计 算 机 研 究 与 发 展
Journal of Computer Research and Development
https://crad.ict.ac.cn
63(1): 227−242, 2026

面向区块链的 UC 安全门限 SM9 签名方案

董佶圣[1,3,4] 李 聪[2,3,4] 沈子楠[1,4] 胡达齐[1,4] 沈晴霓[1,3,4] 吴中海[1,2,3,4]

[1](北京大学软件与微电子学院 北京 102600)

[2](北京大学计算机学院 北京 100871)

[3](北京大学软件工程国家工程研究中心 北京 100871)

[4](北京大学-八分量区块链与隐私计算联合实验室(北京大学) 北京 100871)

(dongjisheng@pku.edu.cn)

Threshold SM9 Signature Scheme with UC Security for Blockchain

Dong Jisheng[1,3,4], Li Cong[2,3,4], Shen Zinan[1,4], Hu Daqi[1,4], Shen Qingni[1,3,4], and Wu Zhonghai[1,2,3,4]

[1](*School of Software and Microelectronics*, *Peking University*, *Beijing* 102600)

[2](*School of Computer Science*, *Peking University*, *Beijing* 100871)

[3](*National Engineering Research Center for Software Engineering*, *Peking University*, *Beijing* 100871)

[4](*Peking University-Octa Innovations Laboratory for Blockchain and Privacy Computing* (*Peking University*), *Beijing* 100871)

Abstract Blockchain has received a lot of attention in recent years, and the digital signature scheme is a legitimate way to authenticate transactions in the blockchain. The threshold signature scheme enables the key controlled by a single participant to be managed by multiple participants at the same time, and the transaction can be signed only if more than a threshold number of participants sign at the same time, which expands the application mode and application scenario of cryptocurrency, and reduces the risk of single point of failure. However, existing threshold signature schemes are mostly limited to traditional public key signature algorithms such as ECDSA (elliptic curve digital signature algorithm), which incur significant overhead in public key certificate storage and management. IBC (identity-based cryptosystem) can effectively address this issue, but there is no corresponding research on threshold SM9 signature schemes. Implementing a threshold signature scheme usually requires the use of a two-party secret sharing conversion protocol. To address the low efficiency of the existing two-party secret sharing conversion protocol based on the Paillier encryption algorithm, we propose a two-party secret sharing conversion protocol based on the ElGamal encryption algorithm, and apply it to construct a threshold signature scheme based on SM9. In addition, in order to prove the security of the scheme, an identity-based threshold signature ideal function is constructed, and the security of the scheme is proved under the UC (universally composable) model. Theoretical analysis and experimental results show that the threshold SM9 signature scheme based on the ElGamal algorithm is more efficient than the existing schemes. In particular, when the threshold is 100, the speed is about 6.1 times that of the Paillier-based scheme. The threshold signature scheme is tested on three different blockchain platforms, Ethereum, Hyperledger Fabric and ChainMaker on Tencent blockchain as a service. The experimental results show that the scheme is feasible in the blockchain scenario.

Key words digital signature; threshold signature; SM9; UC security; blockchain

收稿日期: 2024-10-31; **修回日期**: 2025-07-02

基金项目: 新疆两区科技发展计划项目(2024LQ03004); 国家重点研发计划项目(2022YFB2703301)

This work was supported by the Science and Technology Development Program of Two Districts in Xinjiang (2024LQ03004) and the National Key Research and Development Program of China (2022YFB2703301).

通信作者: 沈晴霓(qingnishen@ss.pku.edu.cn)

摘 要 近年来,区块链受到公众广泛关注,数字签名方案是区块链中交易的合法鉴权方式。门限签名方案可以使得由单一参与方控制的密钥同时被多个参与方管理,只有在超过阈值的多个参与方同时进行签名的情况下才能完成交易签名,扩展了加密数字货币的应用方式和应用场景,降低了单点故障的风险。然而,目前的门限签名方案多局限于ECDSA(elliptic curve digital signature algorithm)等传统公钥签名算法,由此带来的公钥证书存储和管理开销较大。基于标识的密码体制可以有效解决这一问题,但还没有相应的针对基于国密SM9标识密码的门限签名方案研究。实现门限签名方案通常需要两方秘密分享转换方案,针对现有基于Paillier加密算法的两方秘密分享转换方案效率低的问题,提出了一种基于ElGamal加密算法的两方秘密分享转换方案,并应用其构建了一个基于国密SM9的门限签名方案。为证明方案的安全性,构建了基于标识的门限签名理想函数,并在通用可组合(universally composable, UC)模型下证明了该方案的安全性。理论分析和实验结果表明,与现有方案相比,提出的门限签名方案在基于ElGamal算法的秘密分享转换方案下效率更高。特别地,当门限值为100时,速度约为基于Paillier方案的6.1倍。在以太坊、Hyperledger Fabric和腾讯云服务平台长安链体验网络3个不同的区块链平台上对该门限签名方案进行了测试,实验结果表明方案在区块链场景中具有可行性。

关键词 数字签名;门限签名;SM9;UC安全;区块链

中图法分类号 TP311

DOI: 10.7544/issn1000-1239.202440830 **CSTR:** 32373.14.issn1000-1239.202440830

近年来,随着比特币[1]等数字货币的发展,区块链技术得到了广泛关注,它提供了一种在不可信网络中进行信息与价值传递交换的可信通道。区块链凭借着独有的信任建立机制,成为了金融和科技深度融合的重要方向,并引领着新一轮技术和产业的变革浪潮,如医疗保健[2]、智慧城市[3]、物联网[4]、金融风控[5]等。在区块链交易过程中,每笔上链的交易由发起方的数字签名授权,用于签名的私钥一旦泄露可能导致交易被伪造,因此适当的密钥存储方式对区块链的安全性至关重要。以区块链为核心技术的主流加密货币如比特币、以太坊等,大多使用ECDSA(elliptic curve digital signature algorithm)签名方案作为单一签名方案,在密钥存储方面存在诸多潜在安全问题。例如,对个人而言,加密货币钱包的存储设备一经攻破会导致未经授权的访问或财产被盗;对组织而言,内部任一节点发生单点故障可能导致整个系统的瘫痪。

门限签名方案是解决上述问题的一种有效途径,(t,n)-门限签名方案允许n个参与方(或服务器)共享在单个公钥下发布数字签名的权力,各参与方(或服务器)指定一个阈值t,使得任何包含t个参与方的子集都可以共同签名,但任何更小的子集都不能完成签名。门限签名的目标是生成与现有的签名方案兼容的签名,其中,密钥生成和签名过程被双方之间的通信协议所取代,但生成的签名与集中式方案兼容,因此验证算法不变。如图1所示,使用(t,n)-门限签名方案,区块链签名密钥的控制权可以分布在n个参与方(或服务器)之间,需要至少其中的t个参与方(或服务器)来协作生成签名。至关重要的是,即使这些参与方(或服务器)中多达$t-1$个遭到攻击,合法签名仍不可伪造,钱包中的资金也仍将保持安全。

目前针对于门限签名方案的设计主要集中于ECDSA签名的门限生成,但ECDSA等方案依赖传统公钥密码体制,需要进行公钥证书存储和管理。而基于标识的密码体制(identity-based cryptosystem, IBC)无需证书管理工作,可直接使用标识信息字符串(例如邮箱地址、电话号码)作为公钥,有效降低了用户的存储和管理开销。1985年Shamir[6]提出基于标识的密码体制,2001年Boneh等人[7]应用椭圆曲线上的双线性映射技术,提出了一个实用且可证明安全的IBC加密方案。2016年国家密码管理局正式发布了我国自主设计的国密SM9标识密码算法标准[8],其中的数字签名算法于2018年成为了ISO/IEC国际标准[9]。

近年来,标识密码算法在区块链上的应用和研究开始受到关注。Wan等人[10]提出了一种分层区块链系统,使用分层标识密码方案实现物联网设备和数据的可扩展和问责管理;Sharma等人[11]使用基于标识的加密算法改进医疗保健行业的数据共享过程,使用户能够控制包括存储在云环境中的敏感信息;Yin等人[12]提出了一种基于区块链的新型分布式身份,旨在建立个体主权身份并保护敏感属性、链上身

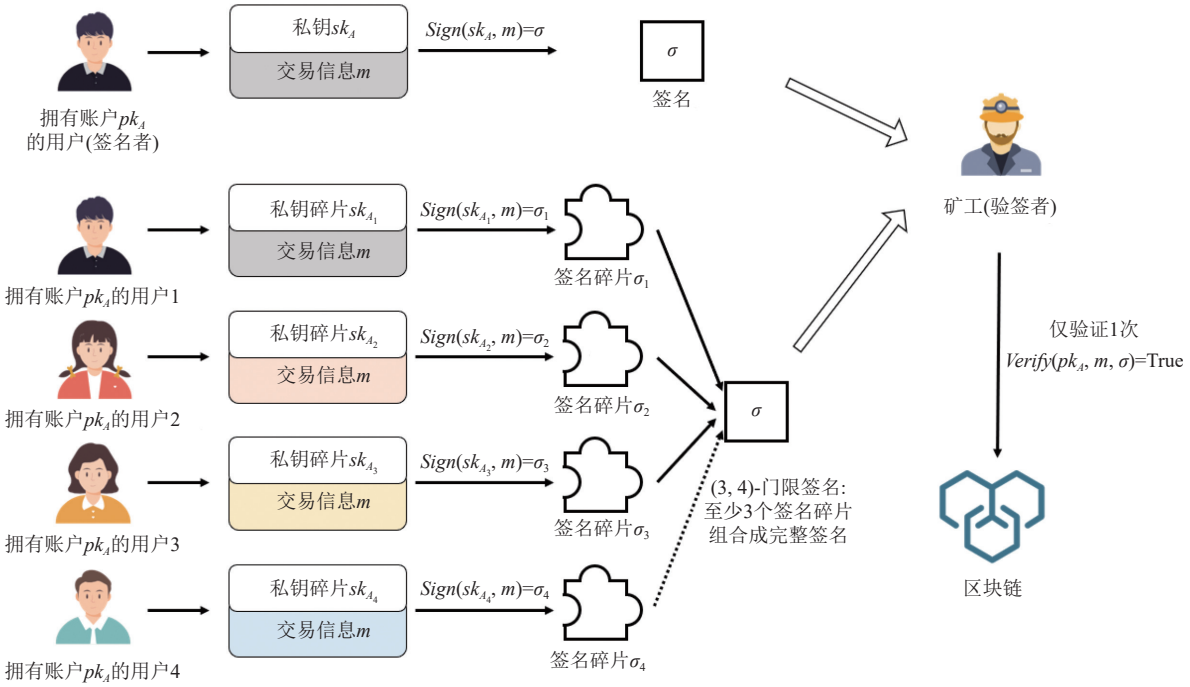


Fig. 1　Application example of threshold signature scheme on blockchain

图 1　区块链门限签名方案应用示例

份数据和凭证链接的隐私。此外，长安链将 SM9 标识密码算法应用于区块链系统[13]，杨亚涛等人[14]提出了一种基于 SM9 标识密码算法的可证明安全区块链隐私保护方案；安浩杨等人[15]提出了一种基于 SM9 数字签名的环签名方案，并将其应用至区块链隐私保护中。由于用户需要根据密钥生成中心（key generate center，KGC）获取自身密钥，因此基于 SM9 的标识密码区块链方案多应用于联盟链和私有链等许可链场景中。然而，在标识密码方案中如何实现门限签名，以规避单一签名带来的安全风险仍是一个亟待解决的问题，目前尚未有支持门限的 SM9 签名方案。针对上述问题，本文具体贡献有 3 个方面：

1）提出了基于 ElGamal 加密方案的两方秘密分享转换协议 SMtAwc（scalar-multiplicative-to-additive with check），相比于基于 Paillier 加密方案的两方秘密分享转换协议 MtAwc（multiplicative-to-additive with check），该协议可以更高效地匹配门限 SM9 签名方案，这是由于 ElGamal 加密方案与 SM9 签名方案所用参数一致，可以省略由 Paillier 加密方案带来的昂贵的零知识范围证明。

2）提出了门限 SM9 签名方案，并在通用可组合（universally composable，UC）安全框架内对门限 SM9 签名协议进行安全分析。为此，构建了一个基于标识的门限签名理想函数，并证明了在恶意敌手存在的情况下，设计的协议可以安全地实现该理想函数。这种 UC 安全协议可以保证在作为更复杂的系统构建模块使用的情况下仍然安全，此属性在去中心化和高度安全敏感的分布式系统（如区块链加密货币系统）中尤为关键。

3）本文从理论和实验角度对门限 SM9 签名方案进行了详细分析，与现有的两方 SM9 签名方案的扩展版本相比，门限 SM9 签名方案在计算和存储性能上均有较为明显的提升。特别地，当门限值为 20 时，门限 SM9 签名方案速度约为对比方案的 7.0 倍；当门限值为 100 时，门限 SM9 签名方案速度约为对比方案的 6.1 倍。

1　相关工作

1991 年，Desmedt 等人[16]为 RSA 算法设计了门限签名方案，此后，针对不同签名算法的门限方案也相继被提出。ECDSA 签名算法的门限方案需要在不泄露秘密值a和b的前提下计算ab的加法共享，因此其门限方案的构造更加困难。1996 年，Gennnaro 等人[17]使用 Shamir 秘密分享方案[18]分别对 2 个秘密值进行分享，但这需要$2t-1$个参与方才能构建签名。2016 年，Gennaro 等人[19]引入了 Paillier 加密方案，利用其加法同态性质构造了门限最优的 ECDSA 门限签名方案。但为确保安全性，在该方案的密钥生成阶段各参与方需要分布式地生成 Paillier 加密密钥，这

导致该方案不具有实用性。2018 年, Gennaro 等人[20]和 Lindell 等人[21]分别对该方案进行了改进, 构建基于 Paillier 的 MtA(multiplicative-to-additive)两方秘密分享转换协议, 避免了文献 [19] 中需要分布式生成 Paillier 加密密钥的问题。2020 年, Canetti 等人[22]在文献 [20−21] 的基础上进行了优化, 降低了方案的通信轮数, 并在通用可组合框架下证明了其安全性, 表明该方案具备通用可组合性质。然而, 由于 Paillier 的消息空间和 ECDSA 的签名空间不一致, 上述方案中基于 Pailliar 的 MtA 协议均需引入相对昂贵的零知识范围证明。此外, 2021 年 Tymokhanov 等人[23]指出, 文献 [20] 中的 MtA 协议缺少对 Paillier 加密密钥 N 的范围检查, 这使得攻击者可以利用 MtA 协议提取出诚实参与方的密钥。此外, Castagnos 等人[24-25]在文献 [20] 的基础上, 通过将 MtA 协议中的 Paillier 加密替换为同样具有加法同态性质的 CL 加密设计了新的门限方案, 由于 CL 的消息空间和 ECDSA 的签名空间一致, 因此不需要进行零知识范围证明。然而 CL 加密算法本身计算复杂度高, 因此基于 CL 的 MtA 协议会带来较大计算开销。Doerner 等人[26-27]基于不经意传输(oblivious transfer, OT)协议构造的 MtA 协议在计算效率方面实现了明显提升, 但因可能受到比特级的错误攻击和不一致输入的替换攻击, 从而需要额外的一致性检查, 导致需要至少 6 轮交互, 显著增加了带宽需求, 相对于上述 MtA 协议而言实用性较差。

Baek 等人[28]提出了基于标识的门限签名方案, 并在随机谕言机模型下证明了选择消息攻击的不可伪造性; Xiong 等人[29]提出了一个多 KGC 的基于标识的门限签名方案, 并在标准模型下同样证明了选择消息攻击的不可伪造性; Atapoor[30]分别侧重效率和鲁棒性并提出了 2 个基于标识的门限签名方案, 但该方案基于后量子算法, 实用性较差; Jiang 等人[31]提出了一个完全分布式的方案以抵抗恶意 KGC 和拒绝服务攻击。然而, 上述方案的构造均不是基于国际标准的主流标识密码算法, 且均未实现 UC 安全性。

SM9 签名算法与 ECDSA 签名算法略有不同, 但通过对私钥的秘密分享, 可以避免对 SM9 中椭圆曲线上点的 Paillier 加密。基于上述观察, 2020 年, Mu 等人[32]在 Lindell[33]设计的两方 ECDSA 签名方案的基础上提出了基于 SM9 签名算法的两方签名方案。然而, 到目前为止, 基于 SM9 的门限签名方案尚未有进一步发展, 有效的多方门限 SM9 签名方案尚未被提出。

2 基础知识

本节主要介绍相关的数学基础知识。

2.1 双线性映射

双线性映射基于椭圆曲线上的非对称乘法循环群, 设 q 为大素数, $(G_1,+)$和$(G_2,+)$是阶为 q 的加法循环群, $(G_T,\cdot)$是阶为 q 的乘法循环群。P_1是$(G_1,+)$的生成元, P_2是$(G_2,+)$的生成元, 存在$(G_2,+)$到$(G_1,+)$的同态映射 ψ, 使得 $\psi(P_2)=P_1$。双线性映射 e 是从 $(G_1,+)\times(G_2,+)$到$(G_T,\cdot)$的映射, 且满足 3 个性质:

1)双线性。对任意 $P\in G_1, Q\in G_2, a,b\in\mathbb{Z}_q$, 有 $e(aP,bQ)=e(P,Q)^{ab}$。

2)非退化性。存在$P\in G_1, Q\in G_2$, 使得$e(P,Q)\neq 1_{G_T}$, 其中 1_{G_T} 是群 G_T 的单位元。

3)可计算性。对任意 $P\in G_1, Q\in G_2$, 存在有效的算法来计算 $e(P,Q)$。

2.2 SM9 数字签名算法

SM9 标识密码算法的主要内容包括数字签名算法、密钥交换协议、密钥封装机制和公钥加密算法。设 λ 为安全参数, q 是与 λ 相关的大素数, $\mathcal{BP}=(G_1,G_2,G_T,e,q)$是根据 λ 生成的非对称双线性群, P_1,P_2 分别为群 G_1 和 G_2 的生成元, $H_1,H_2:(0,1)^*\times\mathbb{Z}_q\to\mathbb{Z}_q$ 是 2 个密码学哈希函数。下面给出 SM9 数字签名算法的简单描述:

1)系统建立 $Setup$。输入安全参数 λ, 密钥生成中心选择非对称双线性群 $\mathcal{BP}$, 并选择随机数 $msk=\alpha\in\mathbb{Z}_q$ 作为签名主私钥, 计算 $P_{\text{pub}}=\alpha P_2\in G_2$ 和 G_T 中的元素 $g=e(P_1,P_{\text{pub}})$。选择用 1B 表示的私钥生成函数识别符 hid, 公开主公钥 $mpk=(P_{\text{pub}},g)$和系统公共参数 $pp=\{\mathcal{BP},P_1,P_2,H_1,H_2,hid\}$。

2)密钥生成 $KeyGen$。用户标识 ID 对应的私钥为 $sk=\left(\dfrac{\alpha}{\alpha+H_1(ID\|hid,q)}\bmod q\right)P_1$。为便于描述, 下文使用符号 $H_1(ID)$ 代替 $H_1(ID\|hid,q)$。

3)签名生成 $Sign$。设待签名消息为 M, 签名者选取随机数 $r\in\mathbb{Z}_q$, 计算 $w=g^r, h=H_2(M\|w,q), l=(r-h)\bmod q$ 以及 $S=l\times sk$。签名者将 (h,S) 作为签名值。

4)签名验证 $Verify$。签名者接收到消息 m 和签名值 (h,S) 后, 计算 $P=H_1(ID)P_2+P_{\text{pub}}, u=e(S,P)$ 以及 $w=u\times g^h$, 验证等式 $h=H_2(M\|w,q)$ 是否成立, 若成立, 则验证通过, 否则验证不通过。

2.3 非延展的陷门承诺

陷门承诺方案允许承诺方以保证信息论隐私的

方式对消息进行承诺，这意味着给定对某一消息的承诺，具有无限计算能力的接收方无法比随机猜测更好地获得已承诺的消息。此外，陷门承诺方案引入一个计算困难的陷门，并允许拥有该陷门信息的用户以任何方式打开给定的承诺。陷门承诺方案由4个算法组成：

1）密钥生成$KeyGen$。该算法为随机性算法，它接收安全参数λ作为输入，输出承诺公钥pk和对应的陷门tk。

2）承诺$Commit$。该算法为随机性算法，它接收公钥pk、待承诺消息M和承诺阶段所使用的随机数R作为输入，输出$C(M)$为对消息M的承诺，$D(M)$为其对应的打开字符串。

3）验证$Verify$。该算法为确定性算法，它接收公钥pk及C,D作为输入，输出消息M或$\perp$。

4）替换$Equivocate$。该算法为随机性算法，它接收公钥pk、字符串M和R及$[C,D]=Commit(pk,M,R)$、消息$M'\neq M$、另一个字符串T作为输入，若$T=tk$，则算法输出D'，满足$Verify(pk,C,D')=M'$。

陷门承诺方案应当满足正确性、信息论安全性和安全绑定性。对这3个性质的简要介绍如下：

1）正确性。若$[C(M),D(M)]=Commit(pk,M,R)$，则$Verify(pk,C(M),D(M))=M$。

2）信息论安全性。对于任意消息对M和M'，$C(M)$和$C(M')$的分布是统计不可区分的。

3）安全绑定性。对于任意敌手$\mathcal{A}$，输出C,D,D'，满足$Verify(pk,C,D)=M,Verify(pk,C,D')=M'$且$M\neq M'$的概率是可忽略的。

上述承诺方案也被称为非延展的，如果对任意敌手$\mathcal{A}$，给定对消息M的承诺C，产生另一个承诺C'，并在随后收到C对应的打开字符串D后将C'打开为一个与M相关的消息M'的概率是可忽略的。在实际应用中，可以使用任意安全哈希函数H，定义$h=H(M,R)$为对消息M的承诺，其中R为随机选择的字符串，并假设H表现为随机谕示机。本文在实验中使用了这种承诺方式。

2.4 ElGamal加密算法

ElGamal加密算法是一个基于Diffie-Hellman密钥交换的非对称加密算法。因其与SM9签名形式相契合，且具有加法同态性质，同时相较于其他基于椭圆曲线的加法同态加密算法效率更高，故本文选择ElGamal加密算法用于方案构造。设λ是安全参数，q是与λ相关的大素数，G是阶为q的加法循环群，P是群G的生成元。下面给出ElGamal加密算法的简单描述：

1）密钥生成$KeyGen$。选择随机数$x\in\mathbb{Z}_q$作为其私钥，并计算$X=xP$作为公钥。

2）加密$EGEnc$。设待加密消息为m，解密者公钥为X。加密者将m映射为G上的一个元素M，选择随机数$r\in\mathbb{Z}_q$，计算密文$EGEnc_X(M)=(C_1,C_2)$，其中$C_1=M+rX,C_2=rP$。

3）解密$EGDec$。已知密文(C_1,C_2)及解密者私钥x。解密者计算$S=xC_2,M'=C_1-S$，并将M'映射回明文m。

加法同态性质描述为：给定2组密文$EGEnc_X(M_1)=\left(C_1^1,C_2^1\right)$，$EGEnc_X(M_2)=\left(C_1^2,C_2^2\right)$及$k\in\mathbb{Z}_q$，定义运算$+_{\mathrm{E}}$：$G^2\times G^2\to G^2$和$\times_{\mathrm{E}}:\mathbb{Z}_q\times G^2\to G^2$：

$$EGEnc_X(M_1)+_{\mathrm{E}}EGEnc_X(M_2)=\left(C_1^1+C_1^2,C_2^1+C_2^2\right),$$

$$k\times_{\mathrm{E}}EGEnc_X(M_1)=\left(kC_1^1,kC_2^1\right),$$

则有$EGEnc_X(M_1)+_{\mathrm{E}}EGEnc_X(M_2)=EGEnc_X(M_1+M_2)$和$k\times_{\mathrm{E}}EGEnc_X(M_1)=EGEnc_X(kM_1)$成立。

2.5 MtAwc协议

MtA（multiplicative-to-additive）协议是一种安全的两方秘密分享转换协议，能够将两方持有的乘法秘密分享安全地转换为加法秘密分享。具体而言，假设Alice和Bob分别持有秘密$\sigma\in\mathbb{Z}_q^*$的乘法分享份额$r_1,r_2\in\mathbb{Z}_q$，满足$\sigma=r_1r_2 \bmod q$。Alice和Bob分别将r_1和r_2作为协议的输入，并分别得到输出$\gamma_1\in\mathbb{Z}_q$和$\gamma_2\in\mathbb{Z}_q$，满足$\gamma_1+\gamma_2=r_1r_2=\sigma \bmod q$。MtAwc协议是对MtA协议的改进，其中Alice公开关于r_1的额外输入$w=g^{r_1}$，并向Bob发送额外证明以保证协议输入r_1与w的一致性。现有的MtA协议基于同态加密或不经意传输构造，其中基于Paillier同态加密的方案因其相对更好的综合性能而应用最为广泛。假设Bob持有Pallier加密公钥X_2，其中$N<q^8$且Alice接受对N的范围证明，下面给出基于Paillier加密的MtAwc协议构造：

1）Bob执行

①使用自身公钥X_2计算$C_2=Enc_{X_2}(r_2)$；

②计算关于关系$\{C_2:\exists r_2 \text{ s.t.} C_2=Enc_{X_2}(r_2)\wedge r_2<q^3\}$的零知识证明$\pi_2$；(Pf.1)

③发送(C_2,π_2)给Alice。

2）Alice执行

①验证π_2，若验证不通过则中止协议；

②随机选择$\gamma_1'\in\mathbb{Z}_{q^5}$，并计算$C_1=(r_1\times_{\mathrm{E}}C_2)+_{\mathrm{E}}Enc_{X_2}\left(\gamma_1'\right)=Enc_{X_2}\left(r_1r_2+\gamma_1'\right)$；

③计算关于关系$\{(C_1,C_2,w):\exists\left(r_1,\gamma_1'\right) \text{ s.t. } w=$

$g^{r_1} \wedge C_1 = (r_1 \times_{\mathrm{E}} C_2) +_{\mathrm{E}} Enc_{X_2}(\gamma_1') \wedge r_1 < q^3 \wedge \gamma_1' < q^7\}$的零知识证明$\pi_1$; (Pf.2)

④发送(C_1, π_1)给 Bob, 并输出$\gamma_1 = -\gamma_1' \bmod q$。

3)Bob 执行

①验证π_1, 若验证不通过则中止协议;

②计算并输出$\gamma_2 = Dec_{x_2}(C_1)$。

值得注意的是, 由于 Paillier 加密的消息空间远大于协议的输入/输出空间, 协议中需要引入昂贵的范围证明以防止恶意敌手通过恶意构造输入而造成诚实方的隐私信息泄露, 这使得 MtAwc 协议的开销显著增加。

2.6 通用可组合模型

通用可组合模型是一种通过现实/理想范式(real/ideal paradigm)来证明安全性的方法, 是一种用于定义密码协议安全性的一般模型, 这种模型用以描述任意协议在并发执行下的安全性。它的核心包括环境、协议(参与方)和敌手 3 部分, 通过把协议实际运行中可能出现的所有情况抽象为环境、参与方和敌手之间的交互, 其中环境的视角包括所有通信和敌手的外部行为(包括腐化参与方); 敌手控制参与方之间的通信, 并可以腐化参与方。一个协议是 UC 安全的, 是指对于任意环境都有一个与攻击协议的敌手实现相同目标的理想敌手(称为模拟器), 使得环境无法区分是在与敌手(在现实世界中)以及各参与方进行交互还是在与模拟器(在理想世界中)进行交互。如果没有环境能够区分真实进程和理想进程之间的差异, 那么即使该协议与一些其他复杂的系统相组合, 协议也是安全的。有关 UC 模型的更多介绍详见文献 [34]。

在真实进程中, 考虑一个由$n+2$台交互式图灵机组成的系统, 其中n台图灵机代表n个参与方在协议Π中的执行, 1台图灵机$\mathcal{A}$代表敌手, 1台图灵机$\mathcal{Z}$代表环境, 所有机器都在相同的安全参数上初始化。首先通过外部输入激活环境$\mathcal{Z}$, $\mathcal{Z}$激活各方, 选择他们的输入并读取他们的输出。$\mathcal{A}$可以腐化各参与方, 指示他们向$\mathcal{A}$泄露信息并执行任意指令。当环境终止时, 实际进程也将终止, 用$REAL_{\Pi,\mathcal{A},\mathcal{Z}}$表示环境在上述过程中的输出。在理想进程中, 与真实过程相似, 1台图灵机$\mathcal{S}$代表理想敌手, 1台图灵机$\mathcal{Z}$代表环境, 不同的是代表n个参与方的图灵机不运行Π, 它们都将所有输入转发给子程序机, 称为理想函数$\mathcal{F}$。理想函数$\mathcal{F}$在本地处理所有输入并返回输出给n个参与方, 用$IDEAL_{\mathcal{F},\mathcal{S},\mathcal{Z}}$表示环境在上述过程中的输出。

定义 1. 称协议Π能够 UC-实现理想函数$\mathcal{F}$, 如果对于每一个真实敌手$\mathcal{A}$, 都存在一个理想敌手$\mathcal{S}$, 使得对于每一个环境$\mathcal{Z}$, 都有$REAL_{\Pi,\mathcal{A},\mathcal{Z}} \approx IDEAL_{\mathcal{F},\mathcal{S},\mathcal{Z}}$。

2.7 非交互式零知识证明

非交互式零知识(non-interactive zero knowledge, NIZK)证明由 Blum 等人[35]提出, 其理想函数$\mathcal{F}_{\mathrm{NIZK}}$由 Groth 等人[36]形式化。对于任何元组$(x,y)$上的 NP 关系$R: \{0,1\}^* \times \{0,1\}$, 其对应的语言为$L = \{x: \exists y \text{ s.t. } R(x,y) = 1\}$, 非交互式零知识证明使得证明者能够说服任意验证者相信某个公共元素x属于语言L, 而该证明者拥有证据y。本文涉及的零知识证明理想函数如表 1 所示, 这些理想函数均可以由Σ-协议[37]安全实现, 并能够通过 Fiat-Shamir 变换[38]转换为非交互式。对这些零知识证明协议的具体描述详见附录 A。

Table 1 NIZK Ideal Functionalities and Descriptions

表 1 NIZK 理想函数及描述

理想函数	描述
$\mathcal{F}_{\mathrm{NIZK}}^{\mathrm{DL}}$	对$L_1 = \{(A,B): \exists l \text{ s.t. } B = l_i A\}$的零知识证明, 以证明离散对数的正确性.
$\mathcal{F}_{\mathrm{NIZK}}^{\mathrm{Enc}}$	对$L_2 = \{C_2: \exists R \text{ s.t. } C_2 = EGEnc_{X_2}(R)\}$的零知识证明, 以证明 ElGamal 加密的正确性.
$\mathcal{F}_{\mathrm{NIZK}}^{\mathrm{Homo}}$	对$L_3 = \{(w, C_1, C_2): \exists (r, R_1') \text{ s.t. } w = g^r \wedge C_1 = r \times_{\mathrm{E}} C_2 +_{\mathrm{E}} EGEnc_{X_2}(R_1')\}$的零知识证明, 以证明同态计算的正确性.

3 SMtAwc 协议

本文构造一种特殊的 MtAwc 协议, 称为 SMtAwc 协议。该协议将 MtAwc 协议中的乘法运算替换为椭圆曲线上的标量乘法, 能够将两方持有的标量乘法秘密分享安全地转化为加法秘密分享, 这使得能够直接计算 SM9 签名中随机数r与私钥sk乘积的加法秘密分享。此外, 将传统 MtAwc 协议中的 Paillier 加密方案替换为同样支持同态加密的 ElGamal 加密方案。一方面, ElGamal 算法可以直接加密椭圆曲线中的元素, 这与 SM9 的签名形式相契合; 另一方面, 在基于 Paillier 的 MtA 协议中, 由于 Paillier 加密的消息空间远大于协议的输入/输出空间, 协议中需要引入昂贵的范围证明(Pf.1, Pf.2)以防止恶意敌手通过恶意构造输入而造成诚实方的隐私信息泄露, 而 ElGamal 加密算法可以定义在任何存在离散对数难题的循环群G上, 因此避免了引入范围证明, 显著降低了协议中零知识证明的开销。

设G是阶为q的椭圆曲线加法循环群, P是群G的生成元。Alice 和 Bob 分别持有标量乘法分享份额$r \in \mathbb{Z}_q$, $R \in G$, 且 Alice 公开关于r的额外输入$w = g^r$。其中

Alice 和 Bob 分别将r和R作为 SMtAwc 协议的输入并得到输出R_1和R_2，满足$R_1+R_2=rR$。下面描述 SMtAwc 协议的具体构造，其中图 2 展示了两方交互过程。

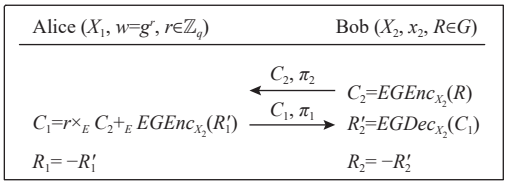


Fig. 2 Sharing phase of SMtAwc protocol

图 2 SMtAwc 协议的分享阶段

系统建立阶段：Alice 和 Bob 分别运行 ElGamal 加密的密钥生成算法，生成其公共参数pp_i，即 ElGamal 加密公钥$X_i=x_iP$。Alice 和 Bob 分别对其公钥的正确性进行零知识证明，即使用L_1的证明方法生成关于关系$\{(X_i,P):\exists x_i \text{ s.t. } X_i=x_iP\}$的离散对数零知识证明。

分享阶段：

1）Bob 执行

①使用自身公钥X_2计算$C_2=EGEnc_{X_2}(R)$；

②使用L_2的证明方法计算关于关系$\{C_2:\exists R \text{ s.t. } C_2=EGEnc_{X_2}(R)\}$的零知识证明$\pi_2$；

③发送(C_2,π_2)给 Alice。

2）Alice 执行

①随机选择$R_1'\in G$，并计算$C_1=(r\times_E C_2)+_E EGEnc_{X_2}(R_1')=EGEnc_{X_2}(rR+R_1')$；

②使用L_3的证明方法计算关于关系$\{(w,C_1,C_2):\exists(r,R_1') \text{ s.t. } w=g^r\wedge C_1=r\times_E C_2+_E EGEnc_{X_2}(R_1')\}$的零知识证明$\pi_1$；

③发送(C_1,π_1)给 Bob，并输出$R_1=-R_1'$。

3）Bob 执行

①验证π_1，若验证不通过则中止协议；

②计算并输出$R_2=EGDec_{x_2}(C_1)$。

定理 1. 如果 ElGamal 加密算法是语义安全的，那么对于任意自适应的恶意敌手$\mathcal{A}$，上述 SMtAwc 协议可以在$\{\mathcal{F}_{\text{NIZK}}^{\text{DL}},\mathcal{F}_{\text{NIZK}}^{\text{Enc}},\mathcal{F}_{\text{NIZK}}^{\text{Homo}}\}$-混合模型中 UC-实现 SMtAwc 理想函数$\mathcal{F}_{\text{SMtA}}$，如图 3 所示。

证明. 在系统建立阶段，模拟器$\mathcal{S}$可以随机选择其 ElGamal 公钥，并调用$\mathcal{F}_{\text{NIZK}}^{\text{DL}}$模拟器生成证明，以模拟敌手$\mathcal{A}$的视图。在分享阶段，敌手$\mathcal{A}$可以腐化 Alice 或腐化 Bob，下面针对这 2 种情况进行证明。

1）若 Alice 为恶意参与方。模拟器$\mathcal{S}$随机选择$R'\in G$，计算$C_2=EGEnc_{X_2}(R')$和零知识证明π_2'并将其发送给敌手$\mathcal{A}$。随后，$\mathcal{S}$接收$\mathcal{A}$调用 Alice 发送的密文C_1及零知识证明π_1。$\mathcal{S}$从零知识证明π_1中提取r,R_1'，将消息(Input,sid,r)发送给$\mathcal{F}_{\text{SMtA}}$，并收到消息$(\text{Output},sid,R_1)$。随后$\mathcal{S}$通过回溯环境，重新计算密文$C_1$并调用$\mathcal{F}_{\text{NIZK}}^{\text{Homo}}$模拟器生成证明$\pi_1$，将其发送给$\mathcal{A}$，并将$\mathcal{A}$的输出作为 Alice 的输出。模拟器与真实协议之间的唯一差别在于密文C_2和零知识证明π_2的计算。根据 ElGamal 加密算法的语义安全性，任意多项式时间敌手均无法区分模拟器生成的密文C_2和真实协议中的密文C_2。此外，根据证明π_2的零知识性，任意多项式敌手均无法区分模拟器生成的证明π_2与真实协议中的π_2。

2）若 Bob 为恶意参与方。在收到敌手$\mathcal{A}$调用 Bob 发送密文C_2及零知识证明π_2后，模拟器$\mathcal{S}$从零知识证明π_2中提取R。$\mathcal{S}$随机选择R_2'，计算$C_1'=EGEnc_{X_2}(R_2')$并生成证明π_1'发送给$\mathcal{A}$，$\mathcal{S}$将消息(Input,sid,R)发送给$\mathcal{F}_{\text{SMtA}}$，收到消息$(\text{Output},sid,R_2)$作为理想函数中 Bob 的输出。随后$\mathcal{S}$通过回溯环境重新计算$C_1=EGEnc_{X_2}(R_2)$并调用$\mathcal{F}_{\text{NIZK}}^{\text{Enc}}$模拟器生成证明$\pi_1$，将其发送给$\mathcal{A}$，并将$\mathcal{A}$的输出作为 Bob 的输出。模拟器与真实协议之间的唯一差别在于密文C_1和零知识证明π_1的计算。由于R_1'是随机选取的，符合群G上的均匀分布，因此模拟器生成的密文C_1与真实协议中的密文C_1及 Bob 解密C_1得到的消息均具有相同的分布，故敌手无法区分模拟器生成的密文C_1与真实协议中的密文C_1。此外，根据证明π_1的零知识性，任意多项式敌手均无法区分模拟器生成的证明π_1与真实协议中的证明π_1。

证毕.

系统建立阶段。当收到Alice和Bob发送的初始化消息(Init, *sid*)时：分别向Alice和Bob发送消息(Init-complete, *sid*), 并将该消息保存。
分享阶段。当分别收到Alice和Bob发送的消息(Input, *sid*, $r\in\mathbb{Z}_q$)和(Input, *sid*, $R\in G$)时, 若*sid*未被使用, 且消息(Init-complete, *sid*)已被保存：
1）随机选择$R_1\in G$，并计算$R_2=rR-R_1$。
2）向Alice发送(Output, *sid*, R_1)。
3）向Bob发送(Output, *sid*, R_2)。

Fig. 3 Ideal functionality $\mathcal{F}_{\text{SMtA}}$ in SMtAwc protocol

图 3 SMtAwc 协议的理想函数$\mathcal{F}_{\text{SMtA}}$

4 基于 SM9 的门限签名方案

本节给出基于 SM9 数字签名算法的(t,n)-门限签名方案的构造。

1）系统建立*Setup*。SM9 的系统建立阶段由 KGC 独立完成。设安全参数为λ，KGC 根据λ选择一个非对称双线性群$\mathcal{BP}=(G_1,G_2,G_T,e,q)$，其中$q$为大素数且$q>\lambda$，随机选择生成元$P_1\in G_1,P_2\in G_2$，满足$P_1=\psi(P_2)$。选择随机数$\alpha\in\mathbb{Z}_q$，2 个密码哈希函数$H_1:\{0,1\}^*\times\mathbb{Z}_q\to\mathbb{Z}_q,H_2:\{0,1\}^*\times\mathbb{Z}_q\to\mathbb{Z}_q$。计算$G_2$中的元

素 $P_{\text{pub}}=\alpha P_2$，计算 G_T 中的元素 $g=e(P_1,P_{\text{pub}})$。选择用 1B 表示的私钥生成函数识别符 hid，系统的主公钥 mpk 和主私钥 msk 分别为 $mpk=\left(P_{\text{pub}},g\right),msk=\alpha$，系统参数 $pp=(\mathcal{BP},P_1,P_2,H_1,H_2,hid)$。

2）密钥生成 $KeyGen$。已知用户标识 $ID\in\{0,1\}^*$，对用户私钥 sk 的生成遵循标准 SM9 签名算法中私钥生成方式，而对私钥 sk 的秘密分享由 KGC 独立完成。Shamir 秘密分享方案[18]可以直接应用在椭圆曲线上加法运算的分享中，以生成 sk 的秘密分享。具体而言，KGC 执行如下 3 个步骤：

①计算 $d=\alpha(H_1(ID)+\alpha)^{-1},sk=dP_1$。

②运行 (t,n)-Shamir 秘密分享协议计算 d 的分享份额 $d_i(i\in\{1,2,\cdots,n\})$。

③计算 $sk_i=d_iP_1$ 作为参与方 $\mathcal{P}_i$ 的分享份额，其中任意 t 个参与方的集合 S 均可通过计算插值公式 $dP_1=\left(\sum\limits_{i\in S}\left(d_i\prod\limits_{j\in S,j\neq i}\frac{j}{j-i}\right)\right)P_1=\sum\limits_{i\in S}\left(d_iP_1\prod\limits_{j\in S,j\neq i}\frac{j}{j-i}\right)$ 恢复秘密 sk_{ID}。

各参与方 $\mathcal{P}_i$ 在群 G_1 上执行 ElGamal 密钥生成算法生成其公私钥对 (X_i,x_i)。$\mathcal{P}_i$ 使用 L_1 的证明方法计算关系 $\{(X_i,P_1):\exists x_i\ \text{s.t.}\ X_i=x_iP_1\}$ 的离散对数零知识证明 π_i 并将其公布。

3）签名生成 $Sign$。给定待签名消息 m 和上述密钥生成协议中各方的输出。设 $P\subseteq N=\{\mathcal{P}_1,\mathcal{P}_2,\cdots,\mathcal{P}_n\}$ 为签名生成协议的参与方集合，其中 $|P|=t'(t\leqslant t'\leqslant n)$。对于签名生成协议中任何需要共享的临时秘密，都可以使用 (t',t')-秘密分享方案，而不需要使用一般的 (t,n)-方案。各参与方可以通过计算拉格朗日系数 $\lambda_{i,P}$ 将其持有的 (t,n)-秘密分享 sk_i 映射为对应的 (t',t')-秘密分享 $K_i=\lambda_{i,P}sk_i\ \text{s.t.}\ \sum\limits_{i\in P}K_i=sk$。

第 1 轮. 各参与方 $\mathcal{P}_i$ 随机选择 $r_i\in\mathbb{Z}_q$，规定 $r=\sum\limits_{i\in P}r_i$。计算 $[C_i,D_i]=Commit(g^{r_i})$ 并公布 C_i。

第 2 轮. 各参与方公布 D_i。设承诺 C_i 打开为 w_i，则各方可以计算 $w=\prod\limits_{i\in P}w_i,h=H_2(m\|w)$。

第 3 轮. 每 2 个参与方 $\mathcal{P}_i,\mathcal{P}_j$ 执行 1 次 SMtAwc 协议，将 r_i 和 K_j 作为协议输入，设协议输出分别为 X_{ij}，Y_{ij}，即

$$r_iK_j=X_{ij}+Y_{ij}。$$

参与方 $\mathcal{P}_i$ 计算 $R_i=r_iK_i+\sum\limits_{j\neq i}X_{ij}+\sum\limits_{j\neq i}Y_{ji}$。$R_i$ 即为 $r\times sk=\sum\limits_{i\in P}R_i$ 的 (t',t')-秘密分享。

第 4 轮. 各参与方计算 $S_i=R_i-hK_i$。有 $\sum\limits_{i\in P}S_i=\sum\limits_{i\in P}R_i-h\sum\limits_{i\in P}K_i=r\times sk-h\times sk=S$，$S_i$ 是 S 的 (t',t')-秘密分享。此后，各参与方执行如下操作。

4A. 各参与方 $\mathcal{P}_i$ 随机选择 $l_i,\rho_i\in\mathbb{Z}_q$，规定 $l=\sum\limits_{i\in P}l_i$，$\rho=\sum\limits_{i\in P}\rho_i$。计算 $V_i=S_i+l_iP_1,A_i=\rho_iP_1,B_i=l_i\rho_iP_1$。计算 $\left[\hat{C}_i,\hat{D}_i\right]=Commit(V_i,A_i,B_i)$ 并公布 $\hat{C}_i$。

4B. 参与方 $\mathcal{P}_i$ 使用 L_1 的证明方法生成关于关系 $\{(A_i,B_i):\exists l_i\ \text{s.t.}\ B_i=l_iA_i\}$ 的离散对数零知识证明 $\hat{\pi}_i$，并公布 $\hat{D}_i$ 和 $\hat{\pi}_i$。若该证明验证不通过则协议中止。令 $V=\prod\limits_{i\in P}e\left(V_i,H_1(ID)P_2+P_{\text{pub}}\right)g^hw^{-1}$。

4C. 参与方 $\mathcal{P}_i$ 计算 $U_i=V^{\rho_i}$ 及 $T_i=l_i\left(\sum\limits_{j\neq i}A_j\right)=l_i(\rho-\rho_i)P_1$。$\mathcal{P}_i$ 计算 $\left[\widetilde{C}_i,\widetilde{D}_i\right]=Commit(U_i,T_i)$ 并公开 $\widetilde{C}_i$。

4D. 参与方 $\mathcal{P}_i$ 公开 $\widetilde{D}_i$。验证 $\prod\limits_{i\in P}e(T_i+B_i,H_1(ID)P_2+P_{\text{pub}})=\prod\limits_{i\in P}U_i$，若不成立则中止协议。

4E. 各参与方 $\mathcal{P}_i$ 公开 S_i，可以计算 $S=\sum\limits_{i\in P}S_i$。若 (h,S) 是一个合法签名则将其作为协议的输出并结束协议，否则中止协议。

若各参与方均诚实执行此协议，则步骤 4E 输出的签名 (h,S) 为公钥 ID 下的标准 SM9 签名，如下所示：

$$\begin{aligned}S&=\sum_iS_i=\sum_i(R_i-hK_i)=\\&\sum_i\left(r_iK_i+\sum_{j\neq i}X_{ij}+\sum_{j\neq i}Y_{ji}\right)-h\sum_iK_i=\\&\sum_{i,j}r_iK_j-h\sum_iK_i=\\&\sum_ir_i\sum_iK_i-h\sum_iK_i=\left(\sum_ir_i-h\right)\sum_iK_i=\\&(r-h)\times sk=l\times sk。\end{aligned}$$

5 安全性分析

5.1 理想函数

下面对基于标识的门限签名理想函数 $\mathcal{F}_{\text{ID-tSig}}$ 进行描述，并详细叙述了 $\mathcal{F}_{\text{ID-tSig}}$ 中每次可能的激活情况。该理想函数是对 Canetti 等人[22]的门限签名理想函数的改进，如图 4 所示。在 UC 框架中，每个理想函数的实例都有一个唯一的会话标识 sid，为简单起见，令第 1 个 $Setup$ 激活决定 sid，它由一方的标识符和唯一的会话标识符组成，因此，sid 还决定了负责设置和生

Setup。当接收到来自参与方$\mathcal{P}_{Setup}$发送的初始化消息(Init, sid)时，解析$sid=(\cdots,\mathcal{P}_{Setup})$，

1) 若不存在$(\mathcal{P}_{Setup},pp,msk)$的记录，则向$\mathcal{S}$发送消息(Init, sid)。当接收到来自$\mathcal{S}$的消息(Init, $sid, S.Setup, S.KeyGen, S.Verify$)后，执行：

① 生成$(pp,msk)\leftarrow S.Setup$，向$\mathcal{S}$发送消息(PublicParams, sid, pp)；

② 当接收到来自$\mathcal{S}$的消息(PublicParams, sid, pp, OK)后，记录$(\mathcal{P}_{Setup},pp,msk)$，并记录算法$S.Setup, S.KeyGen, S.Verify$。

2) 否则使用已有记录$(\mathcal{P}_{Setup},pp,msk)$。

3) 向参与方$\mathcal{P}_{Setup}$发送消息(MasterKey, sid,pp,msk)。

KeyGen。当接收到来自某个参与方$\mathcal{P}_i$的消息(KeyGen, $sid,\mathcal{P}_i$)时，解析$sid=(\cdots,N,\mathcal{P}_{Setup})$，其中$N=\{\mathcal{P}_1,\mathcal{P}_2,\cdots,\mathcal{P}_n\}$，

1) 若$\mathcal{P}_i\in N$，则向$\mathcal{S}$发送消息(KeyGen, $sid,\mathcal{P}_i$)，并记录(KeyGen, $sid,\mathcal{P}_i$)，否则忽略此消息。

2) 若对所有$\mathcal{P}_j\in N$，(KeyGen, $sid,\mathcal{P}_j$)都已被记录，则向$\mathcal{S}$发送消息(PublicKey, sid)，当接收到来自$\mathcal{S}$的消息(PublickKey, sid, ID)后，记录(sid, ID)。

3) 向$\mathcal{P}_{Setup}$发送消息(KeyGenRequest, sid,pp,ID)，当接收到来自$\mathcal{P}_{Setup}$的消息(KeyGen, sid)后，执行：

① 若$\mathcal{P}_{Setup}$被腐化，则记录(KeyGen, $sid, ID, \perp$)；

② 否则，生成私钥$sk\leftarrow S.KeyGen(pp,msk,ID)$，并记录(KeyGen, sid, ID, sk)；

③ 向$\mathcal{S}$发送消息(KeyGen, sid, ID, sk)，并接收到来自$\mathcal{S}$的消息(KeyGen, sid, ID, sk, OK)。

4) 当收到来自参与方$\mathcal{P}_i$的消息(PublicKey, $sid,\mathcal{P}_i$)时，若$\mathcal{P}_i\in N$，且存在(sid, ID)的记录，则向$\mathcal{P}_i$输出(PublicKey, sid, ID)，否则忽略该消息。

Sign。

1) 当接收到来自某个参与方$\mathcal{P}_i$的消息(Sign, $sid, ssid, m$)时，将消息发送至$\mathcal{S}$，并记录(Sign, $sid, ssid, m, i$)；

2) 当接收到来自$\mathcal{S}$的消息(Sign, $sid, ssid, m, i$)时，若$\mathcal{P}_i$被腐化，则记录(Sign, $sid, ssid, m, i$)，否则忽略该消息；

3) 若对所有$\mathcal{P}_j\in P$，其中$P\subseteq N, |P|=t'(t\le t'\le n)$，(Sign, $sid, ssid, m, j$)都已被记录，则向$\mathcal{S}$发送消息(Sign, $sid, ssid, m$)，并执行：

① 当收到来自$\mathcal{S}$的消息(Signature, $sid, ssid, m, \sigma$)后，若存在记录$(sid, ssid, m, \sigma, 0)$，则输出错误，否则记录$(sid, ssid, m, \sigma, 1)$；

② 当收到来自$\mathcal{P}_i\in P$的消息(Signature, $sid, ssid, m$)后，若存在记录$(sid, ssid, m, \sigma, 1)$，则向$\mathcal{P}_i$输出消息(Signature, $sid, ssid, m, \sigma$)，否则忽略该消息。

Verify。当接收到来自参与方$\mathcal{P}_i$的消息(Verify, sid, m, σ, ID)后，

1) 将消息(Verify, sid, m, σ, ID)发送给$\mathcal{S}$，并执行：

① 若元组(m,σ,β')已被记录，则令$\beta=\beta'$；

② 否则，若m从未被签名过或者并非所有P中的参与方都被腐化，则令$\beta=0$；

③ 否则，令$\beta=S.Verify(m,\sigma,ID)$。

2) 记录(m,σ,β)，并向$\mathcal{P}_i$输出消息(IsTure, $sid, m, \sigma, ID, \beta$)。

Corrupt。当接收到来自$\mathcal{S}$的消息(Corrupt, $sid,\mathcal{P}_i$)后，记录参与方$\mathcal{P}_i$被腐化。

Fig. 4 Ideal functionality $\mathcal{F}_{\text{ID-tSig}}$ for identity-based threshold signature

图4 基于标识的门限签名理想函数$\mathcal{F}_{\text{ID-tSig}}$

成密钥的唯一方。该理想函数的实例将*sid*存储在第1个*Setup*激活中，并忽略所有具有不同*sid*的激活。系统建立、密钥生成、签名和验证激活的输入由环境$\mathcal{Z}$决定。

5.2 安全性证明

本节展示了本文方案可以UC-实现上述基于标识的门限签名理想函数，定理2表明如果本文方案不能UC-实现该理想函数，则存在一个针对SM9签名算法的PPT存在伪造者，这与SM9签名方案的安全性假设相矛盾。假设$\mathcal{P}_{Setup}$只能在执行1次*Setup*激活之后被腐化，这一设定与现有的基于标识的签名方案[39-41]一致，这些方案均采用了诚实*Setup*过程。这种限制也存在于其他方案中，如Blömer等人[42]基于属性的签名方案中$\mathcal{P}_{Setup}$也只能在成功执行*Setup*后被腐化。可以通过在CRS模型中定义来避免这种限制，如Abe等人[43]的盲签名方案。此外，本文将哈希函数理想化为随机谕言机模型，并使用文献[34]中定义的随机谕言机理想函数$\mathcal{F}_{\text{RO}}$及承诺理想函数$\mathcal{F}_{\text{Com}}$。

定理2. 如果SM9签名方案在自适应选择消息攻击下是存在性不可伪造的，那么对于任意自适应恶意敌手，基于SM9的门限签名方案$\Pi_{\text{SM9-tSig}}$可以在

$\{\mathcal{F}_{\text{SMtA}},\mathcal{F}_{\text{NIZK}}^{\text{DL}},\mathcal{F}_{\text{Com}},\mathcal{F}_{\text{RO}}\}$-混合模型中 UC-实现门限签名理想函数$\mathcal{F}_{\text{ID-tSig}}$。

证明. 即证对于任何现实世界的敌手$\mathcal{A}$, 都存在一个理想世界模拟器$\mathcal{S}$, 使得对于所有环境$\mathcal{Z}$, $REAL_{\Pi,\mathcal{A},\mathcal{Z}} \approx IDEAL_{\mathcal{F},\mathcal{S},\mathcal{Z}}$成立。也就是说, 没有环境$\mathcal{Z}$能够判断它是在与$\mathcal{A}$和运行方案$\Pi_{\text{SM9-tSig}}$的各参与方进行交互, 还是在与$\mathcal{S}$和理想函数$\mathcal{F}_{\text{ID-tSig}}$进行交互。在自适应恶意敌手的情况下, 环境$\mathcal{Z}$将激活$\mathcal{A}$, 且所有腐败方都将遵循$\mathcal{A}$的所有后续指令。在理想世界中, $\mathcal{S}$可以在内部运行$\mathcal{A}$并模拟它在现实世界中的视图以及腐败方的所有操作。

首先构建一个理想模拟器$\mathcal{S}$, $\mathcal{S}$维护一个用户注册表$\mathcal{L}_{\text{U}}$、一个腐败参与方名单$\mathcal{L}_{\mathcal{A}}$、一个哈希表$\mathcal{L}_m$和一个签名表$\mathcal{L}_\sigma$, 并初始化为空。

在谕示机询问过程, 当收到来自$\mathcal{Z}$的消息(Query, m)后, 若$(m,a)\in\mathcal{L}_m$, 则返回(Answer,a); 否则调用$\mathcal{F}_{\text{RO}}$获得输出值a, 返回(Answer,a)并将(m,a)添加到列表$\mathcal{L}_m$。

在*Setup*阶段, 当$\mathcal{S}$接收到来自$\mathcal{F}$的消息(Init, sid)后, 将(Init, sid, *Setup*, *KeyGen*, *Verify*)发送至$\mathcal{F}$。

在*KeyGen*阶段, 当$\mathcal{S}$接收到来自$\mathcal{F}$的消息(PublicKey, sid)后, 将$(\text{PublicKey},sid,ID)$发送至$\mathcal{F}$。若$\mathcal{P}_{Setup}$是诚实的, 则当收到来自$\mathcal{F}$的消息$(\text{KeyGen},sid,ID,sk)$后, 记录$(pp,ID,sk)$至用户注册表$\mathcal{L}_{\text{U}}$中, 并将(KeyGen, sid,ID,sk,OK)发送至$\mathcal{F}$。当看到从$\mathcal{F}$发送至$\mathcal{P}_i$的公开延迟输出(KeyGen,sid,ID)后, 模拟$\mathcal{P}_{Setup}$进行密钥生成部分。若被腐化的参与方数量小于$t(t\leqslant n)$, 模拟器检查是否已经存在(pp,ID,sk)的记录, 若存在, 则将(pp,ID,sk)从$\mathcal{P}_{Setup}$发送至$\mathcal{P}_i$; 否则, 模拟器询问SM9谕示机获得ID对应的私钥sk, 并按照协议中的密钥生成过程使用sk为各参与方生成秘密分享份额sk_i, 将(pp,ID,sk_i)从$\mathcal{P}_{Setup}$发送至$\mathcal{P}_i$。若$\mathcal{P}_{Setup}$被腐化, 则当收到来自$\mathcal{P}_{Setup}$的消息$(\text{KeyGen},sid,pp',ID,sk)$后, 若$(ID,sk)$是一组公开参数$pp'$下的合法公私钥对, 则代表$\mathcal{P}_{Setup}$将消息$(\text{KeyGen},sid)$发送至$\mathcal{F}$。当看到从$\mathcal{F}$发送至$\mathcal{P}_i$的消息$(\text{PublicKey},sid,ID)$时, 记录$(pp',ID,sk)$。

在*Sign*阶段, 若P中全部参与方均为腐败方, 则模拟器已知全部的内部状态, 模拟是完美的。若参与签名的用户中至少有1个用户是诚实的, 即恶意参与方数至多为$t-1$, 不妨设用户$\mathcal{P}_b(b\in\{1,2,\cdots,n\})$为诚实用户, 则$\mathcal{S}$模拟过程如下:

第1轮, 对$\mathcal{P}_b$, 按照协议流程随机选取r_b并广播C_b来执行协议。

第2轮, 模拟$\mathcal{P}_b$将第1轮中的随机值揭露。当收到其他所有参与方揭露的随机值w_i后, 询问SM9谕示机对消息m进行签名, 获得签名结果(h,S)并记录到签名表$\mathcal{L}_\sigma$中, 计算$w=g^h e\left(S,H_1(ID)P_2+P_{\text{pub}}\right)$。随后模拟器通过回溯环境, 调用$\mathcal{F}_{\text{Com}}$模拟器将$C_b$重新打开为$w\prod\limits_{i\in P,i\neq b}w_i^{-1}$。

第3轮, 根据$\mathcal{F}_{\text{SMtA}}$模拟参与方执行$r$和$K$的SMtA协议。此时$\mathcal{S}$正确地运行了协议, 但是它无法解密共享$X,Y$, 因此$\mathcal{S}$将$X_{ib}$设置为$\mathbb{Z}_q$中的随机值, 并计算$Y_{ib}=r_iK_b-X_{ib}$。由$\mathcal{F}_{\text{SMtA}}$的安全性可知, $\mathcal{S}$选择的随机份额正确运行了协议, 因此$\mathcal{S}$有效地广播了R_i。

第4轮, 若第3轮中所有参与者使用的K_i都与*KeyGen*阶段的一致, 则模拟器可以检测$\mathcal{P}_b$广播的S_b的正确性, 因此由于模拟器知道所有恶意参与方的S_i值, 故可以计算$S_b=S-\sum\limits_{i\neq b}S_i$, 并根据协议模拟后续操作, 此模拟过程与真实环境无法区分。

否则, $\mathcal{P}_b$的贡献值U_b将以高概率导致4D过程失败从而中止。此时, 对$\mathcal{P}_b$, 随机选择l_b,ρ_b并计算A_b,B_b, 调用$\mathcal{F}_{\text{NIZK}}^{\text{DL}}$生成相应的零知识证明, 并计算

$$
\begin{aligned}
V=&\prod_{i\in P}e\left(V_i,H_1(ID)P_2+P_{\text{pub}}\right)g^hw^{-1}=\\
&e\left(\sum_{i\in P}V_i,H_1(ID)P_2+P_{\text{pub}}\right)g^hw^{-1}=\\
&e\left(\sum_{i\in P}(R_i-hK_i+l_iP_1),H_1(ID)P_2+P_{\text{pub}}\right)g^hw^{-1}=\\
&e\left(-h\times sk+R+lP_1,H_1(ID)P_2+P_{\text{pub}}\right)g^hw^{-1}=\\
&e\left(R+lP_1,H_1(ID)P_2+P_{\text{pub}}\right)\times\\
&e\left(h\times sk,H_1(ID)P_2+P_{\text{pub}}\right)^{-1}g^hw^{-1}=\\
&e\left(R+lP_1,H_1(ID)P_2+P_{\text{pub}}\right)e(h\alpha P_1,P_2)^{-1}g^hw^{-1}=\\
&e\left(R+lP_1,H_1(ID)P_2+P_{\text{pub}}\right)e\left(hP_1,P_{\text{pub}}\right)^{-1}g^hw^{-1}。
\end{aligned}
$$

模拟器根据上述结果计算$U_i=V^{\rho_i}$及T_i, 这与真实环境是不可区分的。此外, 由于承诺的信息论安全性, 4A和4C中的承诺与真实环境是不可区分的。

在收到环境的$(\text{Corrupt},sid=(\cdots,N),\mathcal{P}_i)$后, 将$\mathcal{P}_i$加入到参与方名单$\mathcal{L}_{\mathcal{A}}$中, 向环境发送$\mathcal{P}_i$的全部内部状态。

由上述模拟过程可知, 若环境$\mathcal{Z}$能够区分与之交互的是真实环境还是理想环境, 则环境$\mathcal{Z}$可以以不可忽略的概率输出消息m的门限签名伪造, 那么模拟器可以以不可忽略的概率输出对消息m的SM9签名伪造。 证毕.

6 性能分析

6.1 理论分析

本节主要从理论层面对本文所提出的门限 SM9 签名方案的计算开销和通信开销进行了分析。由于系统建立和密钥生成算法可以由 KGC 一次性完成，不影响门限签名的效率，故本节不对其进行分析。

表 2 给出了与同为随机谕言机模型下基于素数阶群的现有文献 [28] 方案的计算开销对比，结果表明本文方案在计算效率上与文献 [28] 方案具有可比性，值得注意的是，文献 [28] 方案并未实现 UC 安全强度。

Table 2 Computation Cost Comparison for Identity-Based Threshold Signature

表 2 基于标识的门限签名计算开销对比

对比项	本文方案	文献 [28] 方案
签名时间	$6T_{\text{pm-1}}+T_{\text{pm-2}}+4T_{\text{pm-3}}+T_{\text{h}}+2tT_{\text{bp}}+3T_{\text{com}}+3(t-1)T_{\text{de-com}}+2(t-1)T_{\text{smta}}$	$T_{\text{pm-3}}+T_{\text{h}}+T_{\text{bp}}$
验签时间	$T_{\text{pm-1}}+3tT_{\text{pm-3}}+4T_{\text{bp}}+T_{\text{h}}$	$T_{\text{pm-3}}+T_{\text{h}}+2T_{\text{bp}}$

注：$T_{\text{pm-1}}$和$T_{\text{pm-2}}$分别表示群G_1和G_2上的标量乘计算时间，$T_{\text{pm-3}}$表示G_T上的指数计算时间，T_{bp}表示双线性对计算时间，T_{h}表示哈希函数平均计算时间，T_{com}表示承诺平均计算时间，$T_{\text{de-com}}$表示承诺平均验证时间，T_{smta}表示协议中 SMtA 协议计算时间。由于其他运算的开销较小，因此作出适当忽略。

此外，对协议中 SMtA 协议的开销进行了单独分析，并与文献 [19] 中的 MtA 协议进行对比，MtA 协议的计算开销对比如表 3 所示。本文提出的 SMtA 协议只需要进行G_1上的标量乘运算和离散对数零知识证明计算，而文献 [19] 中的 MtA 协议需要进行昂贵的范围证明，因此可以看出 SMtA 协议在计算开销上优于文献 [19] 中的 MtA 协议。

Table 3 Computation Cost Comparison for Protocols

表 3 协议的计算开销对比

运行方	SMtA	MtA
Alice	$10T_{\text{pm-1}}+T_{\text{pm-3}}$	$T_N+T_{\text{p-range}}+T_{\text{v-range}}$
Bob	$11T_{\text{pm-1}}+2T_{\text{pm-3}}$	$3T_N+T_{\text{p-range}}+T_{\text{v-range}}$

注：N为 Paillier 加密密钥，T_N表示群$\mathbb{Z}_{N^2}$上的指数计算时间，$T_{\text{p-range}}$和$T_{\text{v-range}}$分别表示范围证明平均证明时间和验证时间，其余符号同表 2。由于其他运算的开销较小，因此作出适当忽略。

表 4 给出了 2 种协议的理论通信开销对比。同样地，SMtA 协议在通信开销上优于文献 [19] 中的 MtA 协议。

Table 4 Comunication Cost Comparison for Protocols

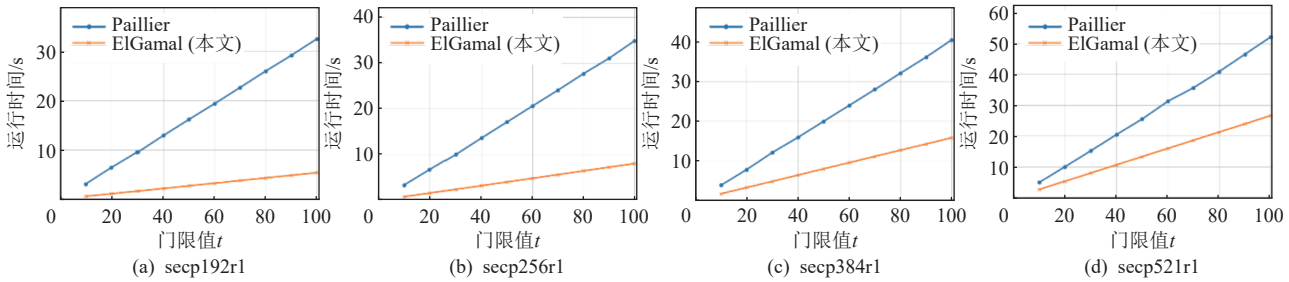
表 4 协议的通信开销对比

运行方	SMtA	MtA
Alice	$3\lvert G_1\rvert+\lvert\mathbb{Z}_q\rvert$	$\lvert\mathbb{Z}_{N^2}\rvert+\lambda_{\text{range}}$
Bob	$4\lvert G_1\rvert+2\lvert G_T\rvert+\lvert\mathbb{Z}_q\rvert$	$\lvert\mathbb{Z}_{N^2}\rvert+\lambda_{\text{range}}$

注：$\lvert\mathbb{Z}_q\rvert$表示群$\mathbb{Z}_q$中元素的长度，$\lvert\mathbb{Z}_{N^2}\rvert$表示群$\mathbb{Z}_{N^2}$中元素的长度，$\lvert G_1\rvert$表示群$G_1$中元素的长度，$\lvert G_T\rvert$表示群$G_T$中元素的长度，$\lambda_{\text{range}}$表示范围证明长度。

6.2 实验分析

本节主要从实验仿真角度对本文提出的门限 SM9 签名方案进行了实验测试，并与基于文献 [19] 中的 MtA 协议实现的门限 SM9 方案（附录 B）进行了性能对比。该对比方案与本文方案结构相似，具有很高的可比性，唯一区别在于对比方案使用了文献 [19] 中基于 Paillier 实现的 MtA 协议，而非本文提出的 SMtA 协议，并借鉴文献 [27] 中对私钥进行随机化的思想使 SM9 签名算法能够与 Paillier 加密相结合。使用标准 C++语言以及双线性对密码库[44] 进行本文算法实现，并使用 ZenGo X 的 multi-party-ecdsa 密码库[45] 中的 MtA 协议作为文献 [19] 中 MtA 协议的代码实现。从美国国家标准技术研究院（National Institute of Standards and Technology, NIST）标准[46] 中选取了 192~521 b 的不同素数域下的椭圆曲线，分别对本文提出的门限 SM9 签名方案与对比方案进行了性能测试。本地运行环境为虚拟机，硬件环境分别为内存大小 54 GB、处理器型号为 Intel® Xeon® Silver 4314 CPU 以及硬件环境为内存大小 8 GB、处理器型号为 Intel® Core™ i5-12500H，操作系统版本为 64 位的



(a) secp192r1 (b) secp256r1 (c) secp384r1 (d) secp521r1

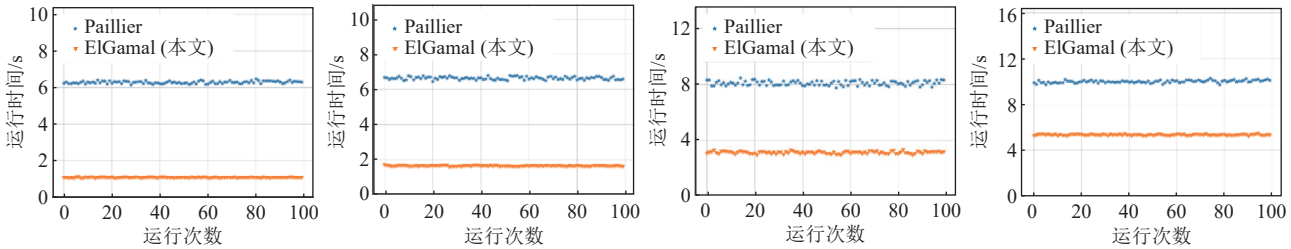
Fig. 5 Time comparison for the threshold signature algorithm under different curves

图 5 不同曲线下门限签名算法时间比较

Ubuntu 22.04.1 LTS。

在忽略网络延迟的情况下对 1 次签名协议中单个参与方的运行时间进行了测试，由于实际协议中各参与方的计算可以并行运行，因此这可以代表除网络延迟外运行协议所需的总时间。通过设置门限值t从 10 增长至 100，步长为 10 进行多次实验，并根据 NIST 标准，选取了 192~521 b 的不同素数域下的 4 条椭圆曲线，得到不同椭圆曲线下 1 次签名所需时间与门限值t之间关系，如图 5 所示。为了保证实验结果的准确性，每个算法均重复执行 100 次，得到平均 1 次签名的时间开销。可以看出，签名所需时间均随门限值t增加而呈线性增长，但相对于对比方案而言本文方案增长更慢，总体而言本文方案总体时间开销显著少于对比方案。以 secp256r1 曲线为例，当门限值t = 20时，对比方案的签名协议运行时间约为 6.374 s，本文签名协议运行时间约为 1.061s，速度约为对比方案的 7.0 倍；当门限值t = 100时，对比方案的签名协议运行时间约为 32.576 s，本文签名协议运行时间约为 5.359 s，速度约为对比方案的 6.1 倍。为表明 100 次重复实验数据的具体统计分布，将t = 20时每次实验结果以散点图形式展示，如图 6 所示。

为了进一步评估 SMtA 协议的性能，如表 5 所示，使用上述 4 条椭圆曲线对文献 [19] 中基于 Paillier 实现的 MtA 协议和本文提出的基于 ElGamal 的 SMtA 协议进行了单独测试。可以看出，SMtA 协议运行速度显著高于文献 [19] 中基于 Paillier 的 MtA 协议。具体而言，在 LAN 环境中，SMtA 运行速度约为 MtA 的 7~11 倍；在 WAN 环境中，SMtA 运行速度约为 MtA 的 2~5.5 倍。同时，对 MtA 协议及 SMtA 协议中开销最大的零知识证明部分进行了单独测试，其中 AliceProof 和 BobProof 分别表示协议中 Alice 和 Bob 两方所进行的零知识证明。为体现 SMtA 协议的运



(a) secp192r1 (b) secp256r1 (c) secp384r1 (d) secp521r1

Fig. 6 Time distribution of the 100-times threshold signature algorithm when t=20 under different curves

图 6 不同曲线下 t=20 时 100 次门限签名算法时间分布

Table 5 Comparison of MtA and SMtA Protocols

表 5 MtA 协议和 SMtA 协议比较

协议	曲线	证明过程	证明时间/ms	验证时间/ms	LAN 通信时间/ms	WAN 通信时间/ms	通信量/KB	总时间/ms
MtA	secp192r1	AliceProof	54.23	41.14	5.85	292.94	8.75	388.99(LAN) 969.42(WAN)
		BobProof	107.18	100.04	6.72	300.06	21.56	
	secp256r1	AliceProof	76.89	43.79	6.90	287.57	12.50	452.51(LAN) 1 000.33(WAN)
		BobProof	114.73	113.01	7.78	274.93	29.75	
	secp384r1	AliceProof	201.57	133.79	7.72	280.35	27.75	1 503.79(LAN) 2 065.1(WAN)
		BobProof	430.33	452.06	9.01	297.69	54.38	
	secp521r1	AliceProof	686.94	534.23	8.02	287.58	49.00	3 518.96(LAN) 4 076.37(WAN)
		BobProof	1 035.38	1 005.72	7.49	282.64	117.5	
SMtA	secp192r1	AliceProof	1.06	0.93	4.23	217.49	0.16	**45.58**(LAN) **471.11**(WAN)
		BobProof	9.63	17.53	4.38	216.65	0.52	
	secp256r1	AliceProof	1.65	1.55	3.79	232.26	0.22	**65.17**(LAN) **509.02**(WAN)
		BobProof	14.19	24.99	4.77	220.15	0.69	
	secp384r1	AliceProof	3.61	3.66	4.41	236.68	0.33	**151.57**(LAN) **600.1**(WAN)
		BobProof	33.61	66.75	4.79	221.05	1.03	
	secp521r1	AliceProof	6.40	7.56	4.39	207.42	0.44	**327.18**(LAN) **741.08**(WAN)
		BobProof	85.24	163.80	4.62	215.49	1.38	

注：黑体数值表示相对优势数值。

行效率，对 SMtA 协议进行测试，如表 6 所示。

Table 6 Computation Cost of SMtA Protocol

表 6 SMtA 协议计算开销 ms

曲线	阶段 1	阶段 2	阶段 3	总时间
secp192r1	2.16	11.12	18.07	31.35
secp256r1	3.43	16.63	25.87	45.93
secp384r1	7.50	39.23	68.67	115.40
secp521r1	13.53	96.51	167.32	277.36

注：阶段 1、阶段 2、阶段 3 分别对应 SMtA 中分享过程中的 3 个阶段。

结合上述 SMtA 运行结果，对本文基于 SM9 的门限签名方案通信开销进行了分析，当门限为 t 时，各轮通信开销及总通信开销如表 7 所示。

Table 7 Communication Cost of SM9 Based Threshold Signature Scheme

表 7 基于 SM9 的门限签名方案通信开销 KB

曲线	第 1 轮	第 2 轮	第 3 轮	第 4 轮	总通信量
secp192r1	0.02t	0.02t	0.68t	0.15t	0.87t
secp256r1	0.03t	0.03t	0.91t	0.19t	1.16t
secp384r1	0.05t	0.05t	1.36t	0.28t	1.74t
secp521r1	0.06t	0.06t	1.82t	0.38t	2.32t

为了验证本文方案在区块链场景中的可行性，在以太坊、Hyperledger Fabric 和腾讯云服务平台[47]长安链体验网络 3 个不同的区块链平台上对该门限签名方案进行了测试。如图 7 所示，在 Hyperledger Fabric 中，使用了 NIST 标准的 4 条不同曲线进行测试，测试版本为 Fabric 2.6，测试网络环境包括 2 个 peer 节点和 1 个 orderer 节点。此外，使用以太坊客户端 Geth（版本为 1.10.8）创建了 1 个私链网络，并使用智能合约的方式进行了原型实现。由于算法中双线性配对操作和复杂的椭圆曲线上运算会消耗大量的 gas，故参照以太坊中对 BLS 签名算法的集成方式[48-49]，采用编写以太坊预编译合约并使用智能合约调用预编译合约的方式实现签名算法，本文使用以太坊预编译合约中仅支持的 bls12381 曲线和 bn256 曲线进行实验。腾讯云服务平台中长安链体验网络版本为 v2.2.1，网络组织包括北京、上海、深圳和广州 4 个测试组织，具体网络配置信息为：节点配置 16 核 CPU、32 GB 内存、1 000 GB 磁盘。由于网络延迟、通信开销等因素影响，测试结果较本地 Fabric 运行结果有约 30% 的增加。实验结果表明，本文基于 SM9 的门限签名方案运行时间随门限值 t 的增加线性增长，且随曲线素数域的增大而增加，在区块链环境中具有可接受的运行速度。

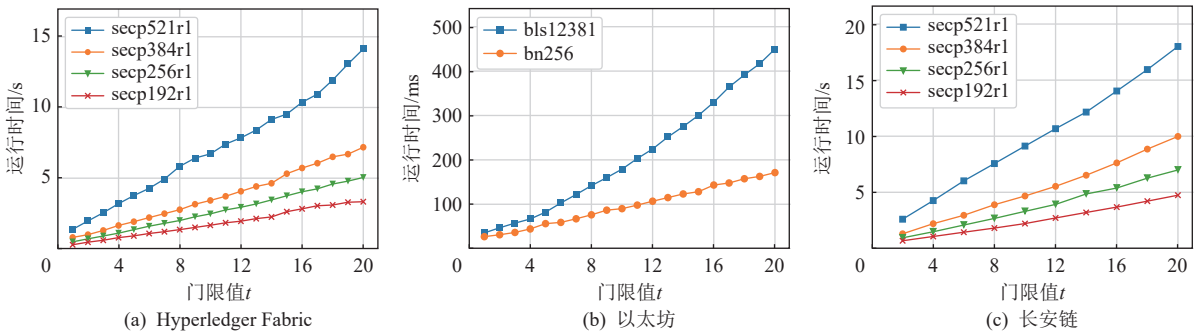


Fig. 7 Running time comparison for the signature algorithm on different blockchain platforms

图 7 不同区块链平台上签名算法运行时间比较

7 总 结

本文提出了一种基于 ElGamal 加密算法的 SMtAwc 协议，并应用其构建了一个基于国密 SM9 的门限签名方案。此外，该方案在 UC 框架内实现了门限签名理想函数，在随机谕言机模型下证明了其安全性。理论分析和实验结果表明，与现有方案相比，本文基于的 SM9 签名方案效率更高。特别地，当门限值为 100 时，速度约为对比方案的 6.1 倍。本文提出的基于 SM9 的门限签名方案是一种 IBC 体制下的门限签名方案，如何构造 IBC 体制下通用构造模型，是后续可能的改进方向。

作者贡献声明： 董佶圣、沈子楠负责方案整体设计并撰写论文；李聪负责部分算法思路和实验方法；胡达齐负责方案实验验证；沈晴霓、吴中海提出指导意见并修改论文。

参 考 文 献

[1] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system[EB/OL].

[2025-06-12]. https://www.bitcoin.org/bitcoin.pdf

[2] Liang Wei, Xie Siqi, Li K C, et al. MC-DSC: A dynamic secure resource configuration scheme based on medical consortium blockchain[J]. IEEE Transactions on Information Forensics and Security, 2024, 19: 3525−3538

[3] Ullah F, Al-Turjman F. A conceptual framework for blockchain smart contract adoption to manage real estate deals in smart cities[J]. Neural Computing and Applications, 2023, 35(7): 5033−5054

[4] Kashif M, Kalkan K. EPIoT: Enhanced privacy preservation based blockchain mechanism for Internet-of-things[J]. Computer Networks, 2024, 238: 110107

[5] Li Zhezhou, Liang Xiaolong, Wen Qiyuan, et al. The analysis of financial network transaction risk control based on blockchain and edge computing technology[J]. IEEE Transactions on Engineering Management, 2024, 71: 5669−5690

[6] Shamir A. Identity-based cryptosystems and signature schemes[C]// Proc of the 4th CRYPTO. Berlin: Springer, 1985: 47−53

[7] Boneh D, Franklin M. Identity-based encryption from the Weil pairing[C]//Proc of the 21st Annual Int Cryptology Conf. Berlin: Springer, 2001: 213−229

[8] State Cryptography Administration. GM/T 0044—2016 Identity-Based Cryptographic Algorithms SM9[S]. Beijing: Standard Press of China, 2016 (in Chinese)
(国家密码管理局. GM/T 0044—2016 SM9 标识密码算法[S]. 北京: 中国标准出版社, 2016)

[9] The International Organization for Standardization and the International Electrotechnical Commission. ISO/IEC 14888−3: 2018/CDAmd 1, IT Security Techniques — Digital Signatures with Appendix — Part 3: Discrete Logarithm Based Mechanisms [S]. Geneva: ISO, 2018

[10] Wan Zhiguo, Liu Wei, Cui Hui. HIBEChain: A hierarchical identity-based blockchain system for large-scale IoT[J]. IEEE Transactions on Dependable and Secure Computing, 2022, 20(2): 1286−1301

[11] Sharma P, Moparthi N R, Namasudra S, et al. Blockchain-based IoT architecture to secure healthcare system using identity-based encryption[J]. Expert Systems, 2022, 39(10): e12915

[12] Yin Jin, Xiao Yang, Pei Qingqi, et al. SmartDID: A novel privacy-preserving identity based on blockchain for IoT[J]. IEEE Internet of Things Journal, 2022, 10(8): 6718−6732

[13] ChainMaker. ChainMaker introduces a new identity authentication system IBC: Breaking through incomprehensible limitations[EB/OL]. (2023-04-04)[2025-06-12]. https://chainmaker.org.cn/news/html/22f49007-8fc0-4664-8423-89a502dc3097.html (in Chinese)
(长安链. 长安链引入全新身份认证体系 IBC: 突破公钥不可理解的限制[EB/OL]. (2023-04-04)[2025-06-12]. https://chainmaker.org.cn/news/html/22f49007-8fc0-4664-8423-89a502dc3097.html)

[14] Yang Yatao, Cai Juliang, Zhang Xiaowei, et al. Privacy preserving scheme in block chain with provably secure based on SM9 algorithm[J]. Journal of Software, 2019, 30(6): 1692−1704 (in Chinese)
(杨亚涛, 蔡居良, 张筱薇, 等. 基于 SM9 算法可证明安全的区块链隐私保护方案[J]. 软件学报, 2019, 30(6): 1692−1704)

[15] An Haoyang, He Debiao, Bao Zijian, et al. Ring signature based on the sm9 digital signature and its application in blockchain privacy protection[J]. Journal of Computer Research and Development, 2023, 60(11): 2545−2554 (in Chinese)
(安浩杨, 何德彪, 包子健, 等. 基于 SM9 数字签名的环签名及其在区块链隐私保护中的应用[J]. 计算机研究与发展, 2023, 60(11): 2545−2554)

[16] Desmedt Y, Frankel Y. Shared generation of authenticators and signatures[C]//Proc of the 11th CRYPTO. Berlin: Springer, 1991: 457−469

[17] Gennaro R, Jarecki S, Krawczyk H, et al. Robust threshold DSS signatures[C]//Proc of the Int Conf on the Theory and Application of Cryptographic Techniques (EUROCRYPT'96). Berlin: Springer, 1996: 354−371

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

[19] Gennaro R, Goldfeder S, Narayanan A. Threshold-optimal DSA/ECDSA signatures and an application to bitcoin wallet security[C]//Proc of the 14th Int Conf on Applied Cryptography and Network Security. Berlin: Springer, 2016: 156−174

[20] Gennaro R, Goldfeder S. Fast multiparty threshold ECDSA with fast trustless setup[C]//Proc of the 25th ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2018: 1179−1194

[21] Lindell Y, Nof A. Fast secure multiparty ECDSA with practical distributed key generation and applications to cryptocurrency custody[C]//Proc of the 25th ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2018: 1837−1854

[22] Canetti R, Gennaro R, Goldfeder S, et al. UC non-interactive, proactive, threshold ECDSA with identifiable aborts[C]//Proc of the 27th ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2020: 1769−1787

[23] Tymokhanov D, Shlomovits O. Alpha-rays: Key extraction attacks on threshold ECDSA implementations[J/OL]. Cryptology ePrint Archive: 2021/1621, 2021 [2025-06-12]. https://eprint.iacr.org/2021/1621

[24] Castagnos G, Catalano D, Laguillaumie F, et al. Two-party ECDSA from hash proof systems and efficient instantiations[C]//Proc of the 39th Annual Int Cryptology Conf. Berlin: Springer, 2019: 191−221

[25] Castagnos G, Catalano D, Laguillaumie F, et al. Bandwidth-efficient threshold EC-DSA[C]//Proc of the 23rd IACR Int Conf on Practice and Theory of Public-Key Cryptography. Berlin: Springer, 2020: 266−296

[26] Doerner J, Kondi Y, Lee E, et al. Secure two-party threshold ECDSA from ECDSA assumptions[C]//Proc of the 39th IEEE Symp on Security and Privacy. Piscataway, NJ: IEEE, 2018: 980−997

[27] Doerner J, Kondi Y, Lee E, et al. Threshold ECDSA from ECDSA assumptions: The multiparty case[C]//Proc of the 40th IEEE Symp on Security and Privacy. Piscataway, NJ: IEEE, 2019: 1051−1066

[28] Baek J, Zheng Yuliang. Identity-based threshold signature scheme from the bilinear pairings[C]//Proc of the Int Conf on Information Technology: Coding and Computing (ITCC'04). Piscataway, NJ: IEEE, 2004: 124−128

[29] Xiong Hu, Qin Zhiguang, Li Fagen. Identity-based threshold signature secure in the standard model[J]. International Journal of Network Security, 2010, 10(1): 75−80

[30] Atapoor S. Identity-based threshold signatures from isogenies[C]// Proc of the 19th IMA Int Conf on Cryptography and Coding. Berlin: Springer, 2023: 220−240

[31] Jiang Yan, Zhu Youwen, Wang Jian, et al. Fully distributed identity-based threshold signatures with identifiable aborts[J]. Frontiers of Computer Science, 2023, 17(5): 175813

[32] Mu Yongheng, Xu Haixia, Li Peili, et al. Secure two-party SM9 signing[J]. Science China Information Sciences, 2020, 63(8): 1−3

[33] Lindell Y. Fast secure two-party ECDSA signing[C]//Proc of the 37th Annual Int Cryptology Conf. Berlin: Springer, 2017: 613−644

[34] Canetti R. Universally composable security: A new paradigm for cryptographic protocols[C]//Proc of the 42nd IEEE Symp on Foundations of Computer Science. Piscataway, NJ: IEEE, 2001: 136−145

[35] Blum M, Feldman P, Micali S. Non-interactive zero-knowledge and its applications[C]//Proc of the 20th Annual ACM Symp on Theory of Computing. New York: ACM, 1988: 103−112

[36] Groth J, Ostrovsky R, Sahai A. Perfect non-interactive zero knowledge for NP[C]//Proc of the 25th Int Conf on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2006: 339−358

[37] Camenisch J, Krenn S, Shoup V. A framework for practical universally composable zero-knowledge protocols[C]//Proc of the 17th Int Conf on the Theory and Application of Cryptology and Information Security. Berlin: Springer, 2011: 449−467

[38] Fiat A, Shamir A. How to prove yourself: Practical solutions to identification and signature problems[C]//Proc of the 6th CRYPTO. Berlin: Springer, 1986: 186−194

[39] Paterson K G, Schuldt J C N. Efficient identity-based signatures secure in the standard model[C]//Proc of the 11th Australasian Conf on Information Security and Privacy. Berlin: Springer, 2006: 207−222

[40] Choon J C, Hee C J. An identity-based signature from gap Diffie-Hellman groups[C]//Proc of the 6th Int Workshop on Practice and Theory in Public Key Cryptography. Berlin: Springer, 2002: 18−30

[41] Galindo D, Garcia F D. A Schnorr-like lightweight identity-based signature scheme[C]//Proc of the 2nd Int Conf on Cryptology in Africa. Berlin: Springer, 2009: 135−148

[42] Blömer J, Eidens F, Juhnke J. Enhanced security of attribute-based signatures[C]//Proc of the 17th Int Conf on Cryptology and Network Security. Berlin: Springer, 2018: 235−255

[43] Abe M, Ohkubo M. A framework for universally composable non-committing blind signatures[J]. International Journal of Applied Cryptography, 2012, 2(3): 229−249

[44] Yu J. SM9_core[EB/OL]. [2025-06-12]. https://github.com/John-Yu/SM9_core

[45] ZenGo-X. Multi-party-ECDSA[EB/OL]. [2025-06-12]. https://github.com/ZenGo-X/multi-party-ecdsa

[46] Chen L, Moody D, Regenscheid A, et al. Recommendations for discrete logarithm-based cryptography: Elliptic curve domain parameters[R]. Gaithersburg: National Institute of Standards and Technology, 2019

[47] Tencent Cloud. Tencent Blockchain as a Service TBaaS[EB/OL]. [2025-06-12]. https://cloud.tencent.com/product/tbaas (in Chinese)
(腾讯云. 腾讯云区块链服务平台 TBaaS[EB/OL]. [2025-06-12]. https://cloud.tencent.com/product/tbaas)

[48] Ethereum. Go-ethereum[EB/OL]. [2025-06-12]. https://github.com/ethereum/go-ethereum/tree/master/crypto

[49] Witnet. Bls-solidity[EB/OL]. [2025-06-12]. https://github.com/witnet/bls-solidity



Dong Jisheng, born in 1993. PhD. His main research interests include cryptography and blockchain.

董佶圣, 1993 年生。博士。主要研究方向为密码学、区块链。



Li Cong, born in 1990. PhD. His main research interests include applied cryptography, blockchain, cloud security, and big data security.

李　聪, 1990 年生。博士。主要研究方向为应用密码学、区块链、云安全、大数据安全。



Shen Zinan, born in 2001. Master candidate. Her main research interests include cryptography and blockchain.

沈子楠, 2001 年生。硕士研究生。主要研究方向为密码学、区块链。



Hu Daqi, born in 2001. Master candidate. His main research interests include cryptography, blockchain, trusted execution environment, and zero-knowledge proof.

胡达齐, 2001 年生。硕士研究生。主要研究方向为密码学、区块链、可信执行环境、零知识证明。



Shen Qingni, born in 1970. PhD, professor. Senior member of CCF. Her main research interests include operating system and virtualization security, privacy preserving in cloud and big data, and trusted computing.

沈晴霓, 1970 年生。博士，教授。CCF 高级会员。主要研究方向为操作系统和虚拟化安全、云和大数据中的隐私保护、可信计算。



Wu Zhonghai, born in 1968. PhD, professor. Distinguished member of CCF. His main research interests include big data, machine learning, software and system security, blockchain and privacy computing, and industrial IoT and intelligence.

吴中海, 1968 年生。博士，教授。CCF 杰出会员。主要研究方向为大数据、机器学习、软件与系统安全、区块链与隐私计算、工业物联与智能。

附录 A. 零知识证明协议。

本节叙述 2.7 节中的零知识证明实例。

1）$\Pi_{\mathrm{NIZK}}^{\mathrm{DL}}$描述。

对$L_1=\{(A,B):\exists l \text{ s.t. } B=lA\}$的零知识证明叙述如下：

①证明者随机选择$t\in\mathbb{Z}_q$，计算$T=tA$；

②证明者计算$c=H(T)$；

③证明者计算$z=l+ct$，并将$\{T,z\}$发送给验证者；

④验证者计算$c=H(T)$，并验证等式$zA=B+cT$，若成立则验证通过，否则不通过。

2）$\Pi_{\mathrm{NIZK}}^{\mathrm{Enc}}$描述。

对$L_2=\{C_2=(C_{21},C_{22}):\exists r_1\in\mathbb{Z}_q,R\in G \text{ s.t. } C_2=(R+r_1X_2,r_1P)=EGEnc_{X_2}(R)\}$的零知识证明叙述如下：

①证明者随机选择$a\in\mathbb{Z}_q$，计算$A=aP$；

②证明者计算$c=H(A)$，其中$H(\cdot):G\to\mathbb{Z}_q$为一个公开的哈希函数；

③证明者计算$z=r_1+ca$，并将$\{A,z\}$发送给验证者；

④验证者计算$c=H(A)$，并验证等式$zP=C_{22}+cA$，若成立则验证通过，否则不通过。

3）$\Pi_{\mathrm{NIZK}}^{\mathrm{Homo}}$描述。

对$L_3=\{(w,C_1=(C_{11},C_{12}),C_2=(C_{21},C_{22})):\exists r,r_2\in\mathbb{Z}_q,R_1'\in G \text{ s.t. } w=g^r\wedge C_1=(rC_{21}+R_1'+r_2X_2,rC_{22}+r_2P)=r\times{}_{\mathrm{E}}C_2+{}_{\mathrm{E}}EGEnc_{X_2}(R_1')\}$的零知识证明叙述如下：

①证明者随机选择$a_1,a_2\in\mathbb{Z}_q$，计算$A_0=g^{r_1},A_1=a_1C_{22},A_2=a_2P$；

②证明者计算$c=H(A_1,A_2)$；

③证明者计算$z_1=r+ca_1$，$z_2=r_2+ca_2$，并将$\{A_0,A_1,A_2,z_1,z_2\}$发送给验证者；

④验证者计算$c=H(A_1,A_2)$，并验证等式$z_1P=wA_0^c,z_1C_{22}+z_2P=C_{12}+cA_1+cA_2$，若成立则验证通过，否则不通过。

附录 B. 基于文献 [15] 的门限签名对比方案。

本节叙述 6.2 节中基于文献 [19] 中 MtA 协议构造的对比方案。其中系统建立和密钥生成阶段同本文方案。下面给出(t,n)-门限签名生成协议构造。设待签名消息为m，$P\subseteq[1,2,\cdots,n]$为签名生成协议的参与方集合，其中$|P|=t'(t\leqslant t'\leqslant n)$，密钥生成协议中参与方$\mathcal{P}_i$所获得的签名私钥秘密分享份额为$sk_i$，各参与方可以通过计算拉格朗日系数$\lambda_{i,P}$将其持有的$(t,n)$-秘密分享$sk_i$映射为对应的$(t',t')$-秘密分享$K_i=\lambda_{i,P}sk_i$。

第 1 轮. 各参与方$\mathcal{P}_i$随机选择$r_i\in\mathbb{Z}_q$，规定$r=\sum_{i\in P}r_i$。计算$[C_i,D_i]=Commit(g^{r_i})$并公布$C_i$。

第 2 轮. 各参与方公布D_i。设承诺C_i打开为w_i，则各方可以计算$w=\prod_{i\in P}w_i,h=H_2(m\|w)$。

第 3 轮. 每 2 个参与方$\mathcal{P}_i,\mathcal{P}_j$执行 1 次 MtAwc 协议，其中$\mathcal{P}_i$和$\mathcal{P}_j$分别持有秘密$r_i$和$K_j$。$\mathcal{P}_j$随机选择$\gamma_j\in\mathbb{Z}_q$，计算并公开$\Gamma_j=\gamma_j^{-1}K_j$。$\mathcal{P}_i$和$\mathcal{P}_j$分别将$r_i$和$\gamma_j$作为 MtAwc 协议的输入，设协议输出分别为$x_{ij},y_{ij}$，即

$$r_i\gamma_j=x_{ij}+y_{ij}。$$

参与方$\mathcal{P}_i$计算$R_i=r_iK_i+\sum_{j\neq i}x_{ij}\Gamma_j+\sum_{j\neq i}y_{ji}\Gamma_i$。$R_i$即为$r\times sk=\sum_{i\in P}R_i$的$(t',t')$-秘密分享。

第 4 轮. 各参与方计算$S_i=R_i-hK_i$。有$\sum_{i\in P}S_i=\sum_{i\in P}R_i-h\sum_{i\in P}K_i=r\times sk-h\times sk=S$，$S_i$是$S$的$(t',t')$秘密分享。

4A. 各参与方$\mathcal{P}_i$随机选择$l_i,\rho_i\in\mathbb{Z}_q$，规定$l=\sum_{i\in P}l_i,\rho=\sum_{i\in P}\rho_i$。计算$V_i=S_i+l_iP_1,A_i=\rho_iP_1,B_i=l_i\rho_iP_1$。计算$\left[\hat{C}_i,\hat{D}_i\right]=Commit(V_i,A_i,B_i)$并公布$\hat{C}_i$。

4B. 参与方$\mathcal{P}_i$使用L_1的证明方法生成关于关系$\{(A_i,B_i):\exists l_i \text{ s.t. } B_i=l_iA_i\}$的离散对数零知识证明$\hat{\pi}_i$，并公布$\hat{D}_i$和$\hat{\pi}_i$。若该证明验证不通过则协议中止。令$V=\prod_{i\in P}e\left(V_i,H_1(ID)P_2+P_{\mathrm{pub}}\right)g^hw^{-1}$。

4C. 参与方$\mathcal{P}_i$计算$U_i=V^{\rho_i}$及$T_i=l_i\left(\sum_{j\neq i}A_j\right)=l_i(\rho-\rho_i)P_1$。$\mathcal{P}_i$计算$\left[\widetilde{C}_i,\widetilde{D}_i\right]=Commit(U_i,T_i)$并公开$\widetilde{C}_i$。

4D. 参与方$\mathcal{P}_i$公开$\widetilde{D}_i$。验证$\prod_{i\in P}e(T_i+B_i,H_1(ID)P_2+P_{\mathrm{pub}})=\prod_{i\in P}U_i$，若不成立则中止协议。

4E. 各参与方$\mathcal{P}_i$公开S_i，可以计算$S=\sum_{i\in P}S_i$。若(h,S)是一个合法签名则将其作为协议的输出并结束协议，否则中止协议。