一种基于 SM2 的支持门限修改的数字签名方案

周权[1]，谢淑婷[1]，曾志康[1]，陈丽丽[2]，卢子冲[2]
（1.广州大学数学与信息科学学院，广东 广州 510006；2.广州大学计算机科学与网络工程学院，广东 广州 510006）

摘　要：现有的数字签名方案在电子商务、金融交易、医疗系统等场景中广泛应用，但无法在保持签名不变时对已签名的消息内容进行修改，而重复进行签名操作会提高复杂性和延迟。对此，提出一种基于 SM2 算法并支持门限修改的数字签名方案 SM2-TM。该方案支持以门限的方式协同修改已签名的消息内容，并通过参数控制与构造哈希碰撞来实现签名的不变性。此外，它有效解决了复杂的证书存储和管理以及密钥托管问题。隐私与安全性分析结果表明，SM2-TM 方案具有不可伪造性、不变性、可追踪性等。性能分析结果表明，SM2-TM 方案在计算开销上具有一定优势，在公/私钥及签名的存储开销上也有显著优势，存储开销分别仅为 64、32 和 64 Byte。

关键词：SM2 算法；数字签名；门限修改；无证书；不变性
中图分类号：TP309　　**文献标志码：**A　　**DOI：**10.19678/j.issn.1000-3428.0070282

SM2-based Digital Signature Scheme Supporting Threshold Modification

ZHOU Quan[1], XIE Shuting[1], ZENG Zhikang[1], CHEN Lili[2], LU Zichong[2]
(1. School of Mathematics and Information Sciences, Guangzhou University, Guangzhou 510006, Guangdong, China;
2. School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou 510006, Guangdong, China)

【Abstract】Existing digital signature schemes are widely used in e-commerce, financial transactions, and medical systems. However, they cannot modify the content of signed messages while keeping the signature unchanged. Moreover, repeatedly performing signature operations increases complexity and latency. To address this issue, a digital signature scheme based on the SM2 algorithm and a supporting threshold modification, called SM2-TM, is proposed. This scheme supports the collaborative modification of signed message content in a threshold manner and achieves signature immutability through parameter control and the construction of hash collisions. Furthermore, it effectively solves complex certificate storage and management as well as key escrow issues. Privacy and security analyses results show that the SM2-TM scheme possesses unforgeability, immutability, and traceability. The performance analysis results indicate that the SM2-TM scheme has certain advantages in terms of computational overhead and significant advantages in terms of the storage overhead of public keys, private keys, and signatures, with storage overheads of only 64, 32, and 64 Byte, respectively.

【Key words】SM2 algorithm; digital signature; threshold modification; certificateless; invariance

0　引言

在数字化时代，通信双方或多方通常使用数字签名技术来确认消息的完整性和来源。传统的签名方式依赖于笔迹、指纹等物理标识来验证签名者，并证明其内容的真实性。然而，随着科技的迅猛发展，这些传统模式已难以满足现代生活和工作模式的需求。为了适应不断变化的环境，数字签名技术应运而生，并展现出其诸多优势。首先，数字签名能够提供更高的安全性，确保信息在传输过程中不被篡改。此外，数字签名的应用促进了“无接触式”交易的普及，这种方式不仅降低了物理接触的风险，还提高了效率。同时，数字签名技术具有低成本、高效便捷、易于管理等特点，使得它正逐渐取代传统签名，并广泛应用于电子政务[1]、医疗保健[2-4]、数字交易[5-6]、电子合同[7-9]等领域。

尽管数字签名技术已广泛应用于多个行业，但大部分应用依赖于国外的密码技术，这限制了其在算法层面上的自主可控性。而 SM2 作为我国制定的密码算法[10]，因其具备速度快、安全性高等特点，已广泛应用于各种安全通信和认证场景中，为数字签名技术的发展提供了有效的国产化解决方案。

现有的数字签名技术大致可分为基于公钥基础

基金项目：国家重点研发计划(2021YFA1000600)；国家自然科学基金(12171114)；广州大学研究生创新能力培养项目(JCCX2024-012)。
作者简介：周权，男，副教授、博士，主研方向为可信计算、传感器网络安全、云计算安全；谢淑婷(通信作者)，硕士研究生；曾志康，博士研究生；陈丽丽、卢子冲，硕士研究生。
收稿日期：2024-08-22　　**修回日期：**2024-12-21　**E-mail：**s.t.xie@foxmail.com

设施(PKI)[11-12]和基于身份(ID)[7-8]两类。基于PKI的数字签名技术部署简单,维护方便,但需要较高的存储成本,并面临复杂的证书管理问题。而基于ID的数字签名技术可以缓解上述不足并提高效率,但存在密钥托管的风险,一旦密钥生成中心(KGC)遭到破坏,用户的密钥将会被泄露。为解决这些问题,学者们提出了无证书签名技术[13-15]。一般而言,用户的完整密钥由其生成的秘密值和KGC生成的部分密钥组成,这可确保即使攻击者破坏KGC,也无法窃取用户的完整密钥,从而防止其伪造有效的签名。

尽管现有的数字签名技术已有十分广泛的应用,但在签名后对消息进行细微修改时表现不足。一旦消息签名后,任何修改都可能引入复杂性和延迟,尤其是在需要多方审核时,会导致处理流程变长,影响信息更新和决策效率。例如,在医疗系统中,医生需要对患者的电子病历进行更新和签署,病历确认后,如果需要小幅修改(如更新症状或治疗方案),通常需要重新经过多个医务人员的审核和签署,这会导致延误并影响患者治疗。因此,理想的电子医疗记录系统应在遵循隐私法规的前提下,简化修改流程,同时保留历史记录。类似的情况在电子政务、电子商务等行业中也存在。

本文提出一种国产化的、安全的、基于SM2的数字签名方案,在不改变原有消息签名的情况下,支持对消息进行细微修改。本文的主要贡献如下:

1)提出一个SM2-TM方案,允许通过门限的方式对已签名的消息内容进行协同修改,并通过参数控制和构建的哈希碰撞来实现签名的不变性。

2)克服复杂的证书存储、管理和密钥托管以及在签名后对消息进行细微修改时表现不足的问题,拓展SM2签名的功能。

3)分析并证明SM2-TM方案能满足隐私和安全性需求。与现有工作进行对比,结果表明,SM2-TM方案在计算开销上有一定优势,同时在公/私钥和签名的存储开销上具有明显优势,能够有效地降低存储开销。

1 相关工作

1.1 基于SM2的数字签名

SM2是中国国家密码局发布的国家密码算法标准[10],与传统的RSA算法相比,SM2在密钥生成、签名和验证过程中表现出更高的运算效率。此外,SM2还具有“本地化”的特点,因此,更适合在政府、企业、教育等领域使用,以提升其对信息安全系统的自主可控能力。基于此,已有许多基于SM2的方案被提出。文献[16]在双环技术的基础上,提出一个基于SM2的环签名方案,并证明了其具有不可伪造性和匿名性。文献[17]提出了2个分别针对SM2的基于乘积和基于求和的秘密共享方案,但其会因本地化存储私钥而导致私钥泄露。文献[18]提出了一个基于标准SM2数字签名的、可证明安全的多签名方案,并在双射随机预言机模型下证明该方案具有不可伪造性。文献[19]提出了一种基于SM2的高效盲签名方案,但没有对所提不可伪造性进行正式的证明。文献[20]提出了一个不使用双线性对的、基于SM2的无证书签名方案,但其采用零知识证明来验证用户公钥,会导致公钥长度增加并引入额外的计算开销[21]。文献[22]提出了一种基于SM2的无证书环签密方案,实现了有条件的隐私保护。文献[23]提出了一种基于SM2的连续抗泄漏方案,即使部分密钥在侧信道攻击下连续泄露,攻击者也不能伪造签名,保证了电子数据的安全。文献[24]提出了一种基于SM2的、门限密钥分散的签名方案,满足了用户在移动终端进行电子签名的需求,同时利用门限签名来保证数据的完整性。

1.2 签名场景的应用

现有数字签名技术已有许多广泛的应用。在电子政务中,文献[1]提出了一种基于PKI的电子身份认证技术,有效地保证了电子政务的安全性。在医疗保健中,文献[2]提出了一种可净化签名方案,该方案中的签名者不仅能决定哪些消息块可以修改,还可以决定可修改块的最大值和修改的有效期。在此基础上,文献[2]还提出了一个电子医疗记录共享方案,通过上述的可净化签名,保证了医疗数据共享的安全性。文献[3]提出了一种可净化签名的安全电子医疗记录共享方案,医生可以指定患者可修改的字段,使得患者无须与医生进行交互,即可将原始签名转换为修改医疗记录后的新签名。文献[4]提出了一种基于SM2的密钥分割和协作签名技术,为用户提供了更加优质便捷的医疗安全服务。在数字交易中,文献[5]提出了一种可链接环签名方案,利用签名的可链接性,判断两个签名是否由同一个签名者产生,从而解决了电子现金交易中的双花问题,保证了交易的匿名性。文献[6]提出了一个基于代理盲签名的电子现金支付系统,解决了现有方案不满足交易可变的问题。在电子合同中,文献[7]使用双线性对,提出了一个基于身份的电子合同签署协议,其所有签署方都用自己的身份作为公钥,从而

简化了证书管理。文献[8]在 Waters 签名方案的基础上,提出了一种带有免责合约的签名方案,其具有可证明安全性和盲性,但其面临着密钥托管问题。文献[9]提出了一种多方合同签署的高效无证书合并签名方案,提高了签名者的签名验证效率,并在随机预言机模型下证明了其满足不可伪造性。然而,文献[13]认为文献[9]方案不能抵抗公钥替换攻击和内部签名者的联合攻击,其提出了一种改进的无证书聚合签名方案。文献[14]提出了一个基于多元公钥密码体制的无证书广播多重签名方案,避免了基于 ID 的公钥密码体制中的密钥托管问题和传统公钥基础设施中的证书管理问题,并在随机预言机模型下证明了其不可伪造性。文献[15]提出了一个低复杂度的、适合在电子合同中使用的无证书签密方案。

2 预备知识

本章主要介绍一些预备知识,包括 Shamir 门限秘密共享、变色龙哈希和椭圆曲线上的离散对数问题。

2.1 Shamir 门限秘密共享

Shamir 的(t,n)-门限秘密共享方案[25]支持在 n 个用户 id_i 中共享秘密值ξ,使得对于任意大于或等于 t 个用户的群体即可恢复秘密值ξ。该方案由以下两个算法组成:

1)$\mathrm{Gen}(p,\xi)\rightarrow\{(\mathrm{id}_i,\xi_i)\}_{i=1}^{n}$。输入一个秘密值$\xi>0$和一个素数$p>\max(\xi,n)$,选取$t-1$个系数$a_1,\cdots,a_{t-1}\in_R\mathbb{Z}_p$并定义一个在$\mathbb{Z}_p$上的$t-1$次多项式$f(x)=\xi+\sum_{i=0}^{t-1}a_i x^i$。然后,计算$\xi_i=f(\mathrm{id}_i)$并将其通过安全信道发送给用户$\mathrm{id}_i$。最后输出$\{(\mathrm{id}_1,\xi_1),\cdots,(\mathrm{id}_n,\xi_n)\}$。

2)$\mathrm{Reconstruct}\{(\mathrm{id}_i,\xi_i)\}_{i=1}^{n}\rightarrow\xi$。输入任意$t$个用户的$(x_i,y_i)\leftarrow(\mathrm{id}_i,\xi_i)$,然后通过拉格朗日插值公式计算出$f(x)$的系数$a_i$,最后通过下面的公式恢复秘密值$\xi=f(0)$:

$$f(x)=\sum_{i=0}^{t-1}y_i\prod_{t}\frac{x-x_j}{x_i-x_j} \tag{1}$$

2.2 变色龙哈希

哈希函数是一个可将任意长度的字符串映射成固定长度字符串的函数,具有单向性和抗碰撞性(包括弱抗碰撞性和强抗碰撞性),而变色龙哈希函数[26]是一种带陷门的哈希函数,具有陷门的用户可以轻易找到一组碰撞。变色龙哈希函数由以下 5 个算法组成:

1)$\mathrm{Setup}(\lambda)\rightarrow\mathrm{params}$。输入安全参数$\lambda$,输出系统公开参数 params。

2)$\mathrm{KeyGen}(\mathrm{params})\rightarrow(\mathrm{pk},\mathrm{tk})$。输入系统公开参数 params,输出公钥和陷门密钥(pk,tk)。

3)$\mathrm{Hash}(\mathrm{pk},m,r)\rightarrow c_h$。输入公钥 pk、消息$m$以及随机数$r$,输出变色龙哈希值$c_h$。

4)$\mathrm{Verify}(\mathrm{pk},(m,r,c_h))\rightarrow\{0,1\}$。输入公钥 pk 和一个三元组$(m,r,c_h)$,如果$(r,c_h)$是由消息$m$生成正确的哈希值,则输出 1,否则输出 0。

5)$\mathrm{Forge}(\mathrm{tk},(m,r,c_h),m')\rightarrow r'$。输入陷门密钥 tk、三元组$(m,r,c_h)$以及新的消息$m'$,输出一个新的随机数$r'$。

2.3 椭圆曲线上的离散对数问题

设$E(F_p)$是F_p上的椭圆曲线,选择椭圆曲线$E(F_p)$上阶为p的两个点Q、Q_1。如果存在一个正整数$a\in\mathbb{Z}_p^*$,使得$Q_1=a\cdot Q$成立,则从Q和Q_1中获得a的值称为椭圆曲线离散对数问题(ECDLP)[27]。

3 方案模型

3.1 系统模型

SM2-TM 方案的系统模型主要包括 KGC、签名方(S_i)和验证方(V_i),具体如图 1 所示。方案的符号描述如表 1 所示。

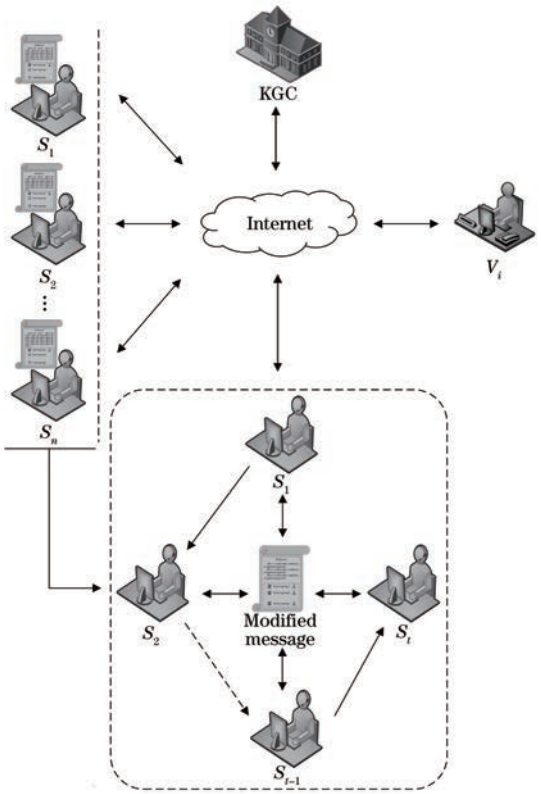

图 1 系统模型

Fig.1 System model

表 1 符号描述

Table 1 Symbols description

Symbol	Description
λ	安全参数
$\mathbb{G}, P$	循环群,生成元
$H_1(\cdot)$	哈希函数
pp	系统参数
id_i	身份标识
ξ_i	部分私钥
pid_i	匿名
$\mathrm{pk}_i, \mathrm{sk}_i$	公钥,私钥
I_i	可修改消息索引集
t_i	时间戳

1)S_i。S_i 主要对消息进行签名。此外,在需要对消息进行细微修改时,S_i 可作为群体数量为 t 中的一个参与成员,在不改变原来消息签名的情况下对消息进行细微修改。此外,各个签名方之间都是独立的、非勾结的实体。

2)V_i。V_i 主要对消息签名进行验证。

3)KGC。KGC 是一个拥有足够存储空间和计算能力的可信机构,主要为签名方和验证方生成公共参数和用于消息签名的部分私钥。

3.2 形式化定义

SM2-TM 方案由 5 个算法组成,具体的形式化定义如下:

1)$\mathrm{Setup}(\lambda)\rightarrow\{\mathrm{pp},\mathrm{msk},\mathrm{tk}\}$。由 KGC 执行系统初始化算法。输入安全参数 λ,输出系统公开参数 pp、系统私钥 msk 和追踪密钥 tk。

2)$\mathrm{KeyGen}(\mathrm{pp},\mathrm{id}_i)\rightarrow\{(\mathrm{pk}_i,\mathrm{sk}_i),\mathrm{pid}_i\}$。由 KGC 和实体 E_i(包括签名方和验证方)交互执行密钥生成算法。输入系统公开参数 pp、E_i 的身份标识 id_i,输出其相应的公/私钥对 $(\mathrm{pk}_i,\mathrm{sk}_i)$ 和匿名 pid_i。

3)$\mathrm{Sign}(\mathrm{pp},\mathrm{pid}_i,\mathrm{sk}_i,m_i)\rightarrow\sigma_i$。由签名方 S_i 执行签名算法。输入系统公开参数 pp、匿名 pid_i、签名私钥 sk_i 和消息 m_i,输出消息 m_i 的签名 $\sigma_i=(k_i,s_i)$。

4)$\mathrm{Verify}(m_i,\sigma_i,\mathrm{pk}_i)\rightarrow(0,1)$。由验证方 V_i 执行验证算法。输入消息 m_i、签名 σ_i 和验证公钥 pk_i。如果签名 σ_i 通过验证算法,输出 1;否则,输出 0。

5)$\mathrm{Modify}(\mathrm{pp},m'_i,\{(x_i,\xi_i)\}_{i=1}^{t})\rightarrow\sigma'_i$。由 t 个签名方协同执行消息修改算法。输入系统公开参数 pp、修改后的消息 m'_i 和 $\{(x_i,\xi_i)\}_{i=1}^{t}$,输出一个使得 $\beta'_i=\beta_i$ 成立的哈希碰撞和签名 σ'_i。

3.3 隐私和安全性需求

SM2-TM 方案应当满足以下隐私和安全性需求:

1)匿名性。任意实体的身份应当是匿名的,但在一定条件下可被可信机构追踪。

2)不可链接性。任意敌手都无法通过公开信道中的两个签名来判断其是否由同一个签名方生成。

3)不可伪造性。任意签名方产生的签名是不可伪造的,即任意敌手无法通过伪造一个有效的签名来通过验证算法。

4)不变性。对任意 t 个签名方来说,其协同对细微修改后的消息所生成的签名应当与其未被修改的消息签名保持一致。应当注意的是,这与上述提及的不可伪造性是不一样的,签名的“不变性”是指 t 个签名方通过预留的门限来构造一个哈希碰撞以实现“不变性”,而不可伪造性是敌手能通过一个概率多项式时间(PPT)算法,以不可忽略的优势来赢得其与挑战者之间定义的一个游戏(具体见定理 1),这等价于挑战者能以不可忽略的优势解决 ECDLP 问题。

5)抗重放攻击。所提方案能抵抗敌手短时间内的消息重放攻击。

4 本文方案设计

SM2-TM 方案主要包含系统初始化、密钥生成、消息签名、消息验证和消息修改等步骤。

4.1 系统初始化

给定一个安全参数 1^{λ},KGC 在有限域 F_p 上定义一条椭圆曲线 $E(F_p)$,其方程为 $y^2=x^3+ax+b$。令 $\mathbb{G}$ 为满足方程的点构成的循环群,其阶为 η,$P=(x_P,y_P)$ 为生成元。然后,选择一个安全的哈希函数 $H_1:\{0,1\}^*\rightarrow\mathbb{Z}_p^*$,紧接着选择 msk、tk、$\xi\in_R\mathbb{Z}_p^*$,计算 $\mathrm{mpk}=\mathrm{msk}\cdot P$、$h=\xi\cdot P$ 和 $d=\mathrm{tk}\cdot P$。进一步地,选择 $t-1$ 个数 $a_1,\cdots,a_{t-1}\in_R\mathbb{Z}_p$。最后,输出系统公开参数 $\mathrm{pp}=\{p,E(F_p),\mathbb{G},P,\eta,\mathrm{mpk},h,d,a_1,\cdots,a_{t-1},H_1\}$。

4.2 密钥生成

任意实体 E_i 都有其唯一的身份标识 id_i,它们都要向 KGC 进行注册以生成其部分私钥 ξ_i 和匿名 pid_i,最后由 E_i 生成其公/私钥对 $(\mathrm{pk}_i,\mathrm{sk}_i)$。

1)KGC 构造一个多项式 $f(x)=\xi+\sum_{v=0}^{t-1}a_v x^v$,计算 $\{x_i,\xi_i,\mathrm{pid}_i\}$,然后将 $\langle\mathrm{pid}_i,\xi_i\rangle$ 通过安全信道发送给 E_i:

$$\begin{cases} x_i = H_1(\mathrm{id}_i) \\ \xi_i = f(x_i) \\ \mathrm{pid}_i = \mathrm{id}_i \oplus H_1(\mathrm{msk} \cdot d) \end{cases} \tag{2}$$

2)接收到$\langle \mathrm{pid}_i, \xi_i \rangle$后，$E_i$ 选择 $c_i \in_R \mathbb{Z}_p^*$，计算 $\mathrm{sk}_i = c_i + \xi_i$，$\mathrm{pk}_i = \mathrm{sk}_i \cdot P$。最后，输出其公/私钥对$(\mathrm{pk}_i, \mathrm{sk}_i)$和匿名 pid_i。

4.3 消息签名

S_i 执行下列步骤对消息 $m_i = \{m_{i1}, \cdots, m_{in}\}$进行签名，同时设定可修改消息索引集 $I_i \in [1, n]$，其中 $m_{il} \in \{0,1\}$，$l \in [1, n]$。

1)计算 $\alpha_i = H_1(\mathrm{pid}_i \| m_i)$和 $\beta_i = H_1(\alpha_i \cdot P + h)$。

2)选择 $r_i \in \mathbb{Z}_p^*$，计算$\{Q_i, k_i, s_i\}$。若 $s_i = 0$、$k_i = 0$ 或 $k_i + r_i = \eta$，则重新选择数 r_i；

$$\begin{cases} Q_i = r_i \cdot P = (Q_{x_i}, Q_{y_i}) \\ k_i = (\beta_i + Q_{x_i}) \bmod (\eta) \\ s_i = (1 + \mathrm{sk}_i)^{-1}(r_i - k_i \cdot \mathrm{sk}_i) \bmod (\eta) \end{cases} \tag{3}$$

3)输出消息 m_i 的签名 $\sigma_i = \{k_i, s_i\}$，然后将$\langle \mathrm{pid}_i, m_i, \alpha_i, \beta_i, Q_i, \sigma_i, I_i, t_i \rangle$发送给 V_i 进行验证。

4.4 消息验证

V_i 可对任意 S_i 生成的消息签名进行验证。具体地，接收到$\langle \mathrm{pid}_i, m_i, \alpha_i, \beta_i, Q_i, \sigma_i, I_i, t_i \rangle$的 V_i 通过执行下列步骤，对消息签名进行验证：

1)验证 t_i 的鲜活性。若 t_i 不鲜活，则签名无效；否则继续下列操作。

2)检查 $k_i, s_i \overset{?}{\in} \mathbb{Z}_p^*$。若不成立，则签名无效；否则继续下列操作。

3)计算$\{\alpha_i', \beta_i', u_i\}$。若 $u_i = 0$，则签名无效；否则继续下列操作。

$$\begin{cases} \alpha_i' = H_1(\mathrm{pid}_i \| m_i) \\ \beta_i' = H_1(\alpha_i' \cdot P + h) \\ u_i = (k_i + s_i) \bmod (\eta) \end{cases} \tag{4}$$

4)计算$\{Q_i', k_i'\}$。若 $k_i' = k_i$，则签名验证通过；否则，签名无效。

$$\begin{cases} Q_i' = s_i \cdot P + u_i \cdot \mathrm{pk}_i = (Q_{x_i}', Q_{y_i}') \\ k_i' = (\beta_i' + Q_{x_i}') \bmod (\eta) \end{cases} \tag{5}$$

正确性：已知对于同一个$\{\mathrm{pid}_i, m_i\}$，有 $\alpha_i = \alpha_i'$，$\beta_i' = \beta_i$，则计算 $Q_i' \overset{?}{=} Q_i$ 和 $k_i' \overset{?}{=} k_i$。

$$\begin{aligned} Q_i' &= s_i \cdot P + u_i \cdot \mathrm{pk}_i = \\ & s_i \cdot P + (k_i + s_i) \cdot \mathrm{sk}_i \cdot P = \\ & (1 + \mathrm{sk}_i) \cdot s_i \cdot P + k_i \cdot \mathrm{sk}_i \cdot P = \\ & (1 + \mathrm{sk}_i) \cdot (1 + \mathrm{sk}_i)^{-1} \cdot (r_i - k_i \cdot \mathrm{sk}_i) \cdot P + k_i \cdot \mathrm{sk}_i \cdot P = \\ & (r_i - k_i \cdot \mathrm{sk}_i) \cdot P + k_i \cdot \mathrm{sk}_i \cdot P = r_i \cdot P = Q_i \end{aligned} \tag{6}$$

$$\begin{aligned} k_i' &= (\beta_i' + Q_{x_i}') \bmod (\eta) = \\ & (\beta_i + Q_{x_i}) \bmod (\eta) = k_i \end{aligned} \tag{7}$$

4.5 消息修改

在消息修改阶段，任意 t 个同意修改消息的 S_i 可通过构建一个哈希碰撞，在不改变消息签名的情况下实现对消息内容的细微修改。具体地，本文根据可修改消息索引集 $I_i \in [1, n]$来生成修改后的消息 $m_i' = \{m_{i1}', m_{i2}', \cdots, m_{in}'\}$，其中，$m_{il}' \in \{0,1\}$，$l \in [1, n]$。

1)对于任意 $l \in \mathrm{MS} = [1, n]$，定义一个消息修改规则，满足 $\mathrm{MS}_1 \cup \mathrm{MS}_2 = \mathrm{MS}$、$\mathrm{MS}_1 \cap \mathrm{MS}_2 = \varnothing$和对于任意的 $l \in \mathrm{MS}$ 有 $m_{il}' - m_{il} = \begin{cases} 1, & l \in \mathrm{MS}_1 \\ -1, & l \in \mathrm{MS}_2 \end{cases}$。

$$\begin{cases} \mathrm{MS}_1 = \{l \mid m_{il} = 0,\ m_{il}' = 1\} \\ \mathrm{MS}_2 = \{l \mid m_{il} = 1,\ m_{il}' = 0\} \end{cases} \tag{8}$$

2)任意 t 个签名方 S_i 都有(x_i, ξ_i)，它们协同计算 $f(x) = \sum_{i=0}^{t-1} \xi_i \prod_t \frac{x - x_j}{x_i - x_j}$，然后计算 $f(0)$得到 ξ。

3)通过计算$\{\alpha_i', h'\}$，推断出 $\beta_i' = \beta_i$。

$$\begin{cases} \alpha_i' = H_1(\mathrm{pid}_i \| m_i') \\ h' = \xi' \cdot P \leftarrow \xi \cdot P + (\alpha_i - \alpha_i') \cdot P \end{cases} \tag{9}$$

4)最后，输出消息 m_i'及其签名 $\sigma_i' = \sigma_i$。

正确性：通过如下计算，可推断出 $\beta_i' = \beta_i$。

$$\begin{aligned} & \beta_i' = H_1(\alpha_i' \cdot P + h') = H_1(\alpha_i' \cdot P + \xi' \cdot P) = \\ & H_1(\alpha_i' \cdot P + \xi \cdot P + (\alpha_i - \alpha_i') \cdot P) = \\ & H_1(\xi \cdot P + \alpha_i \cdot P) = H_1(h + \alpha_i \cdot P) = \beta_i \end{aligned} \tag{10}$$

5 隐私和安全性分析

本节首先证明所提方案在随机预言机模型下是不可伪造的。具体地，假设 PPT 敌手 $\mathcal{A}$ 具有一定的计算能力，并且可进行有限次的查询。若对于任意的 $\mathcal{A}$ 能以可忽略的优势赢得游戏 $\mathrm{Game}_{\mathcal{A}}$，则该方案在选择消息攻击下满足存在性不可伪造(EUF-CMA)。

定理 1 若 PPT 的挑战者 $\mathcal{C}$ 能以不可忽略的优势解决 ECDLP 问题，则存在一个敌手 $\mathcal{A}$ 能以不可忽略的优势在游戏 $\mathrm{Game}_{\mathcal{A}}$ 中获胜，并通过构造一个算法来打破所提方案的 EUF-CMA 安全性。

证明 给出一个 ECDLP 问题实例(Q, Q_1)，挑战者 $\mathcal{C}$ 的目的是找到一个数 $a \in \mathbb{Z}_p^*$，使得 $Q_1 = a \cdot Q$ 成立。为了解决这个问题，$\mathcal{C}$ 与 $\mathcal{A}$ 的交互如下：

1)初始化阶段。$\mathcal{C}$ 执行系统初始化算法，生成系统公开参数 $\mathrm{pp} = \{E(F_p), p, P, \eta, \mathrm{mpk}, h, H_1\}$，并将其发送给 $\mathcal{A}$。

2)查询阶段。$\mathcal{A}$ 可以自适应地执行查询操作。同时,$\mathcal{C}$ 管理 3 个初始时为空集的列表 L_1、L_2、L_3,其中,L_1、L_2 和 L_3 分别记录 H_1 查询(Q_{H_1})、私钥提取查询(Q_{esk})和公钥提取查询(Q_{epk})。

(1)H_1 查询(Q_{H_1})。$\mathcal{A}$ 使用存储格式为(id_i,x_i)的元组来创造一个列表 L_1。在收到 $\mathcal{A}$ 关于 id_i 的 Q_{H_1} 后,$\mathcal{C}$ 检查 $\text{id}_i \overset{?}{\in} L_1$ 是否成立。如果 $\text{id}_i \in L_1$ 成立,则将 x_i 返回给 $\mathcal{A}$;否则,重新选择 $x_i \in_R \mathbb{Z}_p^*$ 作为响应,并且将(id_i,x_i)添加到列表 L_1 中。

(2)私钥提取查询(Q_{esk})。$\mathcal{A}$ 使用存储格式为(L_1,id_i,sk_i)的元组来创造一个列表 L_2。在收到 $\mathcal{A}$ 关于 id_i 的 Q_{esk} 后,$\mathcal{C}$ 检查 $\text{id}_i \overset{?}{\in} L_2$ 是否成立。如果 $\text{id}_i \in L_2$ 成立,则返回 sk_i 给 $\mathcal{A}$;否则,重新选择 $\text{sk}_i \in_R \mathbb{Z}_p^*$,然后将 sk_i 发送给 $\mathcal{A}$ 作为响应,并且将(id_i,sk_i)添加到列表 L_2 中。

(3)公钥提取查询(Q_{epk})。$\mathcal{A}$ 使用存储格式为(L_2,id_i,sk_i,pk_i)的元组来创造一个列表 L_3。在收到 $\mathcal{A}$ 关于 id_i 的 Q_{epk} 后,$\mathcal{C}$ 检查 $\text{id}_i \overset{?}{\in} L_3$ 是否成立。如果 $\text{id}_i \in L_3$ 成立,则将 pk_i 返回给 $\mathcal{A}$;否则,重新选择 $\text{sk}_i \in_R \mathbb{Z}_p^*$,计算 $\text{pk}_i = \text{sk}_i \cdot P$,然后将 pk_i 发送给 $\mathcal{A}$ 作为响应,并且将(id_i,sk_i,pk_i)添加到列表 L_3 中。

(4)公钥替换查询(Q_{rpk})。收到 $\mathcal{A}$ 关于(id_i,pk_i)的 Q_{rpk} 后,$\mathcal{C}$ 将 pk_i 替换为 pk'_i,并将(id_i,pk'_i)添加到 L_3 中。

(5)签名查询(Q_{sig})。在收到 $\mathcal{A}$ 关于{id_i,m_i}的 Q_{sig} 后,$\mathcal{C}$ 检查 $\text{id}_i \overset{?}{\in} L_3$ 是否成立。如果 $\text{id}_i \in L_3$ 成立,则用 $\text{sk}_i \in L_3$ 生成 m_i 的签名 σ_i,并把 σ_i 返回给 $\mathcal{A}$;否则,$\mathcal{C}$ 选择 r_i,$\text{sk}_i \in_R \mathbb{Z}_p^*$,然后计算{$\alpha_i$,$\beta_i$,$Q_i$,$k_i$,$s_i$}。最后,将 $\sigma_i = \{k_i, s_i\}$ 发送给 $\mathcal{A}$ 作为响应。

$$\begin{cases} \alpha_i = H_1(\text{pid}_i \parallel m_i) \\ \beta_i = H_1(\alpha_i \cdot P + h) \\ Q_i = r_i \cdot P = (Q_{x_i}, Q_{y_i}) \\ k_i = (\beta_i + Q_{x_i}) \bmod (\eta) \\ s_i = (1+\text{sk}_i)^{-1}(r_i - k_i \cdot \text{sk}_i) \bmod (\eta) \end{cases} \tag{11}$$

3)伪造阶段。在这些查询之后,假设 $\mathcal{A}$ 能以不可忽略的概率赢得游戏 $\text{Game}_{\mathcal{A}}$,这意味着它可以通过{$\text{id}_i^*$,$m_i^*$}伪造出两个合法的签名 σ_i^* 和 σ_i^{**}。根据分叉引理[28],$\mathcal{A}$ 可使用不同的哈希值重复上述查询,生成以下两个签名对(k_i^*,s_i^*)和(k_i^{**},s_i^{**})。

$$k_i^* = \beta'_i + s_i^* \cdot x_P + u_i^* \cdot \text{sk}_i \cdot x_P \tag{12}$$

$$k_i^{**} = \beta'_i + s_i^{**} \cdot x_P + u_i^{**} \cdot \text{sk}_i \cdot x_P \tag{13}$$

令 $k_i^* - k_i^{**} = z_1$、$s_i^* - s_i^{**} = z_2$ 以及 $u_i^* - u_i^{**} = z_3$,其中,z_1,z_2,$z_3 \in \mathbb{Z}_p^*$。然后,由式(12)和式(13),可以得到以下等式:

$$z_1 = z_2 \cdot x_P + z_3 \cdot \text{sk}_i \cdot x_P \tag{14}$$

$\mathcal{C}$ 可以输出 sk_i 作为 ECDLP 的结果,这与 ECDLP 是困难问题相矛盾,因此,本文方案可以在选择消息攻击下满足存在性不可伪造。

接着,本文还分析了所提方案满足的下列隐私和安全性需求:

1)匿名性。所提方案在消息签名的过程中,使用假名 $\text{pid}_i = \text{id}_i \oplus H_1(\text{msk} \cdot d)$ 来与验证方进行通信,保证了签名方的匿名性。

2)不可链接性。在消息签名过程中,任意一个签名的生成都与随机数 r_i 有关。若敌手能判断任意两个签名是否由同一个签名方生成,这等价于他能从签名中恢复 r_i,并解决 ECDLP 问题。这与 ECDLP 问题是困难的相矛盾。

3)不变性。假设签名方 S_i 对消息 m_i 的签名为 $\sigma_i = \{k_i, s_i\}$,其修改后的消息 m'_i 的签名为 $\sigma'_i = \{k'_i, s'_i\}$。从签名的构造可知,在保持 S_i 的{pid_i,r_i,sk_i}不变的情况下使得 $\sigma_i = \sigma'_i$ 的前提是 $\beta'_i = \beta_i$。从式(10)可知,任意 t 个签名方 S_i 可通过构建一个哈希碰撞来使得 $\beta'_i = \beta_i$,这意味着所提方案在不改变消息签名的情况下支持对消息内容进行修改,即在一定程度上保证了签名的不变性。

4)可追踪性。当需要揭示某个签名是由哪个签名方签名时,KGC 可利用其签名中的 pid_i 来计算 $\text{id}_i = \text{pid}_i \oplus H_1(\text{mpk} \cdot \text{tk})$,然后获取其真实身份。

5)抗重放攻击。所提方案在消息签名和验证阶段都嵌入了时间戳 t_i,能有效抵抗重放攻击。

6)密钥托管。在所提方案中,实体的完整密钥 sk_i 由{c_i,ξ_i}两部分组成,其中,ξ_i 由 KGC 生成,c_i 由实体生成。即使攻击者能攻破 KGC 并从中得到 ξ_i,其也难以从 pk_i 中恢复 sk_i。

6 性能分析

本文的模拟实验采用了两种环境,其中环境 1 为运行 Ubuntu 20.04 LTS 系统的计算机(Intel® Core™ i5-1035G1 CPU @1.00 GHz 1.19 GHz,RAM 16.0 GB),环境 2 为运行 openKylin 2.0 系统的计算机(AMD Ryzen 7 8845H w/ Radeon 780M Graphics 3.80 GHz,RAM 8.0 GB)。为了分析各项操作的计算开销,本文使用 Python 的 Hashlib、Pymcl 等模块进行相关计算。此外,本文还对每个常见的操作(如哈希、点乘等)执行 1 000 次取平均值来评估其运行效率,结果如表 2 所示。具体地,本

表 2 运算符号和时间

Table 2 Operational symbols and time 单位:ms

Symbol	Description	Environment 1	Environment 2
T_{H}	H 运算的时间	0.002 71	0.007 47
T_{az}	$\mathbb{Z}_p^*$ 中元素加法运算的时间	0.001 01	0.002 75
T_{mz}	$\mathbb{Z}_p^*$ 中元素乘法运算的时间	0.002 57	0.004 01
T_{ag}	$\mathbb{G}$ 中元素加法运算的时间	0.000 19	0.004 04
T_{mg}	$\mathbb{G}$ 中元素乘法运算的时间	0.000 21	0.007 03
T_{m}	点乘运算的时间	0.000 77	0.000 76
T_{e}	双线性运算的时间	0.122 41	0.144 61
T_{exp}	指数运算的时间	0.001 08	0.003 16

文首先理论分析和对比 SM2-TM 方案与 CBS 方案[20]、CLS 方案[20]、ICS 方案[21]、ASCS. CS 方案[29]、ASCS. SS 方案[29]在消息签名和验证中的计算开销以及公/私钥和签名的存储开销。随后,本文在两种不同的模拟实验环境中测试它们在消息签名和验证中的计算开销。

6.1 计算和存储开销的理论分析

本节主要从理论上分析和对比本文 SM2-TM 方案与 CBS 方案[20]、CLS 方案[20]、ICS 方案[21]、ASCS. CS 方案[29]、ASCS. SS 方案[29]在消息签名和验证中的计算开销以及公/私钥和签名的存储开销,详细对比结果如表 3 所示。

表 3 计算和存储开销的对比结果

Table 3 Comparison results of computation and storage costs

Scheme	Computational cost		Storage cost		
	Message signature	Signature verification	Public key	Secret key	Signature
CBS	$T_{\mathrm{H}}+2T_{\mathrm{az}}+2T_{\mathrm{mz}}+T_{\mathrm{m}}$	$2T_{\mathrm{H}}+4T_{\mathrm{az}}+2T_{\mathrm{ag}}+4T_{\mathrm{m}}$	$\lvert\mathbb{G}\rvert$	$\lvert\mathbb{Z}_p\rvert$	$4\lvert\mathbb{Z}_p\rvert$
CLS	$T_{\mathrm{H}}+3T_{\mathrm{az}}+2T_{\mathrm{mz}}+T_{\mathrm{m}}$	$2T_{\mathrm{H}}+4T_{\mathrm{az}}+2T_{\mathrm{ag}}+4T_{\mathrm{m}}$	$3\lvert\mathbb{G}\rvert$	$4\lvert\mathbb{Z}_p\rvert+\lvert\mathbb{G}\rvert$	$4\lvert\mathbb{Z}_p\rvert$
ICS	$T_{\mathrm{H}}+2T_{\mathrm{az}}+3T_{\mathrm{mz}}+T_{\mathrm{m}}$	$2T_{\mathrm{H}}+4T_{\mathrm{az}}+2T_{\mathrm{ag}}+4T_{\mathrm{m}}+2T_{\mathrm{e}}$	$3\lvert\mathbb{G}\rvert$	$4\lvert\mathbb{Z}_p\rvert+\lvert\mathbb{G}\rvert$	$4\lvert\mathbb{Z}_p\rvert$
ASCS. CS	$2T_{\mathrm{H}}+T_{\mathrm{mg}}+4T_{\mathrm{exp}}$	$2T_{\mathrm{H}}+T_{\mathrm{mg}}+T_{\mathrm{exp}}+2T_{\mathrm{e}}$	$\lvert\mathbb{G}\rvert$	$\lvert\mathbb{Z}_p\rvert$	$3\lvert\mathbb{G}\rvert$
ASCS. SS	$(\lvert I\rvert+2)T_{\mathrm{H}}+(\lvert I\rvert+1)T_{\mathrm{mg}}+(\lvert I\rvert+4)T_{\mathrm{exp}}$	$(\lvert I\rvert+2)T_{\mathrm{H}}+(\lvert I\rvert+1)T_{\mathrm{mg}}+(3\lvert I\rvert+1)T_{\mathrm{exp}}+2T_{\mathrm{e}}$	$\lvert\mathbb{G}\rvert$	$\lvert\mathbb{Z}_p\rvert$	$(2\lvert I\rvert+3)\lvert\mathbb{G}\rvert$
本文方案	$2T_{\mathrm{H}}+2T_{\mathrm{az}}+2T_{\mathrm{mz}}+T_{\mathrm{ag}}+2T_{\mathrm{m}}$	$2T_{\mathrm{H}}+2T_{\mathrm{az}}+2T_{\mathrm{ag}}+3T_{\mathrm{m}}$	$\lvert\mathbb{G}\rvert$	$\lvert\mathbb{Z}_p\rvert$	$2\lvert\mathbb{Z}_p\rvert$

1)计算开销。

在本文 SM2-TM 方案中,签名方通过 2 个 H 运算、2 个 $\mathbb{Z}_p$ 中的加法运算、2 个 $\mathbb{Z}_p$ 中的乘法运算、1 个 $\mathbb{G}$ 中的加法运算和 2 个点乘运算来生成消息签名,其计算开销为 $2T_{\mathrm{H}}+2T_{\mathrm{az}}+2T_{\mathrm{mz}}+T_{\mathrm{ag}}+2T_{\mathrm{m}}$;验证方通过 2 个 H 运算、2 个 $\mathbb{Z}_p$ 中的加法运算、2 个 $\mathbb{G}$ 中的加法运算和 3 个点乘运算来进行签名验证,其计算开销为 $2T_{\mathrm{H}}+2T_{\mathrm{az}}+2T_{\mathrm{ag}}+3T_{\mathrm{m}}$。

在 CBS 方案[20]中,签名方通过 1 个 H 运算、2 个 $\mathbb{Z}_p$ 中的加法运算、2 个 $\mathbb{Z}_p$ 中的乘法运算和 1 个点乘运算来生成消息签名,其计算开销为 $T_{\mathrm{H}}+2T_{\mathrm{az}}+2T_{\mathrm{mz}}+T_{\mathrm{m}}$;验证方通过 2 个 H 运算、4 个 $\mathbb{Z}_p$ 中的加法运算、2 个 $\mathbb{G}$ 中的加法运算和 4 个点乘运算来进行签名验证,其计算开销为 $2T_{\mathrm{H}}+4T_{\mathrm{az}}+2T_{\mathrm{ag}}+4T_{\mathrm{m}}$。

在 CLS 方案[20]中,签名方通过 1 个 H 运算、3 个 $\mathbb{Z}_p$ 中的加法运算、2 个 $\mathbb{Z}_p$ 中的乘法运算和 1 个点乘运算来生成消息签名,其计算开销为 $T_{\mathrm{H}}+3T_{\mathrm{az}}+2T_{\mathrm{mz}}+T_{\mathrm{m}}$;验证方通过 2 个 H 运算、4 个 $\mathbb{Z}_p$ 中的加法运算、2 个 $\mathbb{G}$ 中的加法运算和 4 个点乘运算来进行签名验证,其计算开销为 $2T_{\mathrm{H}}+4T_{\mathrm{az}}+2T_{\mathrm{ag}}+4T_{\mathrm{m}}$。

在 ICS 方案[21]中,签名方通过 1 个 H 运算、2 个 $\mathbb{Z}_p$ 中的加法运算、3 个 $\mathbb{Z}_p$ 中的乘法运算和 1 个点乘运算来生成消息签名,其计算开销为 $T_{\mathrm{H}}+2T_{\mathrm{az}}+3T_{\mathrm{mz}}+T_{\mathrm{m}}$;验证方通过 2 个 H 运算、4 个 $\mathbb{Z}_p$ 中的加法运算、2 个 $\mathbb{G}$ 中的加法运算和 4 个点乘运算来进行签名验证,其计算开销为 $2T_{\mathrm{H}}+4T_{\mathrm{az}}+2T_{\mathrm{ag}}+4T_{\mathrm{m}}+2T_{\mathrm{e}}$。

在 ASCS. CS 方案[29]中,签名方通过 2 个 H 运算、1 个 $\mathbb{G}$ 中的乘法运算和 4 个指数运算来生成消息签名,其计算开销为 $2T_{\mathrm{H}}+T_{\mathrm{mg}}+4T_{\mathrm{exp}}$;验证方通过 2 个 H 运算、1 个 $\mathbb{G}$ 中的乘法运算、1 个指数运算和 2 个双线性运算来进行签名验证,其计算开销为 $2T_{\mathrm{H}}+T_{\mathrm{mg}}+T_{\mathrm{exp}}+2T_{\mathrm{e}}$。

在 ASCS. SS 方案[29]中,签名方通过$(\lvert I\rvert+2)$个 H 运算、$(\lvert I\rvert+1)$个 $\mathbb{G}$ 中的乘法运算和$(3\lvert I\rvert+4)$个指数运算来生成消息签名,其计算开销为$(\lvert I\rvert+2)T_{\mathrm{H}}+(\lvert I\rvert+1)T_{\mathrm{mg}}+(\lvert I\rvert+4)T_{\mathrm{exp}}$;验证方通过$(\lvert I\rvert+2)$个 H 运算、$(\lvert I\rvert+1)$个 $\mathbb{G}$ 中的乘法运算、$(3\lvert I\rvert+1)$个指数运算和 2 个双线性运算来进行签名验证,其计算开销为$(\lvert I\rvert+2)T_{\mathrm{H}}+$

$(|I|+1)T_{mg}+(3|I|+1)T_{exp}+2T_e$。

2)存储开销。

在本文 SM2-TM 方案中,其在密钥生成阶段生成的公/私钥对为(pk_i, sk_i),其中,$pk_i \in \mathbb{G}$,$sk_i \in \mathbb{Z}_p$,因此公钥 pk_i 的存储开销为$|\mathbb{G}|$,私钥 $sk_i \in \mathbb{Z}_p$ 的存储开销为$|\mathbb{Z}_p|$;在消息签名阶段,其生成的签名值为 $\sigma_i=\{k_i, s_i\}$,其中,$k_i, s_i \in \mathbb{Z}_p$,因此签名 σ_i 的存储开销为 $2|\mathbb{Z}_p|$。

在 CBS 方案[20]中,其在密钥生成阶段生成的公/私钥对为(pk_{ID}, sv_{ID}),其中,$pk_{ID} \in \mathbb{G}$,$sv_{ID} \in \mathbb{Z}_p$,因此公钥 pk_{ID} 的存储开销为$|\mathbb{G}|$,私钥 sv_{ID} 的存储开销为$|\mathbb{Z}_p|$;在消息签名阶段,其生成的签名值为 $\sigma_m=\{c_{ID}, r', s'\}$,其中,$c_{ID}=\{r, s\}$,$r, r', s, s' \in \mathbb{Z}_p$,因此签名 σ_m 的存储开销为 $4|\mathbb{Z}_p|$。

在 CLS 方案[20]中,其私钥由两部分组成,即 KGC 生成的部分私钥 $d_{ID}=\{r, s, x, ppk_{ID}\}$ 和签名方选择的秘密值 $sv_{ID}=\{x, y\}$,其中,$r, s, x, y \in \mathbb{Z}_p$,$ppk_{ID} \in \mathbb{G}$,故其私钥的存储开销为 $4|\mathbb{Z}_p|+|\mathbb{G}|$,其公钥为 $pk_{ID}=\{ppk_{ID}, [y]P, \pi\}$,其中,$ppk_{ID}$、$[y]P$、$\pi \in \mathbb{G}$,故其公钥的存储开销为 $3|\mathbb{G}|$;在消息签名阶段,其生成的签名值为 $\sigma_m=\{r, s, r', s'\}$,其中,$r, r', s, s' \in \mathbb{Z}_p$,因此签名 σ_m 的存储开销为 $4|\mathbb{Z}_p|$。类似地,在 ICS 方案[21]中,其私钥、公钥和签名的存储开销分别为 $4|\mathbb{Z}_p|+|\mathbb{G}|$、$3|\mathbb{G}|$ 和 $4|\mathbb{Z}_p|$。

在 ASCS.CS 方案[29]中,其在密钥生成阶段生成的公/私钥对为(spk, ssk),其中,$spk \in \mathbb{G}$,$ssk \in \mathbb{Z}_p$,故公钥 spk 的存储开销为$|\mathbb{G}|$,私钥 ssk 的存储开销为$|\mathbb{Z}_p|$;在消息签名阶段,其生成的签名值为 $\sigma=SIGN_{ssk}(h_c)\|\bar{r}$,其中,$SIGN_{ssk}(h_c)$ 使用文献[30]的签名方案,故有 $SIGN_{ssk}(h_c) \in \mathbb{G}$。而 $\bar{r} \in \mathbb{G}$,故签名 σ 的存储开销为 $3|\mathbb{G}|$。类似地,在 ASCS.SS 方案[29]中,其公钥、私钥和签名的存储开销分别为$|\mathbb{G}|$、$|\mathbb{Z}_p|$和$(2|I|+3)|\mathbb{G}|$。

6.2 模拟实验对比

本节在两种不同的模拟实验环境中测试方案在消息签名和验证中的计算开销,结果如图 2 和图 3 所示。结果显示,方案在不同环境中的性能表现各异,虽然在环境 2 中的计算开销略高于环境 1,但环境 2 在安全性和可控性方面具有显著优势。因此,尽管环境 2 在性能上稍逊,但提供了更高的安全性和灵活性。具体地,在消息签名中,本文 SM2-TM 方案在环境 1 和环境 2 中的计算开销都要优于 ASCS.SS 方案,其计算效率分别提高了 52.22%和 72.32%。与环境 2 中的 ASCS.CS 方案相比,本文 SM2-TM 方案的计算效率提高了 1.70%,但与环境 1 中的 ASCS.CS 方案相比,却降低了 43.82%。与 CBS 方案、CLS 方案、ICS 方案相比,本文 SM2-TM 方案在环境 1 和环境 2 中的计算效率都有所降低,在环境 1 中的效率分别降低 34.49%、22.83%和 8.33%,在环境 2 中的效率分别降低 56.41%、38.86%和 32.07%。在签名验证中,本文 SM2-TM 方案在环境 1 和环境 2 中的计算开销都要优于对比方案(CBS 方案、CLS 方案、ICS 方案、ASCS.CS 方案和 ASCS.SS 方案),其计算效率均有所提高,其中在环境 1 中的效率分别提高 21.59%、21.59%、96.07%、95.96%和 96.41%,在环境 2 中的效率分别提高 16.89%、16.89%、90.56%、90.10%和 92.91%。综上所述,本文 SM2-TM 方案在计算开销上有一定优势。

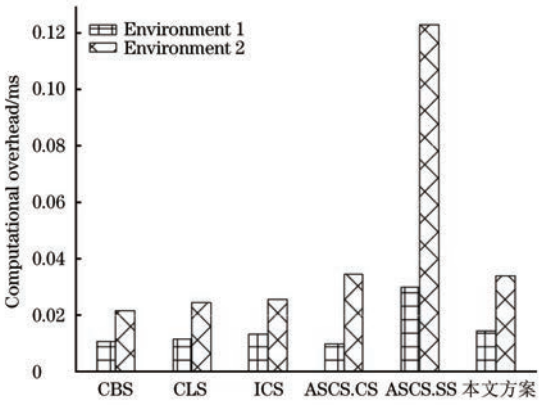

图 2 消息签名在不同环境下的计算开销

Fig.2 Computational overhead of message signature under different environments

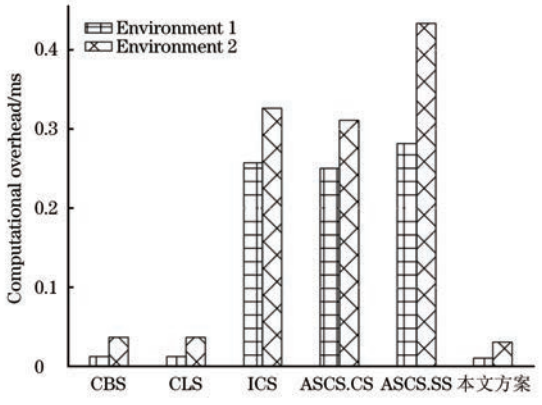

图 3 签名验证在不同环境下的计算开销

Fig.3 Computational overhead of signature verification under different environments

在存储开销中,本文令 $\mathbb{Z}_p$ 和 $\mathbb{G}$ 中的元素尺寸分别为 32 和 64 Byte,方案的对比结果如图 4 所示。显然,本文所提方案的存储开销均优于对比方案。

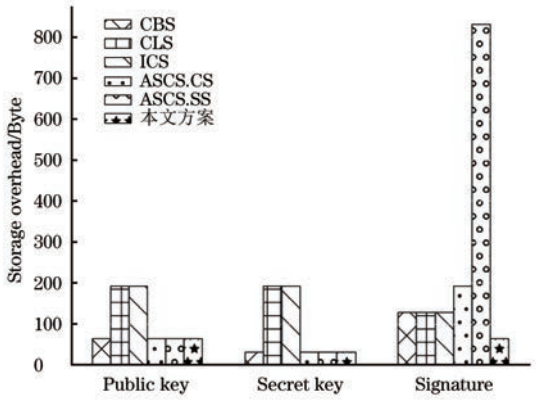

图 4 存储开销对比

Fig. 4 Storage overhead comparison

7 结束语

本文提出一个基于 SM2 的、支持门限修改签名消息的数字签名方案 SM2-TM。该方案通过 Shamir 门限秘密共享和变色龙哈希构建一个门限变色龙哈希，支持 t 个签名方协同构建一个哈希碰撞来实现更改消息内容但不更改其消息签名。此外，SM2-TM 方案还解决了复杂的证书存储和管理以及密钥托管问题。在隐私和安全性上，所提方案满足匿名性、不可链接性、不可伪造性、不变性、可追踪性和抗重放攻击。在性能上，SM2-TM 方案在计算和存储开销上有一定优势。

然而，本文方案在消息签名的计算效率上表现不足。此外，当前方案未考虑多个签名方相互勾结的情况。因此，接下来的工作将重点优化签名的计算效率，并开发能够抵御多个签名方相互勾结的方案。

参考文献

[1] 刘邦桂，曾思财. 电子政务中身份认证技术的研究与实现[J]. 软件工程，2021，24(11)：47-50.
LIU B G，ZENG S C. Research and implementation of identity authentication technology in e-government [J]. Software Engineer，2021，24(11)：47-50. (in Chinese)

[2] YANG Y，YI B X，ZHAN Y H，et al. Secure EHR sharing scheme based on limited sanitizable signature [EB/OL]. [2024-08-05]. https://link.springer.com/chapter/10.1007/978-981-19-8445-7_19.

[3] ZHAN Y H，YI B X，YANG Y，et al. A privilege-constrained sanitizable signature scheme for e-health systems[J]. Journal of Systems Architecture，2023，142：102939.

[4] 王浩，胡艳珂，郜勇. 医院移动电子签名的应用研究[J]. 中国卫生信息管理杂志，2023，20(3)：471-474，481.
WANG H，HU Y K，GAO Y. Research on the application of hospital mobile electronic signature[J]. Chinese Journal of Health Informatics and Management，2023，20(3)：471-474，481. (in Chinese)

[5] LIU X X，ZHANG M H，ZHENG Y P，et al. A linkable ring signature electronic cash scheme based on blockchain [C]// Proceedings of the 3rd International Conference on Smart Block Chain (SmartBlock). Washington D. C.，USA：IEEE Press，2021：1-4.

[6] LIN I C，CHANG C C，CHIANG H C. A novel e-cash payment system with divisibility based on proxy blind signature in web of things [J]. IEICE Transactions on Information and Systems，2022(12)：2092-2103.

[7] WANG C H. An identity-based fair contract signing protocol constructed by the confirmation signature[C]//Proceedings of the IEEE Conference on Dependable and Secure Computing (DSC). Washington D.C.，USA：IEEE Press，2019：1-6.

[8] ZUO L M，ZHANG M L，ZHANG T T，et al. A blind signature scheme with disclaimer contract in the standard model[EB/OL]. [2024-08-05]. https://link.springer.com/chapter/10.1007/978-3-030-00009-7_37.

[9] 曹素珍，王斐，郎晓丽，等. 基于无证书的多方合同签署协议[J]. 电子与信息学报，2019，41(11)：2691-2698.
CAO S Z，WANG F，LANG X L，et al. Multi-party contract signing protocol based on certificateless [J]. Journal of Electronics & Information Technology，2019，41(11)：2691-2698. (in Chinese)

[10] 国家密码管理局. SM2 密码算法使用规范：GM/T0009-2012[S]. 北京：国家密码管理局，2012.
State Cryptography Administration. SM2 cryptography algorithm application specification：GM/T0009-2012 [S]. Beijing：State Cryptography Administration，2012. (in Chinese)

[11] GRIERSON S，BUCHANAN W J，THOMSON C，et al. Scalable multi-domain trust infrastructures for segmented networks[C]//Proceedings of the 28th IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD). Washington D.C.，USA：IEEE Press，2024：182-187.

[12] JEYASHEELA R M J，GEETHA K. Secure decentralized public key infrastructure with multi-signature in blockchains [EB/OL]. [2024-08-05]. https://link.springer.com/chapter/10.1007/978-981-15-7345-3_38.

[13] 杨小东，李梅娟，任宁宁，等. 一种基于无证书的多方合同签署协议的安全性分析与改进[J]. 电子与信息学报，2022，44(10)：3627-3634.
YANG X D，LI M J，REN N N，et al. Security analysis and improvement of a multi-party contract signing protocol based on certificateless [J]. Journal of Electronics & Information Technology，2022，44(10)：3627-3634. (in Chinese)

[14] YU H F，FU S F，LIU Y X，et al. Certificateless broadcast multisignature scheme based on MPKC [J]. IEEE Access，2020，8：12146-12153.

[15] YU H F，WANG Z C. Certificateless blind signcryption with low complexity[J]. IEEE Access，2019，7：115181-115191.

[16] FENG M Q，LIN C，WU W，et al. SM2-DualRing：efficient SM2-based ring signature schemes with logarithmic size[J]. Computer Standards & Interfaces，2024，87：103763.

[17] DING F，LONG Y H，WU P L. Study on secret sharing for SM2 digital signature and its application[C]//Proceedings of the 14th International Conference on Computational Intelligence and Security (CIS). Washington D. C.，USA：IEEE Press，2018：205-209.

[18] XIAO Y C，ZHANG L，YANG Y F，et al. Provably secure multi-signature scheme based on the standard SM2 signature scheme[J]. Computer Standards & Interfaces，2024，89：103819.

[19] ZHANG Y D，HE D B，ZHANG F G，et al. An efficient blind signature scheme based on SM2 signature algorithm [EB/OL]. [2024-08-05]. https://link.springer.com/chapter/10.1007/978-3-030-71852-7_25.

[20] HE C，ZHANG B，ZHANG L，et al. Pairing-free certificateless signature scheme based on SM2 algorithm[EB/OL]. [2024-08-05]. https://link.springer.com/chapter/10.

1007/978-981-97-4519-7_21.
[21] HE C, ZHANG B, ZHANG L, et al. Improving throughput of mobile sensors via certificateless signature supporting batch verification[J]. Electronics, 2023, 12(22): 1-11.
[22] LIU S G, LIU Z T, LIANG J Q, et al. A secure certificateless ring signcryption scheme based on SM2 algorithm in smart grid[J]. Computer Communications, 2024, 218: 188-197.
[23] DING Y J, ZHANG Y N, ZHOU T P. Continual leakage resilient SM2 signature scheme[C]//Proceedings of the 3rd International Conference on Unmanned Systems (ICUS). Washington D.C., USA: IEEE Press, 2020: 931-936.
[24] 范琳琳. 基于 SM2 门限密钥分散的电子签名系统研究与实现[D]. 绵阳: 西南科技大学, 2018.
FAN L L. Research and implementation of electronic signature system based on SM2 threshold key dispersion[D]. Mianyang: Southwest University of Science and Technology, 2018. (in Chinese)
[25] SHAMIR A. How to share a secret[J]. Communications of the ACM, 1979, 22(11): 612-613.
[26] CHEN X F, ZHANG F G, TIAN H B, et al. Discrete logarithm based chameleon hashing and signatures without key exposure[J]. Computers & Electrical Engineering, 2011, 37(4): 614-623.
[27] KOBLITZ N. Elliptic curve cryptosystems[J]. Mathematics of Computation, 1987, 48(177): 203-209.
[28] POINTCHEVAL D, STERN J. Security arguments for digital signatures and blind signatures[J]. Journal of Cryptology, 2000, 13(3): 361-396.
[29] HUANG K, ZHANG X S, MU Y, et al. Building redactable consortium blockchain for industrial Internet-of-Things[J]. IEEE Transactions on Industrial Informatics, 2019, 15(6): 3670-3679.
[30] BONEH D, LYNN B, SHACHAM H. Short signatures from the Weil pairing[J]. Journal of Cryptology, 2004, 17(4): 297-319.

文字编辑 吴云芳
栏目编辑 赖玉玲