by automatically and publicly executed smart contracts. It was completely open and transparent, capable of resisting the rogue public-key attack, and reduced the total number of communication rounds and time overhead in the signing process, addressing the issue of low consensus efficiency. The robustness of this scheme was proven in detail, demonstrating an improvement in the security of consensus mechanisms. Based on the discrete logarithm assumption, the scheme was shown to be unforgeable in the random oracle model. Theoretical analysis and experimental results show that the scheme possesses smaller public key length, private key length, single signature length, and multi-signature length, with fewer communication rounds and reduced time overhead in the signature generation and verification algorithms, providing superior performance when applied to consensus mechanisms.

Keywords: Schnorr signature, multi-signature, robustness, discrete logarithmic assumption, random oracle model

0 引言

共识机制是区块链技术的底层核心, 它的优劣直接影响区块链的性能。现有的共识机制存在共识效率低、可靠性和安全性低、计算复杂度高等缺点^[1]。在此背景下, 多重签名技术应运而生。该技术可以减少正确性验证时间和达成共识的验证时间^[2-4]。多重签名概念是由 Itakura^[4]于 1983 年提出来的, 它的原理是由集合中的成员充当签名者, 负责对同一个消息进行签名。与单签名相比, 多重签名的长度并没有增加, 而且多重签名的验证时间开销与签名者集合的成员数量无关。

在分布式 Oracle 领域, 要求所有参与节点都是诚实的这一假设是不合理的^[5], 因为假设存在这样一种情况: 一个不诚实的节点在其他签名者未授权或不知情的情况下, 通过观察其他签名者的公钥来选择自己的公钥, 从而达到伪造多重签名的目的。这种伪造多重签名的攻击被称为流氓密钥攻击^[6], 它使多重签名机制变得不再安全。为了解决这一问题, Micali 等^[7]提出的方案中要求所有签名者提供公钥的同时还需公开私钥知识 (knowledge of secret key, KOSK) 证明, 为此该方案需要附加一个密钥交互生成协议。这不但会使方案的实现成本变得高昂, 而且密钥交互生成协议会导致公钥本身的长度变得不可预测。Ristenpart 等^[8]对 Micali 等的方案进行了改进, 引入了一个可信第三方, 签名者的 KOSK 不再由每个签名者提供, 而是由可信第三方提供, 但是假设

每个签名者都可以信任第三方是不现实的, 并且会带来中心化的倾向。Bagherzandi 等^[9]提出了 KV (key verification) 模型, 密钥的验证隐藏在公钥中, 签名者不再单独提供公开的 KOSK, 但这会增加验证时间开销。Bellare 等^[10]在普通公钥模型的基础上首次提出了多重签名 BN (Boneh and Naor) 方案, 并在 BN 方案的基础上提出了另一种多重签名方案 MuSig^[11]。这两种方案都需要进行 3 轮交互, 这无疑会增加通信开销。Bagherzandi 等^[12]之前提出了 BCJ (Bagherzandi, Cheon and Jarecki) 方案, 利用同态陷门承诺将通信轮次减少到 2 轮, Maxwell 等受此启发提出了一个 2 轮通信的 MuSig^[11]方案。但是在 2019 年, Drijvers 等^[13]的研究表明, BCJ 和 MuSig 等现有的 2 轮多重签名方案在 k-sum 标准假设下都无法被证明是安全的。他们还提出了新的 2 轮多重签名方案 mBCJ^[13], 并证明该方案在离散对数假设条件下是安全的, 不足之处在于采用的 KV 模型增加了公钥长度。Boneh 等^[14]采用了一种独特的公钥聚合技术, 即定义聚合公钥, 聚合公钥由签名者集合成员的公钥和一个单向哈希算法生成, 提出了一个非交互式的多重签名 MSP (multi-signature protocol) 方案, 由于 MSP 使用 BLS 签名, 因此增加了计算开销, 验证效率极低。Nick^[15]等基于 OMDL (one-more discrete logarithm) 假设, 利用伪随机函数产生随机数抵御流氓密钥攻击, 提出了多重签名 MuSig2 方案。后来, Kılınc 等^[16]提出使用承诺向

量代替单一承诺以防止签名者的恶意行为的多重签名 DWMS 方案。不管是使用随机数空间还是使用承诺向量来抵御流氓密钥攻击，都会增加计算开销。

本文旨在研发一种不引入额外变量抵御流氓密钥攻击的方案。2002 年，Boldyreva 等^[17]首次提出子分组概念（即从集合的全体成员中选择不定数量的成员组成子分组，以代替群组产生多重签名），并基于此概念设计了新的多重签名方案。由于子分组是不可预知的，因此可以有效防止出现拜占庭（Byzantine）叛徒^[18-19]。但是，文献[17]没有给出完整的方案定义，也没有证明方案的安全性。Elrond 等^[20]在 BLS 签名的基础上提出了新的子分组多重签名方案，但是同样没有对方案的安全性进行详细的证明，也没有说明是否可以抵御流氓密钥攻击。Galindo 等^[21]在前人的基础上提出了新的基于 BLS 签名的子分组多重签名方案，并给出了该方案完整的安全性证明过程，同时提出了鲁棒性概念（即若单签名集合中均是被正确生成的单签名，则不存在一个敌手可以将此单签名集合聚合为一个不合法多重签名）。鲁棒性提高了多重签名方案的正确性，但是 Galindo 等的方案是基于 BLS 提出的，这增加了签名过程的计算开销。2022 年，Tian 等^[22]提出了基于子分组的身份基多重签名方案。2023 年，Wang 等^[23]提出了基于无证书的子分组多重签名方案，优化了 Tian 等方案存在的密钥托管问题。文献[22]和[23]分别展示了各自方案的鲁棒性和不可伪造性证明过程。这两种方案均存在计算复杂度高的问题。2023 年，Peng 等^[24]提出了基于 PTP（public third party）的 Schnorr 多重签名方案，但是其安全模型是为应用于 n -of- n 多重签名而设计的，并未涉及 m -of- n 多重签名。

为了解决文献[21-24]中存在的计算复杂度高等问题，本文提出了子分组多重 Schnorr 签名方案。该方案是基于 Schnorr 数字签名密码机制设计的，在计算复杂度、存储空间和达成共识的时间开销方面具有巨大的优势。同时，该方案具有子分组多重签名机制的高可靠性和安全性。此外，该方案还引入了 PTP，每个签名者只与 PTP 进行交互，而无须与其他签名者逐一交互，从而

减少了总通信轮次，提高了达成共识效率。

本文的主要贡献如下：①将子分组多重签名机制与 Schnorr 数字密码机制相结合，使方案既拥有可以抵御流氓密钥攻击的鲁棒性，又因时间复杂度低而提高了达成共识的效率；②详细论证了本方案的正确性和鲁棒性，以及基于离散对数假设，在随机预言模型下的不可伪造性；③通过理论分析和实验仿真，论证了相比现有的多重签名方案、子分组多重签名方案和 Schnorr 签名方案，本文方案在公私钥对长度、签名长度、通信轮次、签名时间开销和聚合验证时间开销等方面具有更优的性能。

1 基础知识

1.1 困难问题(离散对数假设)

定义 1 对于一个群 (G, p, g) ，定义敌手 A 的优势 Adv_G^{dl} ：

$$\text{Adv}_G^{\text{dl}} = \Pr[y = g^x; y \leftarrow \text{,} G, x \leftarrow \text{,} A(y)]$$

从上式可以看出，概率取决于两个因素，一个是 y 值从群中随机抽取，另一个是敌手 A 根据策略求解离散对数分解问题。如果敌手 A 至多在时间 τ 内使得 $\text{Adv}_G^{\text{dl}} \geq \varepsilon$ ，那么说明敌手 $A(\tau, \varepsilon)$ 攻破了离散对数假设。如果不存在这样的敌手，则离散对数假设是 (τ, ε) 困难的。

1.2 广义分叉引理

本方案的安全性证明需要用到广义分叉引理，它是 Pointcheval 等^[25]于 2000 年提出的包含随机预言机的分叉引理的扩展，这个引理的正式提出和证明是由 Bagherzandi 等^[12]完成的。

设置 $\mathcal{A}$ 是一个以 in 和 $f = (\rho, h_1, \dots, h_{q_H})$ 作为输入的算法，它在输入时与随机预言机 $H: \{0, 1\}^* \rightarrow Z_q$ 进行查询与应答交互。令 $q_H (q_H \geq 1)$ 是 $\mathcal{A}$ 询问随机预言机的最大次数。令 $\text{in} \leftarrow \$\text{IG}$ 和 $f = (\rho, h_1, \dots, h_{q_H})$ 是执行 $\mathcal{A}$ 所要输入的随机性， ρ 是随机数， h_i 是随机预言机第 i 次哈希应答。全部 f 组成一个空间 Ω ，所以有 $f|_{j_i} = (\rho, h_1, \dots, h_{i-1})$ 。定义 $\mathcal{A}(\text{in}, f)$ 是向 $\mathcal{A}$ 输入参数的步骤。待 $\mathcal{A}(\text{in}, f)$ 执行结束，若 $\mathcal{A}$ 输出

$\left(J, \{\text{out}_j\}_{j \in J}\right)$ (这里 J 是非空子集, $\{\text{out}_j\}_{j \in J}$ 是其他输出的集合), 则标志成功; 若 $\mathcal{A}$ 输出 $\emptyset$, 则标志失败。定义两个输入参数满足随机性 $f \leftarrow \$\Omega$ 和 $\text{in} \leftarrow \$\text{IG}$, ε 表示 $\mathcal{A}(\text{in}, f)$ 成功的概率。

GF_A 接受一个满足随机性的输入 in , 完整的广义分叉引理算法定义如下:

- 1) pick randomcoins ρ for $\mathcal{A}$.
- 2) $h_1, \dots, h_{q_H} \leftarrow {}_r Z_p$.
- 3) $\text{in} \leftarrow \$\text{IG}$.
- 4) $\left(J, \{\text{out}_j\}_{j \in J}\right) \leftarrow \mathcal{A}(\text{in}, f)$.
- 5) IF $J = \emptyset$ THEN output fail.
- 6) LET $J = \{j_1, \dots, j_n\}, j_1 \leq \dots \leq j_n$.
- 7) FOR $i = 1, \dots, n$ DO

$\text{succ}_i \leftarrow 0; k_i \leftarrow 0;$
 $k_{\max} \leftarrow \frac{8nq_H}{\varepsilon \ln \frac{8n}{\varepsilon}}.$
- 8) REPEAT until $\text{succ}_i = 1$ OR $k_i > k_{\max};$

$f'' \leftarrow \$\Omega$ such that $f''|_{j_i} = f|_{j_i};$
 LET $f'' = (\rho, h_1, \dots, h_{j_i-1}, h_{j_i}'', \dots, h_{q_H}'')$;
 $\left(J'', \{\text{out}_j''\}_{j \in J''}\right) \leftarrow \mathcal{A}(\text{in}, f'').$
- 9) IF $h_{j_i}'' \neq h_{j_i}$ and $J'' \neq \emptyset$ and $j_i \in J''$;
 THEN $\text{out}_{j_i}' \leftarrow \text{out}_{j_i}''; \text{succ}_i \leftarrow 1.$
- 10) IF $\text{succ}_i = 1$ for all $i = 1, \dots, n$
 Output $\left(J, \{\text{out}_j\}_{j \in J}, \{\text{out}_j'\}_{j \in J}\right);$
 ELSE output fail.

引理 1 (广义分叉引理): 运行时间在 τ 内, 对随机预言机的查询次数不超过 q_H , 最后 $\mathcal{A}$ 以 ε 的概率成功。若 $q > \frac{8nq_H}{\varepsilon}$, 并且 $\text{GF}_A(\text{in})$ 能在最大 $\tau \cdot \frac{8n^2q_H}{\varepsilon \cdot \ln\left(\frac{8n}{\varepsilon}\right)}$ 时间内, 以最小 $\frac{\varepsilon}{8}$ 的概率成功。

1.3 流氓密钥攻击

流氓密钥攻击是一种特殊的 KRA(key-replace attack)^[26]。KRA 是指签名参与方擅自修改或替换其密钥对, 由于签名者的身份由其密钥

对决定, 因此签名者的身份将存在被冒充等风险。

流氓密钥攻击则是指恶意签名者收集承诺集, 从而生成只与恶意签名者的密钥有关的聚合公钥。假设参与方中存在恶意敌手 A , 敌手 A 的密钥对是 $(x_i, y_i = g^{x_i})$, 在签名阶段, 敌手 A 收集其他参与方的承诺集 $(R_1, \dots, R_n)$ 后, 生成恶意流氓公钥 $\text{pk}_A = g^{\text{sk}_A} \cdot \left(\prod_{i \neq A} \text{pk}_i\right)^{-1}$, 于是有:

$$\begin{aligned} \text{apk} &= \prod_{i \in I} \text{pk}_i = \text{pk}_A \cdot \prod_{i \neq A} \text{pk}_i = \\ &g^{\text{sk}_A} \cdot \left(\prod_{i \neq A} \text{pk}_i\right)^{-1} \cdot \prod_{i \neq A} \text{pk}_i = g^{\text{sk}_A} \end{aligned}$$

此时, 生成的聚合签名只由敌手 A 独自の私钥提供承诺, 聚合签名退化为普通单签名。

2 方案定义与安全模型

本节首先介绍基于 Schnorr 的子分组多重签名方案的定义和安全模型; 接着介绍允许敌手参与聚合签名过程来构造鲁棒性; 最后介绍敌手输出一个伪造的 (J, σ^*) (这里 J 表示敌手 A 选择的签名者子分组的公钥集合, σ^* 是敌手 A 伪造的多重签名), 再利用 σ^* 和离散对数难题的矛盾性, 达到在这样的一个子分组模型中定义不可伪造性的目的。

2.1 方案定义

子分组多重 Schnorr 签名方案的主要参与方包括 PTP、 N 个群成员 $I = \{I_1, \dots, I_N\}$ (对应的公钥集合为 $\text{PK} = \{\text{pk}_1, \dots, \text{pk}_N\}$)、 n 个实际参与签名的子分组成员集合 $J = \{J_1, \dots, J_n\}$ ($J \subset I$)。该方案包含下列 7 种算法。

(1) Setup 算法

输入安全参数 λ , 输出系统参数 $\text{Par} = (G, p, g, H_1, H_2, H_3, H_4)$, 其中 G 是一个 p 阶的循环群, g 是该循环群的生成元。 H_1 、 H_2 、 H_3 、 H_4 是 4 个随机预言机。

(2) KeyGen 算法

签名者 I_i 生成自己的公私钥对 (x_i, y_i) , 随机选择自己的私钥 $\text{sk}_i: x_i \leftarrow {}_r Z_p^*$, 并计算自己的公

钥 $pk_i; y_i = g^{x_i}$ ，之后将公钥发送给 PTP，PTP 收集公钥集合 PK。

(3) GroupSet 算法

输入签名者的公钥集合 PK，由 PTP 计算出具有唯一性的群标签 gtag 和群公钥 gpk。

(4) AggKey 算法

PTP 由实际签名者子分组集合 J 和群成员公钥集合 PK 计算聚合公钥 apk。

(5) Sign 算法

实际签名者子分组集合 J 的成员们与 PTP 完成交互。由实际签名者承诺集合 $\{R_j\}_{j \in J}$ ，PTP 记录的时间戳哈希值 w ，消息 m 、群标签 gtag、子分组私钥集合 $\{x_j\}_{j \in J}$ 三者的哈希值，以及系统参数 Par，计算出各个子分组成员对消息 m 的签名 S_j ，并把 S_j 发送给 PTP。

(6) Combine 算法

由 PTP 将子分组签名集合 $\{S_j\}_{j \in J}$ 广播给每个实际签名者，实际签名者聚合得到聚合签名 S 。

(7) VerifyMul 算法

输入群公钥 gpk、实际签名者子分组集合 J 、消息 m 、实际签名者承诺集合 $\{R_j\}_{j \in J}$ 、时间戳哈希值 w 、聚合公钥 apk、聚合的多重签名 S 及系统参数 Par，若签名通过验证等式，则输出 1，否则输出 0。

2.2 安全模型

正确性、鲁棒性和不可伪造性三者的完整证明将定义本方案的安全性。

定义 1 (正确性): 若本方案的所有程序算法能够被正确执行，则验证等式一定成立。

定义 2 (鲁棒性): 若单签名集合中均是合法的单签名，则方案的聚合程序无法生成不合法多重签名。

鲁棒性的完整证明过程如下。

(1) SetUp 算法

挑战者选择系统参数 Par 和公私钥对 (sk^*, pk^*) ，并将 (Par, pk^*) 发送给敌手 A 。

(2) Queries 算法

对于任意的 gtag, m ，敌手 A 访问随机预言机可以得到关于 gtag, m 的签名。

(3) OutPut 算法

挑战者接收到敌手 A 输出的群公钥 gpk、消息 m^* 、公钥集合 $PK = \{pk_1, \dots, pk_N\}$ 、实际签名者子分组集合 $J \subset I$ 、实际签名者签名集合 $\{S_j\}_{j \in J}$ 和伪造的聚合签名 σ^* 。

如果敌手 A 的输出列表满足以下 3 条规则，则认为敌手 A 成功破坏了方案的鲁棒性。

(1) $gpk \neq \perp$ 且 $gpk = \text{GroupSet}(PK)$ 。

(2) $pk^* = pk_k$ ， $k \in I$ 且 $k \notin J$ 。

(3) $\text{VerifyMul}(m^*, J, \sigma^*, gpk) = 0$ ，敌手 A 以 pk^* 的身份查询 $(m^*, gtag^*)$ 的签名获得 S^* ， J 是实际签名者的集合， σ^* 是由 $\{S_j\}_{j \in J} \cup S^*$ 生成的聚合签名，其中需要确保 $J/\{k\} \neq \emptyset$ 。

如果敌手 A 在最大运行时间 τ 内、最多 q_s 次签名查询和最多 q_H 次哈希查询的前提下，使得多重签名验证算法输出 0 的概率不小于 ϵ ，可以说算法 $\mathcal{A}(\tau, q_s, q_H, \epsilon)$ 在单签名集合中均是合法的单签名，成功做到了 Combine 算法输出的是不合法多重签名，否则本方案是 $(\tau, q_s, q_H, \epsilon)$ 鲁棒性的。

定义 3 (不可伪造性): 敌手 A 无法同时满足方案全部的程序算法正确运行和输出一个验证等式不通过的签名。

不可伪造性的完整证明如下。

(1) SetUp 算法

挑战者选择系统参数 Par 和公私钥对 (sk^*, pk^*) ，并将 (Par, pk^*) 发送给敌手 A 。

(2) Queries 算法

对于任意的 gtag, m ，敌手 A 访问随机预言机可以得到关于 gtag, m 的签名。

(3) OutPut 算法

挑战者接收到敌手 A 输出的群公钥 gpk、消息 m^* 、公钥集合 $PK = \{pk_1, \dots, pk_N\}$ 、实际签名者子分组集合 $J \subset I$ 、实际签名者签名集合 $\{S_j\}_{j \in J}$ 和伪造的聚合签名 σ^* 。

如果敌手 A 的输出列表满足下列 3 条规则，

则认为敌手 A 成功破坏了方案的不可伪造性。

1) $\text{gpk} \neq \perp$ 且 $\text{gpk} = \text{GroupSet}(\text{PK})$ 。

2) $\text{pk}^* = \text{pk}_k$, $k \in I$ 且 $k \notin J$ 。

3) $\text{VerifyMul}(m^*, J, \sigma^*, \text{gpk}) = 1$, 敌手 A 不能直接询问 (m^*, gtag^*) 的签名 S^* , J 是实际签名者子分组集合, σ^* 是由 $\{S_j\}_{j \in J} \cup S^*$ 生成的聚合签名。

如果敌手 A 在最大运行时间 τ 内、最多 q_S 次签名查询和最多 q_H 次哈希查询的前提下, 使多重签名验证算法输出 1 的概率不小于 ε , 可以说算法 $\mathcal{A}(\tau, q_S, q_H, \varepsilon)$ 在满足此方案的全部程序正确执行的情况下, 使 VerifyMul 算法的输出是无法通过的, 否则, 本方案是 $(\tau, q_S, q_H, \varepsilon)$ 不可伪造性的。

3 子分组多重 Schnorr 签名方案的构造

子分组多重 Schnorr 签名方案是基于 Schnorr 签名^[27]设计的, 因此具有较低的计算成本。同时, 为了提高多重签名的安全性, 该方案引入了子分组多重签名机制^[21], 即在消息 m 、实际签名者子分组集合 J 和大群组 I 的基础上, 定义了 J 是不可预知、任意的, 且 $J \subset I$ 。此外, 为了抵御流氓密钥攻击, 该方案引入了 PTP^[24], PTP 由第三方或智能合约担任。不同于 TTP (trusted third party), PTP 的特点是公开、可验证和自动执行。

总体来说, 子分组多重 Schnorr 签名方案由下列 7 个算法构造而成。

(1) Setup 算法

输入安全参数 λ ; 令 G 是一个 p 阶的循环群, g 是循环群的生成元; 定义以下 4 个哈希函数:

$$H_1: \{0,1\}^* \rightarrow Z_p^*$$

$$H_2: \{0,1\}^* \rightarrow Z_p^*$$

$$H_3: \{0,1\}^* \rightarrow Z_p^*$$

$$H_4: \{0,1\}^* \rightarrow Z_p^*$$

定义 I 表示整个群组, J 表示实际签名者子分组集合, $J \subset I$ 。最终输出的系统参数 $\text{Par} = (G, p, g, H_1, H_2, H_3, H_4)$ 。

(2) KeyGen 算法

签名者 I_i 生成自己的公私钥对 (x_i, y_i) , 随机选择自己的私钥 $sk_i: x_i \leftarrow_r Z_p^*$, 并计算自己的公钥 $pk_i: y_i = g^{x_i}$, 之后将公钥发送给 PTP, PTP 收集公钥集合 $\text{PK} = \{pk_i\}_{i \in I}$ 。

(3) GroupSet 算法

输入公钥集合 $\text{PK} = \{pk_i\}_{i \in I}$, PTP 根据哈希函数 H_1 求出唯一群标签 $\text{gtag} = H_1(\text{PK})$ 和群公钥 $\text{gpk} = (\text{gtag}, \text{PK})$ 。

(4) AggKey 算法

PTP 利用实际签名者子分组集合 J 和群成员公钥集合 PK 计算 $a_j = H_2(y_j, J, \text{PK})_{j \in J}$ 和 $\text{apk} = \prod_{j \in J} y_j^{a_j}$, 再计算出聚合公钥 apk 。

(5) Sign 算法

签名发起人确定实际签名者子分组集合 J , 签名发起人将消息 m 和子分组集合 $J = \{y_1, \dots, y_n\}$ 发送给 PTP。每个属于子分组 J 的成员将执行以下步骤:

①选取随机数 $r_j \in Z_p^*$, 计算承诺 $R_j = g^{r_j}$ 。

②将承诺 R_j 发送给 PTP, PTP 广播收集到的承诺集 $R = \{R_1, \dots, R_n\}$ 。

③PTP 记录时间戳 t , 首先计算 $w = H_3(t)$ 和 $W = g^w$, 然后广播 (w, W) 。

④实际签名者收集到承诺集 R 和 (w, W) 后, 计算 $U = W^n \cdot \prod_{j=1}^n R_j$ 和 $c = H_4(\text{gtag}, U, m)$, 再计算单签名 $S_j = w + r_j + ca_j x_j \bmod p$, 并将单签名发送给 PTP。

(6) Combine 算法

PTP 收集到所有单签名 $\{S_j\}_{j \in J}$ 后便验证等式 $g^{S_j} = W \cdot R_j \cdot y_j^{a_j c}$, 若此等式成立, 则签名 S_j 合法, 否则中断签名。若所有签名均合法, 则计算 $S = \sum_{j=1}^n S_j$ 。

(7) VerifyMul 算法

输入 $\text{gpk} = (\text{gtag}, \text{PK})$ 、实际签名者子分组集合 J 和聚合签名 S , 然后执行以下步骤:

① AggKey 是由 PTP 计算的, 所以 $\text{apk} =$

$\prod_{j \in J} y_j^{a_j}$ 和 $a_j = H_2(y_i, J, PK)_{j \in J}$ 是公开的。

② 计算 $U = W^n \cdot \prod_{j \in J} R_j$, $c = H_4(\text{gtag}, U, m)$ 。

③ 验证等式 $g^S = U \cdot \text{apk}^c$, 若等式成立, 则签名合法, 输出 1; 否则输出 0, 签名非法, 多重签名过程中止。

4 安全性分析

4.1 正确性

下面证明, 如果方案全部的程序算法都能够正确执行, 那么单签名验证等式和多重签名验证等式均成立。

1) 在 S_j 和 gtag 均确定的情况下, 一定有签名者对消息 m 的单签名 S_j 满足等式:

$$g^{S_j} = g^{w + r_j + ca_j x_j} = g^w \cdot g^{r_j} \cdot g^{ca_j x_j} = W \cdot R_j \cdot y_j^{ca_j}$$

2) 在 S 和 gtag 均确定的情况下, 以下等式一定成立:

$$\begin{aligned} g^S &= g^{\sum_{j \in J} S_j} = g^{nw + \sum_{j \in J} r_j + c \sum_{j \in J} a_j x_j} \\ &= W^n \cdot \prod_{j \in J} R_j \cdot \left(\prod_{j \in J} g^{a_j x_j} \right)^c \\ &= U \cdot \text{apk}^c \end{aligned}$$

4.2 鲁棒性

定理 1 本方案得到的签名具有鲁棒性。

证明 本方案全部的程序算法都能够正确运行, 表示产生的单签名都合法。假设存在敌手 $A(\tau, q_S, q_H, \varepsilon)$ 能够破坏本方案的鲁棒性, 那么表示敌手 A 会输出一个不合法的多重签名, 即敌手 A 最后输出一个元组 $(m^*, \{y_i\}_{i \in I}, \text{gpk} = (\text{gtag}^*, \text{PK} =$

$\{y_i\}_{i \in I}, \{S_j\}_{j \in J})$, 这里挑战公钥在公钥集合 PK 中被表示为 pk_k 。若该方案成功输出了多重签名 S^* , 则输出的多重签名为不合法签名, 那么等式 $g^{S^*} = U \cdot \text{apk}^c$ 不成立。因此, 单签名集合 $\{S_j\}_{j \in J}$ 中一定存在一个不合法签名 S_i , 使单签名验证等式 $g^{S_j} = W \cdot R_j \cdot y_j^{ca_j}$ 不成立。此时将出现以下两种情况:

1) 若 $i = k$, 即 pk_k 对应的单签名 S_k 非法, 但是 $S_k = w + r_k + ca_k x_k$, $a_k = H_2(y_k, J, \text{PK})_{k \in J}$,

这与签名 S_k 非法矛盾, 因此这种情况不成立。

2) 若 $i \neq k$, 即 S_i 非法, 但该方案全部的程序算法都能够正确执行的情况下, 在执行 **Combine** 算法之前一定会有单签名验证等式通过, 因此这种情况不成立。

可知, 在子分组多重 Schnorr 签名的生成过程中, 无论是参与签名的诚实实体还是 Byzantine 叛徒均在群组中, 由自己随机选择公钥和密钥, 并要求将公钥公开, 而无论是公钥还是承诺、时间戳的哈希值都存储在 PTP 上, 且是完全公开透明的, 因此, 无论是群组的节点还是 PTP, 均可以使用单签名验证等式以保证其他节点签名的合法性。签名发起者发起签名后, PTP 会从大群组中随机选择子分组执行方案剩余的程序算法。子分组成员在完成单签名生成算法后会将自己的单签名广播给 PTP, 这时 PTP 会利用单签名验证等式验证签名的合法性, 这一步将减轻每个签名节点的计算负担, 所有签名的等式验证都通过, 方案才会继续执行, 否则就会中断签名过程, PTP 将返回 **AggKey** 算法并重新开始执行此方案。只有所有的子分组成员单签名均通过验证, 才开始运行 **Combine** 算法, 以有效保证方案的鲁棒性。

综上所述, 不存在敌手 $A(\tau, q_S, q_H, \varepsilon)$ 能够破坏本方案的鲁棒性, 即本方案是鲁棒性的。

4.3 不可伪造性

定理 2 在随机预言机模型下, 由本方案得到的签名具有不可伪造性, 这是依赖于离散对数困难假设得出的。

证明 令 F 是离散对数的挑战者, A 负责攻击本方案, B 是以 A 为基础的一个算法。定义 H_1, H_2, H_3, H_4 为随机预言机。 B 随机选择 (G, p, g) , 其中 G 是以大素数 q 为阶、 g 为生成元的循环群。

B 随机选择 $(\text{pk}^*, \text{sk}^*)$, 并将 pk^* 和 Par 公开, 同时对 A 的哈希值和签名请求进行应答。 F 的目标是解决离散对数困难问题, 即计算出等式 $\tilde{X} = g^{\tilde{x}}$ 中的幂 $\tilde{x}$, 这一步需要利用 A 与 B 之间的交互和敌手输出的签名, 再合理地使用广义分叉引理, 最终得到离散对数困难问题的解。接下来,

模拟 B 和 A 的请求与应答过程。

1) B 应答 A 关于 H_2 的查询请求, 输出随机向量 $f = (\rho, c_1, \dots, c_{q_H})$ 。

2) H_1 应答: B 维护列表 $L_{H_1} = \{z, v\}$, 初始化为 $\emptyset$ 。 A 向 B 查询 z 对应的哈希值, 如果有 $(z, v) \in L_{H_1}$, 那么 B 将 v 应答给 A ; 否则选择随机数 $v \leftarrow {}^s Z_q^*$ 应答给 A 。每次应答后更新列表 $L_{H_1} \cup \{(z, v)\}$ 。

3) H_2 应答: B 维护列表 $L_{H_2} = \{z, c\}$, 初始化为 $\emptyset$ 。 A 第 i 次向 B 查询 z 对应的哈希值, 如果 $(z, c) \in L_{H_2}$, 那么 B 将 c 应答给 A 。若 $z = (y, J, PK)$ 且 $y^* \in J$, 当 $y^* = y$ 时, B 将 $H_2(y, J, PK) = c_i$ 应答给 A ; 否则, B 选择随机数 $d \leftarrow {}^s Z_q^*$ 应答给 A 。每次应答结束后更新列表 $L_{H_2} \cup \{(z, c)\}$ 。

4) H_4 应答: B 维护一个列表 $L_{H_4} = \{z, \lambda, h\}$, 初始化为 $\emptyset$ 。 A 第 i 次向 B 查询 z 对应的哈希值, 若 $(z, \lambda, h) \in L_{H_4}$, 那么 B 将 h 应答给 A ; 否则, B 选择随机数 $\lambda \leftarrow {}^s Z_q^*$, 然后利用等式 $h = g^\lambda$ 求出 h , 将 h 应答给 A 。每次应答结束后更新列表 $L_{H_4} \cup \{(z, \lambda, h)\}$ 。

5) 签名应答: A 向 B 查询 z 对应的签名时, 若 $(z, \lambda, h) \in L_{H_4}$, 则返回 $\perp$ 。否则, 若 $z = (y, \text{gtag}, m)$ 且 $y^* \in J$, 当 $y = y^*$ 时, 返回 $\perp$; 若 B 分析 $PK = \{y_1, \dots, y_k, \dots, y_N\}$, 这里的 pk^* 表示为 pk_k , 即 y_k , 如果 $H_2(y_k, J, PK)$ 没有被定义, B 就会对哈希表 H_2 进行内部查询, 对 $\{y_j\}_{j \in J}$ 的每个 H_2 查询, 均以 a_j 作为应答, 计算 $\tilde{y} = \prod_{j \in J} y_j^{a_j}$ 。

最终, A 会输出 $J = \{y_1, \dots, y_n\}$ 、 $PK = \{pk_1, pk_2, \dots, pk_N\}$ 、单签名 S^* 、多重签名 σ^* , 消息 m^* 、 $\text{gpk} = (\text{gtag}^*, PK)$ 。规定 A 不能询问 (m^*, gtag^*) 对应的签名, 要求伪造的 (J, σ^*) 能通过 VerifyMul 算法。

令 k 是 pk^* 在 PK 里的下标, 有 $pk^* = pk_k$ 。 j_f 是 $H_2(y^*, J, PK)$ 在 f 中的下标, 即 $H_2(y^*, J, PK) = c_{j_f}, a_{j_f} = H_2(y_j, J, PK)$ 。因此 B 最终输出

$$\left(\{j_f\}, \left(\sigma^*, PK, J, \text{apk}, \{a_j\}_{j \in J} \right) \right)。$$

F 运行 GF_B 算法来求解离散对数困难问题, 运行 GF_B 算法输出的结果为 $(\{j_f\}, \{\text{out}\}, \{\text{out}'\})$ 。两次运行 GF_B 算法满足 $f \neq f', f|_{j_f} = f'|_{j_f}$ 。其中 out, out' 可以表示为

$$\begin{aligned} \text{out} &= (\sigma, PK, J, \text{apk}, \{a_j\}_{j \in J}) \\ \text{out}' &= (\sigma', PK', J', \text{apk}', \{a_j\}_{j \in J}) \end{aligned}$$

GF_B 算法的分叉点在 $a_k = c_{j_f}$ 和 $a'_k = c'_{j_f}$, 即 $a_k \neq a'_k$ 。而签名者群组 I 是不变的, 即 $PK = PK'$ 、 $J = J'$ 。所以除 a_k 外, 其他均满足 $a_k = a'_k$ 。

GF_B 算法输出的签名 σ, σ' 均合法, 因此 VerifyMul 算法输出为 1, 有:

$$\begin{aligned} g^\sigma &= U \cdot \tilde{X}^c \\ g^{\sigma'} &= U \cdot \tilde{X}'^c \end{aligned}$$

将两个多重签名相减:

$$\sigma - \sigma' = \sum_{j \in J} S_j - \sum_{j \in J} S'_j = S_k - S'_k = c(a_k - a'_k)x_k$$

因此存在:

$$x_k = \frac{\sigma - \sigma'}{c(a_k - a'_k)}$$

又因为 $\tilde{X}$ 是聚合公钥, 存在 $\tilde{X} = \prod_{j \in J} y_j^{a_j}$ 、 $\tilde{X}' = g^{\tilde{x}}$, 因此 $\tilde{x} = \sum_{j \in J} x_j a_j$ 。结合上面对 x_k 的计算, 可

$$\text{以得出 } \tilde{x} = \sum_{j \in J} x_j a_j + \frac{a_k(\sigma - \sigma')}{c(a_k - a'_k)}。$$

最终 F 能够据此成功计算出 DL 困难问题的

$$\text{解, 即 } \tilde{X} = g^{\tilde{x}}, \tilde{x} = \sum_{j \in J} x_j a_j + \frac{a_k(\sigma - \sigma')}{c(a_k - a'_k)}。$$

但这与在多项式时间算法下离散对数问题是困难的产生矛盾, 所以根据本方案完整的程序算法得到的签名是不可伪造的。

5 性能分析

通过理论分析和实验仿真获得了本方案的私钥长度、公钥长度、签名长度、通信轮次、时间开销成本, 下面将对本文方案与 mBCJ^[13]、MuSig2^[15]、DWMS^[16]、MEMS^[24]、RSMSP^[21] 方案的性能。

5.1 理论分析

表 1 从私钥长度、公钥长度、签名长度、通信轮次方面将本文方案与 mBCJ^[13]、MuSig2^[15]、DWMS^[16]、MEMS^[24]、RSMSP^[21] 方案进行了对比。其中, $|G|$ 和 $|Z_p|$ 分别表示乘法群和整数群元素的大小。由表 1 可知, 本文方案与 mBCJ、MuSig2、DWMS、MEMS 和 RSMSP 都具有相同且最优的私钥长度 $|Z_p|$; 除了 mBCJ 方案, 其他方案都具有相同且最优的公钥长度 $|G|$; 4 个基于 Schnorr 的两轮多重签名方案, 除了 mBCJ 方案, 其他方案的签名长度都是 $|G| + |Z_p|$, 基于子分组的 BLS 多重签名方案 RSMSP 和本文方案的签名长度是最优的, 分别是 $|G|$ 和 $|Z_p|$, 但是本文方案是基于 Schnorr 签名的, 具有更低的计算复杂度; 本文方案引入了 PTP, 因此在通信轮次方面也是最优的。

表 1 不同方案的性能对比

Table 1 Performance comparison of different schemes

方案	私钥长度	公钥长度	签名长度	通信轮次
mBCJ	$ Z_p $	$2 G + Z_p $	$2 G + 3 Z_p $	$4n(n-1)$
MuSig2	$ Z_p $	$ G $	$ G + Z_p $	$4n(n-1)$
DWMS	$ Z_p $	$ G $	$ G + Z_p $	$4n(n-1)$
MEMS	$ Z_p $	$ G $	$ G + Z_p $	$5n$
RSMSP	$ Z_p $	$ G $	$ G $	$2n(n-1)$
本文	$ Z_p $	$ G $	$ Z_p $	$5n$

本文方案与 mBCJ、MuSig2、DWMS、MEMS 和 RSMSP 方案的时间开销和安全性对比如表 2 所示, 表中的 $k \exp$ 表示 k -幂运算, 与指数运算相比, 加法、乘法和哈希运算的开销可以忽略不计。除了 mBCJ 方案, 其他方案的所有签名都是采用 PPK^[10] 模型抵御流氓密钥攻击, 因此聚合算法开销的幂运算次数都是 n , 而 mBCJ 方案采用的是 KV^[12, 13, 28] 模型, 聚合算法通过乘法实现, 时间开销可以忽略不计, 但代价是增加了公钥长度; 在签名算法开销方面, 本文方案是基于 Schnorr 实现的, 只需要 1 次幂运算, 因此是最优的; 在验证算法开销方面, 本文方案与大多数方案一样, 幂运算次数都是 2; 在安全性方面, 本文方案能够抵御流氓密钥攻击, 并基于 DL 问题在 ROM 模型下给出了鲁棒性和不可伪造性证明。

表 2 时间开销与安全性对比

Table 2 Comparison of time cost and security

方案	签名算法 开销	验证算 法开销	聚合算 法开销	鲁棒性	困难 问题
mBCJ	5exp	6exp	0	否	DL
MuSig2	$(2V-1) \exp$	2exp	$n \exp$	否	OMDL
DWMS	$(mn+m) \exp$	2exp	$n \exp$	否	OMDL
MEMS	1exp	2exp	$n \exp$	否	DL
RSMSP	$n \exp$	2exp	$n \exp$	是	CDH
本文	1exp	2exp	$n \exp$	是	DL

5.2 实验分析

本文方案的仿真实验采用 Java 编程语言和 JPBC 密码库。实验的运行环境如下: CPU 为 AMD Ryzen7 6800H 3.20 GHz, 内存为 16 GB, 操作系统是 Windows 11 家庭版。实验比较了 6 种方案的签名算法时间开销和验证签名算法时间开销。

从图 1 可以看出, 在签名算法的时间开销方面, 本文方案不仅优于基于 BLS 的子分组多重签名 RSMSP 方案, 还优于同样基于 Schnorr 的多重签名 mBCJ、MuSig2、DWMS 和 MEMS 4 种方案。基于 BLS 的子分组多重签名 RSMSP 方案使用了较多的双线性运算, 所以签名算法的时间开销相对较大。而 MuSig2 和 DWMS 虽然都是基于 Schnorr 的多重签名方案, 但是由于引入了随机性数量, 因此签名算法的时间开销同样较大。

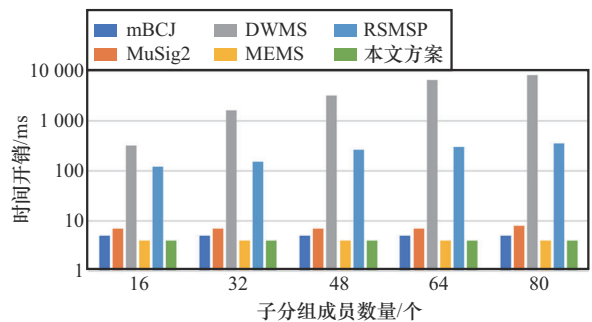


图 1 签名算法的时间开销对比

Figure 1 Comparison of time cost of signature algorithms

从图 2 可以看出, 在验证签名算法的时间开销方面, 本文方案也是 6 种方案中最优的。RSMSP 方案需要进行较多的双线性运算, 而本

文方案采用基于 Schnorr 的签名算法, 只需进行线性计算, 因此验证签名算法的时间开销相对较小。同样基于 Schnorr 的 mBCJ 方案因采用 KV 模型来抵御流氓密钥攻击而增加了公钥长度, 因此验证签名算法的时间开销也较大。

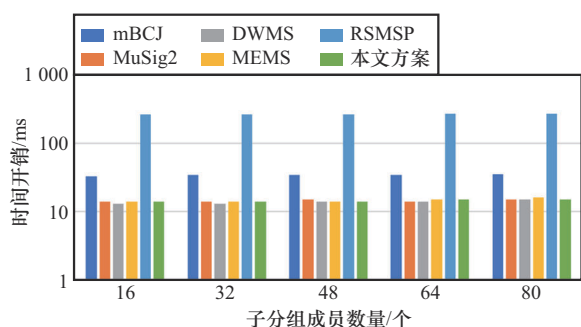


图2 验证签名算法的时间开销对比

Figure 2 Comparison of time cost of validation signature algorithms

本文方案和 MEMS 方案都是基于 Schnorr 的, 从图 1 和图 2 可以看出两者的时间开销几乎相同, 为了深入分析两者的时间消耗差异, 实验扩大了群组 and 子分组的规模, 当整个群组的成员达到 10 000 个时, 从图 3 可以看出 MEMS 方案的时间开销几乎不变, 这是因为 MEMS 是多重签名, 与子分组成员数量无关; 而本文方案的时间开销取决于子分组成员数量, 并且本文方案无论是签名时间开销还是验证时间开销, 都优于 MEMS 方案。

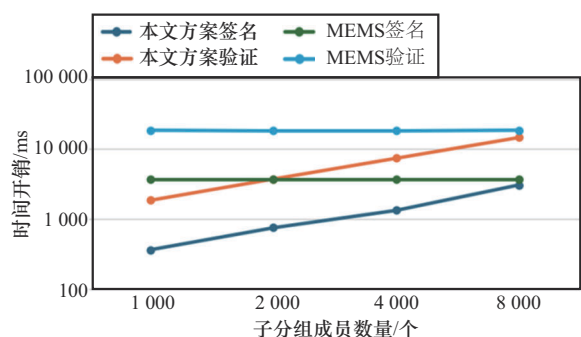


图3 本文方案与 MEMS 方案的时间开销对比

Figure 3 Comparison of time cost between this scheme and MEMS

此外, 本文还统计了 PTP 运行时间开销, 统计结果如图 4 所示, 可以看出 PTP 运行时间开销与子分组成员数量有关, 这是因为 PTP 的计算工作集中在 Oracle 和 VerifyMul 算法两个环节, 而这两个环节的计算工作均需要子分组全体成员参与。

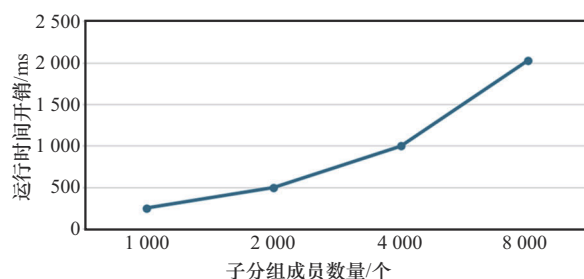


图4 PTP 运行时间开销

Figure 4 PTP runtime cost

6 结束语

本文方案是以文献[21]中的 RSMSP 方案和文献[24]中的 MEMS 方案为基础, 结合子分组多重签名机制设计的。本文方案相比基于 Schnorr 的 mBCJ、MuSig2、DWMS、MEMS 和基于 BLS 的 RSMSP 方案, 不仅拥有更小的公私钥长度和签名长度、更少的通信轮次、更小的签名生成算法和签名验证算法时间开销。本文方案的低计算复杂度和时间开销、可抵御流氓密钥攻击的特性, 能够为分布式 Oracle 提供更加高效、安全、低成本的签名, 具有广阔的应用前景。

在未来的研究工作中, 一方面将对本文方案的性能进行优化, 另一方面将在本方案的基础上, 尝试构造可以高效、安全应用于分布式 Oracle 的基于 Schnorr 的门限签名。

参考文献:

- [1] 旋逸昭, 赵红武, 金瑜. 一种基于双链的区块链共识机制[J]. 计算机工程, 2024, 50(5): 139-148.
XUAN Y Z, ZHAO H W, JIN Y. A dual-chain-based consensus mechanism for blockchain[J]. Computer Engineering, 2024, 50(5): 139-148.
- [2] BONEH D, DRIJVERS M, NEVEN G. Compact multi-signatures for smaller blockchains[C]//Proceedings of International Conference on the Theory and Application of Cryptology and Information Security. Cham: Springer, 2018: 435-464.
- [3] DRIJVERS M, GORBUNOV S, NEVEN G, et al. Pixel: multi-signatures for consensus[C]//Proceedings of the 29th USENIX Security Symposium (USENIX Security 20), Boston: MA, 2020: 2093-2110.
- [4] ITAKURA K, NAKAMURA K. A public-key cryptosystem suitable for digital multisignatures[J]. NEC Research & Development, 1983, 71(3): 1-8.
- [5] 沈学利, 李欣儒. 基于节点动态评分机制的分组共识算法[J]. 计算机应用研究, 2024, 41(4): 989-994.
SHEN X L, LI X R. Group consensus algorithm based on node dynamic scoring mechanism[J]. Application Research of Computers, 2024, 41(4): 989-994.

- [6] LI C M, HWANG T, LEE N Y. Threshold-multisignature schemes where suspected forgery implies traceability of adversarial shareholders[C]//Advances in Cryptology — EUROCRYPT'94. Berlin, Heidelberg: Springer, 1995: 194-204.
- [7] MICALI S, OHTA K, REYZIN L. Accountable-subgroup multisignatures: extended abstract[C]//Proceedings of the 8th ACM Conference on Computer and Communications Security. New York: ACM, 2001: 245-254.
- [8] RISTENPART T, YILEK S. The power of proofs-of-possession: securing multiparty signatures against rogue-key attacks[C]//Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin, Heidelberg: Springer, 2007: 228-245.
- [9] BAGHERZANDI A, JARECKI S. Multisignatures using proofs of secret key possession, as secure as the diffie-Hellman problem[C]//Proceedings of International Conference on Security and Cryptography for Networks. Berlin, Heidelberg: Springer, 2008: 218-235.
- [10] BELLARE M, NEVEN G. Multi-signatures in the plain public-Key model and a general forking lemma[C]//Proceedings of the 13th ACM Conference on Computer and Communications Security. New York: ACM, 2006: 390-399.
- [11] 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.
- [12] BAGHERZANDI A, CHEON J H, JARECKI S. Multisignatures secure under the discrete logarithm assumption and a generalized forking lemma[C]//Proceedings of the 15th ACM Conference on Computer and Communications Security. New York: ACM, 2008: 449-458.
- [13] DRIJVERS M, EDALATNEJAD K, FORD B, et al. On the security of two-round multi-signatures[C]//Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP). Piscataway: IEEE Press, 2019: 1084-1101.
- [14] BONEH D, LYNN B, SHACHAM H. Short signatures from the Weil pairing[C]//Proceedings of International Conference on the Theory and Application of Cryptology and Information Security. Berlin, Heidelberg: Springer, 2001: 514-532.
- [15] NICK J, RUFFING T, SEURIN Y. MuSig2: simple two-round schnorr multi-signatures[C]//Annual International Cryptology Conference. Berlin, Heidelberg: Springer, 2021: 189-221.
- [16] KILINÇ ALPER H, BURDGES J. Two-round trip schnorr multi-signatures via delinearized witnesses[C]//Advances in Cryptology — CRYPTO 2021. Berlin, Heidelberg: Springer, 2021: 157-188.
- [17] BOLDYREVA A. Threshold signatures, multisignatures and blind signatures based on the gap-diffie-Hellman-group signature scheme [C]//Public Key Cryptography — PKC 2003. Berlin, Heidelberg: Springer, 2002: 31-46.
- [18] 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. Berlin, Heidelberg: Springer, 2019: 3-17.
- [19] ZHU X X. Research on blockchain consensus mechanism and implementation[J]. IOP Conference Series: Materials Science and Engineering, 2019, 569(4): 042058.
- [20] ELROND A. Highly scalable public blockchain via adaptive state sharding and secure proof of stake [Z]. Whitepaper. 2019.
- [21] GALINDO D, LIU J. Robust subgroup multi-signatures for consensus[C]//Proceedings of CryptographersTrack at the RSA Conference. Berlin, Heidelberg: Springer, 2022: 537-561.
- [22] 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 C, WANG Z W. Robust subgroup ID-based multi-signature scheme[J]. Computer Science, 2022, 49(12): 346-352.
- [23] 王宇航, 徐哲清, 王志伟, 等. 基于无证书的子分组多重签名方案[J]. 信息安全学报, 2023: 2-10.
- WANG Y H, XU Z Q, WANG Z W, et al. A certificateless subgroup multi-signature scheme [J]. Journal of Cyber Security, 2023: 2-10.
- [24] ZHANG P, GE F, LIU Y H. Achieving maximum efficiency in schnorr-based multi-signature and applications in blockchain[J/OL]. ArXiv e-Prints, 2023: arXiv: 2305.13699.
- [25] POINTCHEVAL D, STERN J. Security arguments for digital signatures and blind signatures[J]. Journal of Cryptology, 2000, 13(3): 361-396.
- [26] 刘罕. 基于区块链的物联网信息安全关键技术研究与应用[D]. 上海: 上海海事大学, 2022.
- LIU H. Research and application on blockchain based key technology in iot information security[D]. Shanghai: Shanghai Maritime University, 2022.
- [27] NEVEN G, SMART N P, WARINSCHI B. Hash function requirements for schnorr signatures[J]. Journal of Mathematical Cryptology, 2009, 3(1): 69-87.
- [28] SYTA E, TAMAS I, VISHER D, et al. Keeping authorities "honest or bust "with decentralized witness cosigning[C]//Proceedings of the 2016 IEEE Symposium on Security and Privacy. Piscataway: IEEE Press, 2016: 526-545.

[作者简介]



张振琦 (2000-), 男, 河南商丘人, 南京邮电大学硕士生, 主要研究方向为密码协议。



朱秋池 (2000-), 女, 江苏盐城人, 南京邮电大学硕士生, 主要研究方向为应用密码学。



王志伟 (1976-), 男, 南京邮电大学教授, 主要研究方向为云/雾计算安全、区块链、密码协议。