软件学报 ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*,2024,35(7):3469−3481 [doi: 10.13328/j.cnki.jos.006929]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

# 基于 SM2 数字签名的匿名凭证协议*

赵艳琦[1], 杨晓艺[2], 冯 琦[3], 禹 勇[4]

[1](西安邮电大学 网络空间安全学院, 陕西 西安 710121)

[2](香港理工大学 电子计算学系, 香港 999077)

[3](武汉大学 国家网络安全学院, 湖北 武汉 430040)

[4](陕西师范大学 计算机科学学院, 陕西 西安 710119)

通信作者: 禹勇, E-mail: yuyong@snnu.edu.cn

**摘 要**: 匿名凭证作为一种隐私保护的数字身份认证技术, 在认证用户数字身份有效性的同时, 能够保护用户身份隐私, 广泛应用于匿名身份认证、匿名通证、去中心化的数字身份管理系统等. 现有的匿名凭证通常采用承诺-签名-证明的构造范式, 通常要求采用的签名算法具备重随机化特性, 如 CL 系列签名、PS 系列签名及结构保持签名. 现实应用中多采用 ECDSA、Schnorr、SM2 等数字签名进行数字身份认证, 但其缺乏对用户身份隐私的保护. 因此, 在认证的同时, 保护身份的隐私性, 构造兼容 ECDSA、Schnorr、SM2 等数字签名的匿名凭证具有一定的现实意义. 探索基于 SM2 数字签名构造匿名凭证协议的方法. 在申请证书阶段, 借助 Pedersen 承诺对用户属性进行承诺, 同时依据 SM2 签名消息为 $H(m)$ 的结构特点, 证明 Pedersen 承诺消息与哈希承诺中消息的相等性. 为实现这种代数结构和非代数结构陈述的等价性证明, 借鉴 ZKB++技术对承诺消息进行转化, 进而实现跨域证明, 并签发基于 SM2 数字签名的授权证书. 在匿名凭证展示阶段, 结合零知识证明技术证明持有 SM2 数字签名, 保证了用户的匿名性. 给出基于 SM2 数字签名的匿名凭证协议的具体构造, 并进一步证明该协议的安全性. 最后, 通过对协议的计算复杂度分析与算法执行效率测试验证协议的有效性和可用性.

**关键词**: SM2; 匿名凭证协议; 零知识证明; ZKB++

**中图法分类号**: TP309

中文引用格式: 赵艳琦, 杨晓艺, 冯琦, 禹勇. 基于SM2数字签名的匿名凭证协议. 软件学报, 2024, 35(7): 3469–3481. http://www.jos.org.cn/1000-9825/6929.htm

英文引用格式: Zhao YQ, Yang XY, Feng Q, Yu Y. Anonymous Credential Protocol Based on SM2 Digital Signature. Ruan Jian Xue Bao/Journal of Software, 2024, 35(7): 3469–3481 (in Chinese). http://www.jos.org.cn/1000-9825/6929.htm

## Anonymous Credential Protocol Based on SM2 Digital Signature

ZHAO Yan-Qi[1], YANG Xiao-Yi[2], FENG Qi[3], YU Yong[4]

[1](School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China)

[2](Department of Computing, The Hong Kong Polytechnic University, Hong Kong 999077, China)

[3](School of Cyber Science and Engineering, Wuhan University, Wuhan 430040, China)

[4](School of Computer Science, Shaanxi Normal University, Xi'an 710119, China)

**Abstract**: As a privacy-preserving digital identity authentication technology, anonymous credentials not only authenticate the validity of the users' digital identity but also protect the privacy of their identity. Anonymous credentials are widely applied in anonymous

* 基金项目: 国家重点研发计划 (2022YFB2701500); 国家自然科学基金 (61872229, U19B2021, 62202375, 62202339); 陕西省杰出青年基金 (2022JC-47); 陕西省科协高校青年托举人才计划 (20220134); 陕西省重点研发计划 (2021ZDLGY06-04, 2020ZDLGY09-06); 陕西省自然科学基础项目 (2022JQ-604); 陕西省教育厅科学研究项目 (22JK0557)
收稿时间: 2022-11-01; 修改时间: 2022-12-15; 采用时间: 2023-02-16; jos 在线出版时间: 2023-08-23
CNKI 网络首发时间: 2023-08-28

authentication, anonymous tokens, and decentralized digital identity systems. Existing anonymous credentials usually adopt the commitment-signature-proof paradigm, which requires that the adopted signature scheme should have the re-randomization property, such as CL signatures, PS signatures, and structure-preserving signatures (SPS). In practical applications, ECDSA, Schnorr, and SM2 are widely employed for digital identity authentication, but they lack the protection of user identity privacy. Therefore, it is of certain practical significance to construct anonymous credentials compatible with ECDSA, Schnorr, SM2, and other digital signatures, and protect identity privacy during the authentication. This study explores anonymous credentials based on SM2 digital signature. Pedersen commitment is utilized to commit the user attributes in the registration phase. Meanwhile, according to the structural characteristics of SM2, the signed message is $H(m)$, and the equivalence between the Pedersen commitment message and the hash commitment message is proven. This study also employs ZKB++ technology to prove the equivalence of algebraic and non-algebraic statements. The commitment message is transformed to achieve the cross-domain proof and issue the users' credentials based on the SM2 digital signature. In the showing phase of anonymous credentials, the zero-knowledge proof is combined to prove the possession of an SM2 signature and ensure the anonymity of credentials. This study provides the construction of an anonymous credential protocol based on SM2 digital signature and proves the security of this protocol. Finally, it also verifies the effectiveness and feasibility of the protocol by analyzing the computational complexity of the protocol and testing the algorithm execution efficiency.

**Key words**: SM2; anonymous credentials protocol; zero-knowledge proof; ZKB++

数字身份作为互联网世界中用户的身份标识, 被广泛应用于电子商务、电子政务等系统, 实现身份认证[1]. 现实应用中多借助 ECDSA、Schnorr、SM2 等[2−4]数字签名构建的公钥基础设施 (public key infrastructure, PKI) 架构进行证书签发和数字身份认证[5]. 数字身份认证技术的广泛使用给人们的生活带来便利, 如, 上海建设的区域性股权市场分布式数字身份系统 DID、德国发行的电子身份证 eID[6]. 电子身份证作为公民的身份标识, 使得公民可以进行电子投票等活动, 但这类电子身份证在使用过程中会揭露用户的全部信息, 如使用基于电子身份证办理的信用卡进行消费, 服务提供商很容易就能关联用户交易, 追踪用户消费行为. 为保护用户数据隐私, 各国出台了相应的法律法规, 约束和规范对用户数据的使用行为, 如美国的《加州消费者隐私法案》(CCPA), 欧盟的《通用数据保护条例》(GDPR) 等. 我国也出台了相应的法律, 如《中华人民共和国个人信息保护法》等法律. 这使得实现数字身份认证的同时, 保护用户数据隐私备受关注.

隐私增强认证 (privacy-enhance authentication) 技术作为一种隐私保护的数字身份认证技术, 在认证用户数字身份有效性的同时, 能够保护用户身份隐私性, 也被称为匿名凭证 (anonymous credential) 技术. 最早在 1985 年由 Chaum 提出[7], 在匿名凭证协议中主要包含 3 个实体: 证书授权中心、用户、服务提供商 (验证者). 协议执行过程中, 用户向证书授权中心进行注册, 证书授权中心验证用户的合法性并为用户签发证书, 持有证书的用户进行凭证展示, 向服务提供商证明自己满足特定的验证条件. 2001 年 Camenisch 等人基于强 RSA 假设给出了首个匿名凭证构造, 该协议提供了多展示不可链接的特性[8]. 匿名凭证的构造通常采用承诺-签名-证明的范式, 对签名方案具有特殊要求, 应具备重随机化等特性. Camenisch 等人提出的 CL 数字签名, 该签名重随机化的性质可以进一步构造匿名凭证协议[9]. 基于 CL 签名, Muth 等人结合 Hyperledger Indy 平台和智能合约进一步实现了匿名凭证的智能化验证[10]. Rathee 等人为提升匿名凭证的验证效率, 进一步结合零知识简洁非交互式知识论证 (zero-knowledge succinct non-interactive arguments of knowledge, zk-SNARK) 技术[11], 提出了 ZEBRA[12]. zk-SNARK 最早由 Bitansky 等人[11]提出, 其具有较短的证明和快速验证过程, 其中 Groth16[13]和 PLONK19[14]能够提供常量级验证时间而被广泛使用于区块链系统中. ZEBRA 中 Rathee 等人借助 Groth16 给出了匿名凭证的实例化构造. Pointcheval 等提出的 PS 数字签名[15,16]相较于 CL 签名进一步压缩了签名长度, 该签名具备的重随机特性使其更适用于构造匿名凭证协议. Sonnino 等基于 PS 签名提出 Coconut, 一种支持门限分发的匿名凭证协议[17]. 为实现对用户匿名性的有效监管, Yu 等提出支持选择撤销的匿名凭证协议[18], 其基于 PS 签名与密码学累加器技术实现了用户证书的有效撤销. Hébant 等人提出可追踪的多授权中心的匿名凭证协议, 结合可验证加密技术实现对用户身份的追踪[19]. 构造标准模型下的匿名凭证通常采用结构保持签名 (structure-preserving signatures, SPS), 即等价类 SPS[20,21]. 最近, Mir 等人基于结构保持签名进一步构造了可授权的匿名凭证协议[22]. Connolly 等人设计了 Protego, 一种有效的可撤销可审计的匿名凭证协议, 并将其应用于 Hyperledger Fabric[23]. 而现实应用中采用的 ECDSA、SM2 等数字签名不具备这类重随机化的特性, 无法和匿名凭证协议有效兼容. 因此, 构造基于 ECDSA、SM2 签名的匿名凭证,

在认证的同时, 能够保护身份隐私需进一步研究. 2016 年, Chase 等人提出支持代数结构和非代数结构陈述的高效零知识证明[24], 并将其应用于构造隐私保护的证书, 但其采用交互式证明方式, 需要执行多轮交互. 如何压缩通信轮数并支持基于 ECDSA、SM2 签名的匿名凭证的证书签发, 仍需进一步探索.

SM2 数字签名是中国国家密码管理局发布的商用密码标准之一, 同时收录于中国国家密码标准 (GB/T 32918-2016) 和国际标准 ISO/IEC 14888-3:2018《信息安全技术带附录的数字签名第 3 部分: 基于离散对数的机制》中. 当前, 中国二代身份证已全面使用 SM2 数字签名技术提供身份鉴别, 北京数字证书、上海数字证书等公司相继加入 SM2 根证书计划, 基于 SM2 证书将同步在各大浏览器上线. 同时, 对 SM2 数字签名进行功能性扩展也受到广泛关注, 如设计基于 SM2 数字签名的盲签名、环签名、协同签名等[25−27]. 为进一步推动信息系统的自主安全, 本文探索基于 SM2 数字签名的匿名凭证协议构造方法, 给出匿名凭证协议的具体构造实例, 保护用户隐私的同时实现数字身份的有效认证. 在协议构造中, 为保证身份隐私性, 在申请证书阶段, 用户借助 Pedersen 承诺对自身属性进行承诺, 同时 SM2 数字签名结构特点使得签名消息为 $H(m)$, 需证明 Pedersen 承诺消息与哈希承诺中消息的等价性. 为实现这种代数结构和非代数结构陈述的等价性证明, 本文借鉴了 ZKB++技术[28]对承诺消息进行转化, 实现跨域证明, 压缩通信轮数, 并基于 SM2 数字签名分发用户证书. 在凭证展示阶段, 结合零知识证明技术证明持有 SM2 签名, 保证证书的匿名性. 本文给出基于 SM2 签名的匿名凭证协议具体构造并进一步证明该协议的安全性. 同时, 通过对协议计算复杂度的分析与算法执行效率测试验证协议的有效性和可用性.

本文第 1 节回顾承诺、零知识证明、SM2 数字签名、ZKB++算法等基础知识. 第 2 节给出匿名凭证协议的语法及安全模型. 第 3 节提出设计的基于 SM2 数字签名的匿名凭证协议, 并对其安全性进行分析. 第 4 节对设计的基于 SM2 数字签名的匿名凭证协议进行分析和效率测试. 第 5 节介绍借助基于 SM2 数字签名的匿名凭证协议构建基于区块链的数字身份认证系统. 第 6 节总结本文工作.

## 1 基础知识

### 1.1 承　诺

承诺协议包括两个参与方: 发送方和接收方. 发送方选择需要隐藏的消息, 并生成承诺值发送给接收方. 之后再对承诺执行打开操作, 将消息公开给接收方[29]. 承诺协议包括初始化、承诺和打开 3 个算法.

(1) $params \leftarrow Setup(1^{\lambda})$ 初始化: 输入安全参数 $\lambda$, 输出协议的公开参数 $params$.

(2) $c \leftarrow Com(params, m)$ 承诺: 输入公开参数 $params$ 和消息 $m$, 输出承诺值 $c$.

(3) $1/0 \leftarrow Open(params, c, m)$ 打开: 输入公开参数 $params$, 消息 $m$ 和承诺值 $c$, 输出 $1/0$, 其中 1 表示承诺有效, 0 则表示承诺无效.

承诺协议需满足隐藏性和绑定性两个性质.

隐藏性 (hiding): 给定承诺值, 恶意的接收方无法获得被承诺消息的任何信息.

绑定性 (binding): 消息 $m$ 的承诺生成后, 恶意的发送方不能将承诺的消息打开为另一不同消息 $m'$, 并通过验证.

### 1.2 零知识证明

零知识证明系统最早由 Goldwasser 等人提出[30], 现已广泛用于密码货币系统等应用中. 零知识证明系统包含证明者 Prover 和验证者 Verifier 两个参与方, 证明者能够说服验证者某个陈述为真, 使得验证者除了知道陈述为真外, 不会知道其他任何信息. 本节回顾非交互式的零知识证明系统[31], 其包含 3 个算法: 初始化, 证明算法和验证算法, 具体如下:

(1) $crs \leftarrow Setup(1^{\lambda})$ 初始化: 输入安全参数 $\lambda$, 生成公共参考串 $crs$.

(2) $\pi \leftarrow Prove(crs, w, x)$ 证明算法: 输入公共参考串 $crs$, 陈述 $x$ 以及证据 $w$, 生成证明 $\pi$.

(3) $1/0 \leftarrow Verify(crs, x, \pi)$ 验证算法: 输入公共参考串 $crs$, 陈述 $x$ 以及证明 $\pi$, 验证证明是否有效. 如果证明有效, 返回 1; 否则, 返回 0.

非交互式的零知识证明系统需满足完备性, 可靠性和零知识性.

完备性: 如果证明者知道陈述的证据, 则一定能通过有效算法使验证者接收证明者的证明.

可靠性: 如果证明者没有相应的证据, 则无法通过验证者的验证.

零知识性: 证明者在证明过程中仅向验证者透露是否拥有相应知识的陈述, 不会泄露任何关于知识的额外信息.

本文借鉴文献 [32] 中零知识证明系统的表示方法, 采用 ZKPoK$\{(x,r): y=g^x h^r\}$ 形式表示交互式证明系统, 该证明系统有效证明持有证据 $x,r$ 满足关系 $y=g^x h^r$, 具体证明过程如图 1 所示.

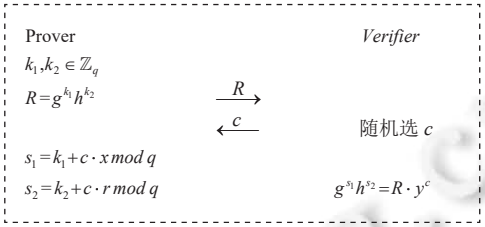


图 1 $y=g^x h^r$ 的具体证明过程

结合 Fiat-Shamir 启发式[33], 可以将交互式零知识证明系统转换为非交互式的零知识证明系统.

### 1.3 ZKB++

Giacomelli 等人基于 MPC-in-the-head 思想[34]提出一种有效的 Σ 协议 ZKBoo[35]. 为压缩 ZKBoo 的证明长度, Chase 等人提出 ZKB++. 结合 Fiat-Shamir 启发式可以将 ZKB++转化为非交互式 ZKB++协议. 本节回顾非交互式的 ZKB++协议[29]. ZKB++可以有效证明算术电路或函数, 即 $y=F(x)$, 证明者证明持有证据 $x$ 满足函数 $F$ 使得输出为 $y$, 同时验证者无需持有秘密输入. 为有效证明持有 $x$, 证明者首先将持有的证据 $x$ 进行分割, 分割成 3 份 $(x_1,x_2,x_3)$. 之后对 $(x_1,x_2,x_3)$ 进行两两分组 $(x_i,x_{i+1})$, 作为 $F''$ 的输入计算 $y_i \leftarrow F''(x_i,x_{i+1})$, $F''$ 的构造应使得 $y_1\oplus y_2\oplus y_3=y$ 成立. 证明者在这一过程中对计算的 3 个视图进行承诺, 验证者发送挑战 $e\in\{1,2,3\}$, 证明者揭露 $F''(x_e,x_{e+1})$ 对应的份额 $x_e,x_{e+1}$ 以及使用到的随机数, 验证者可以复现计算过程, 并验证证明的正确性 .

具体算法如下: 对于 $y=F(x)$, 证明者证明持有 $x$, 这一过程可以重复多轮.

(1) ZKB++.*Prove*, 证明者计算.

1) 抽样随机纸带 $k_1$, $k_2$, $k_3$, 模拟 MPC 协议, 调用 *Share* 算法生成份额, 调用 *Upd* 算法输出视图.

$$(x_1,x_2,x_3)\leftarrow Share(x,k_1,k_2,k_3)=(F'(k_1),F'(k_2),x\oplus F'(k_1)\oplus F'(k_2))$$

$$View_j\leftarrow Upd(\ldots Upd(x_j,x_{j+1},k_j,k_{j+1})\ldots) y_j\leftarrow Output(View_j).$$

2) 以视图作为输入, 计算承诺 $D_j\leftarrow H'(k_j,View_j)$, 输出 $a=(y_1,y_2,y_3,D_1,D_2,D_3)$.

3) 计算挑战 $e=H(a)$, $e\in\{1,2,3\}$.

4) 计算响应 $b=(y_{e+2},D_{e+2})$, 令 $z\leftarrow(View_{e+1},k_e,k_{e+1})$, 如果 $e\neq 1$, 添加 $x_3$ 到 $z$. 设置 $r=[b,z]$.

5) 最后证明者输出 $[e,r]$.

(2) ZKB++.*Verify*, 验证者验证证明 $[e,r]$ 的有效性.

1) 重构, 验证者运行 MPC 协议重构视图. 利用 $z$ 计算 $x_e,x_{e+1}$, $x_1\leftarrow F'(k_1)$, $x_2\leftarrow F'(k_2)$ 或者 $x_3$.

2) 获得 $View_{e+1}$, 调用 $Upd$ 算法计算 $View_e\leftarrow Upd(\ldots Upd(x_j,x_{j+1},k_j,k_{j+1})\ldots)$, 输出 $y_e\leftarrow Output(View_e)$, $y_{e+1}\leftarrow Output(View_{e+1})$, $y_{e+2}\leftarrow y\oplus y_e\oplus y_{e+1}$.

3) 计算承诺, 对于 $j\in\{e,e+1\}$, 计算 $D_j\leftarrow H'(k_j,View_j)$.

4) 恢复出 $a'=(y_1,y_2,y_3,D_1,D_2,D_3)$, 以上步骤可以执行 $t$ 轮.

5) 验证 $H(a')=e$ 是否成立, 如果等式成立, 则输出 1; 否则, 输出 0.

### 1.4 SM2 数字签名

本节回顾 SM2 数字签名方案. 根据《SM2 椭圆曲线公钥密码算法》(https://sca.gov.cn/sca/xwdt/2010-12/17/content_1002386.shtml) 规范, SM2 数字签名的椭圆曲线参数定义如下: 定义 $E$ 为 $F_p$ 满足 $y^2=x^3+ax+b \bmod p$, $a,b\in F_p$

且 $(4a^3+27b^2) \mod p \neq 0$ 的椭圆曲线, 令加法循环群 $\mathbb{G}_1=\{P|P\in E\}\cup\{O\}$ 表示包含 $E$ 所有椭圆曲线点以及无穷远点 $O$ 的集合, $\mathbb{G}_1$ 的生成元为 $G\in\mathbb{G}_1$, 阶为 $q$. 此外, 标准还定义了哈希函数 $H:\{0,1\}^*\times\{0,1\}^*\to\mathbb{Z}_q$, 实践中采用 SM3 哈希算法来实例化. SM2 数字签名算法包括 $Keygen, Sign, Verify$. 默认以上公开参数 $params=(p,F_p,q,\mathbb{G}_1,G,H)$ 均作为每个函数的输入, 随后算法定义时将不再赘述.

(1) $(sk,pk)\leftarrow Keygen(params)$ 密钥生成: 生成公私钥对 $(sk,pk)$. 随机选取 $x\in\mathbb{Z}_q$, 计算 $y=xG$. 令 $(sk=x,pk=y)$.

(2) $(\sigma)\leftarrow Sign(sk,m)$ 签名: 输入签名私钥 $sk$, 消息 $m$, 生成签名 $\sigma$. 随机选取 $k\in\mathbb{Z}_q$, 计算 $K=kG=(x_1,y_1)$, $r=H(m)+x_1 \mod q$, $s=(1+x)^{-1}\cdot(k-rx) \mod q$. $\sigma=(r,s)$.

(3) $1/0\leftarrow Verify(pk,(m,\sigma))$ 验证: 输入公钥 $pk$, 签名消息对 $(\sigma,m)$, 验证签名是否有效. 计算 $sG+(r+s)P=(x_1,y_1)$, 如果等式 $r=x_1+H(m) \mod q$ 成立, 返回 1; 否则, 返回 0.

## 2 匿名凭证的语法及安全模型

### 2.1 匿名凭证的语法

本节介绍匿名凭证的算法构成, 其包含初始化、密钥生成、<请求, 分发>、证明、验证算法. 具体如下.

(1) $(params)\leftarrow Setup(1^\lambda)$ 初始化: 输入安全参数 $\lambda$, 生成系统公开参数 $params$.

(2) $(msk,mpk)\leftarrow Keygen(params)$ 密钥生成: 输入公开参数 $params$, 输出证书授权中心公私钥对 $(msk,mpk)$.

(3) $(ssk,reg)\leftarrow<Request(params,m),Issue(params,msk)>$ <请求, 分发>: 用户与证书授权中心进行交互, 用户请求属性 $m$ 的证书. 协议执行结束, 用户获得证书 $ssk$, 证书授权中心记录用户注册信息 $reg$.

(4) $(\theta)\leftarrow Show(params,ssk)$ 证明: 用户输入公开参数 $params$ 和证书 $ssk$, 生成匿名凭证 $\theta$.

(5) $1/0\leftarrow Verify(params,mpk,\theta)$ 验证: 输入公开参数 $params$, 用户的证明 $\theta$, 证书授权中心公钥 $mpk$, 验证证明是否有效. 如果证明有效, 输出 1; 否则, 输出 0.

### 2.2 安全模型

匿名凭证协议应该满足凭证证明的不可伪造性和匿名性[17]. 匿名性: 对于同一个用户使用相同的证书生成的两个凭证证明, 敌手应该是无法区分的, 匿名性同时也保证了匿名凭证的不可链接性. 不可伪造性: 对于敌手来说, 其无法生成诚实用户的凭证证明, 并通过验证. 下面具体对匿名凭证协议不可伪造性和匿名性的安全目标分别进行定义.

在匿名凭证协议中, 刻画如下谕言机和列表.

诚实用户列表 $H\_User$: 使用变量 $i$ 标识用户, 记录诚实用户标识.

腐化用户列表 $C\_User$: 使用变量 $i$ 标识用户, 记录被敌手腐化的用户标识.

凭证问询列表 $Q\_Show$: 记录敌手对 $Show$ 算法的问询及应答.

注册记录列表 $reg$: 记录用户的注册信息.

证书谕言机 $O.List\_ssk(\cdot)$: 敌手问询标识 $i$ 用户的证书, 谕言机进行查表和应答, 并将 $i$ 添加至 $C\_User$.

请求证书谕言机 $O.Request(\cdot)$: 谕言机执行诚实的 $Request$ 协议. 敌手作为腐化的证书授权中心, 挑战者作为诚实用户与敌手进行交互. 协议结束, 返回用户的证书 $ssk[i]$.

证书分发谕言机 $O.Issue(\cdot)$: 谕言机执行诚实的 $Issue$ 协议. 敌手作为腐化的用户 $i$, 挑战者作为诚实证书授权中心与敌手进行交互. 协议结束, $i$ 添加至 $C\_User$, 返回用户的注册信息 $reg[i]$.

凭证问询谕言机 $O.Show(\cdot)$: 敌手调用 $Show$ 谕言机, 问询诚实用户 $i$ 的有效展示证明. 如果 $i\notin H\_User$, 返回 $\perp$; 否则, 谕言机执行 $Show$ 算法, 敌手作为恶意验证者. 谕言机添加 $i$ 到列表 $Q\_Show$.

借助以上谕言机, 定义如下安全性, 具体安全性游戏如图 2 和图 3 所示.

**定义 1**. 定义匿名凭证协议 $\Pi$, 对于任意概率多项式时间 (probabilistic polynomial time, PPT) 敌手 $\mathcal{A}$, 与挑战者执行游戏记为 $Exp^{ano}_{\Pi,\mathcal{A}}(\lambda)$, 若在游戏中敌手区分用户标识的优势是可忽略的, 则匿名凭证协议 $\Pi$ 满足匿名性, 敌手的优势定义如下:

$$Adv^{ano}_{\Pi,\mathcal{A}}(\lambda)=|\Pr[Exp^{ano-1}_{\Pi,\mathcal{A}}(\lambda)=1]-\Pr[Exp^{ano-0}_{\Pi,\mathcal{A}}(\lambda)=1]|.$$

**定义 2**. 定义匿名凭证协议 $\Pi$, 对于任意概率多项式时间敌手 $\mathcal{A}$, 与挑战者执行游戏记为 $Exp^{\text{forge}}_{\Pi,\mathcal{A}}(1^{\lambda})$, 若在游戏中敌手伪造凭证成功的优势是可忽略的, 则匿名凭证协议 $\Pi$ 满足不可伪造性, 敌手的优势定义如下:

$$Adv^{\text{forge}}_{\Pi,\mathcal{A}}(\lambda)=\Pr[Exp^{\text{forge}}_{\Pi,\mathcal{A}}(\lambda)=1].$$

$Exp^{ano\text{-}b}_{\Pi,\mathcal{A}}(\lambda)$
$(params)\leftarrow Setup(1^{\lambda})$
$(msk,mpk)\leftarrow Keygen(params)$
$\mathcal{O}\leftarrow\{List\_ssk,Request,Show\}$
$(i_0,i_1)\leftarrow\mathcal{A}^{\mathcal{O}}(params,msk,mpk)$
$\theta^{*}\leftarrow Show(params,List\_ssk[i_b])$
$b'\leftarrow\mathcal{A}(params,\theta^{*})$
如果$i_0,i_1\in H_User$, $List\_ssk[i_0]$, $List\_ssk[i_1]\neq\perp$, 返回$b'$
否则, 返回0.

图 2　安全性游戏 1: 匿名性实验

$Exp^{\text{forge}}_{\Pi,\mathcal{A}}(\lambda)$
$(params)\leftarrow Setup(1^{\lambda})$
$(msk,mpk)\leftarrow Keygen(params)$
$\mathcal{O}\leftarrow\{List\_ssk,Issue,Show\}$
$\theta^{*}\leftarrow\mathcal{A}^{\mathcal{O}}(params,mpk)$
如果$Verify(params,\theta^{*})=0$, 返回0;
如果$\exists i\notin C_User$, $\theta^{*}\notin Q\_Show$, $Verify(params,\theta^{*})=1$, 返回1;
否则, 返回0.

图 3　安全性游戏 2: 不可伪造性实验

## 3　基于 SM2 的匿名凭证协议设计

本节构造具体的基于 SM2 的匿名凭证协议. 在申请证书阶段, 借助 Pedersen 承诺对自身属性进行承诺, 同时 SM2 数字签名结构特点使得签名消息为 $H(m)$, 需证明 Pedersen 承诺的消息与哈希承诺中消息的等价性. 本节结合 ZKB++技术, 对承诺消息进行转化实现跨域证明, 并基于 SM2 数字签名分发用户证书. 在匿名凭证展示阶段, 结合零知识证明技术证明持有 SM2 签名, 保证证书的匿名性.

### 3.1　协议具体构造

(1) $(params)\leftarrow Setup(1^{\lambda})$: 输入安全参数 $\lambda$, 生成 SM2 数字签名公开参数, 生成 ZKB++公开参数. 按照标准设定 SM2 数字签名生成参数 $p,F_p,q$, 构造素数阶 $p$ 的循环群 $\mathbb{G}_1,G_1,H_1\in\mathbb{G}_1$ 为循环群 $\mathbb{G}_1$ 的生成元, 构造素数阶 $q$ 的循环群 $\mathbb{G}_2,G_2,H_2\in\mathbb{G}_2$ 为循环群 $\mathbb{G}_2$ 的生成元, 选取哈希函数 $H_0:\{0,1\}^{*}\times\{0,1\}^{*}\rightarrow\mathbb{Z}^{*}_q$, 公开参数 $params=(p,F_p,q,\mathbb{G}_1,G_1,H_1,\mathbb{G}_2,G_2,H_2,H_0)$.

(2) $(msk,mpk)\leftarrow Keygen(params)$: 输入公开参数 $params$, 输出证书授权中心公私钥 $(msk,mpk)$. 调用 SM2 密钥生成算法. 随机选取 $x\in\mathbb{Z}_q$, 计算 $y=xG_2$. $(msk=x,mpk=y)$.

(3) $(ssk,reg)\leftarrow<Request(params,m),Issue(params,msk)>$: 用户与证书授权中心进行交互, 请求分发证书.

1) 用户属性 $m\in\mathbb{Z}_q$, 随机选取 $r_1,r_2\in\mathbb{Z}_q$, 计算 $c_1=mG_2+r_1H_2,c_2=H_0(m)G_2+r_2H_2$, 调用 ZKB++.*Prove*, 证明持有 $(m,e)$, 使得 $e=H_0(m)$ 成立. 用户生成证明 $\pi$ 并发送给证书授权中心.

2) 证书授权中心调用 ZKB++.*Verify* 验证方案的有效性, 如果证明有效, 证书授权中心与用户进行交互分发证书, 过程如图 4 所示.

3) 用户接收 $\sigma=(s,r)$, 调用 SM2 签名验证算法, 验证签名的有效性. 如果有效, 用户获得授权证书 $ssk=(m,\sigma)$.

(4) $(\theta)\leftarrow Show(params,ssk)$: 用户构造基于 SM 签名的匿名凭证的证明算法. 展开 $ssk=(m,\sigma)=(H_0(m),(r,s))$. 用户需要证明持有 SM2 签名及消息. 用户计算承诺对于 $(r,s)$, $C_1=rG_2+R_1H_2$, $C_2=sG_2+R_2H_2$. 同时用户需承诺 $u_1=(r+s),r=H_0(m)+x_1$. 对于椭圆曲线上的点有 $sG_2=(s_x,s_y)$, $u_1y=(u_x,u_y)$, 其中 $y$ 为证书授权中心公钥. 用户计算承诺 $C_3=u_1G_2+R_3H_2$, $C_{s_x}=s_xG_1+R_{s_1}H_1,C_{s_y}=s_yG_1+R_{s_2}H_1$, $C_{u_x}=u_xG_1+R_{u_1}H_1$, $C_{u_y}=u_yG_1+R_{u_2}H_1$, 生成证明:

$$\pi_a\leftarrow\text{ZKPoK1}.Prove\{(s,s_x,s_y,R_2,R_{s_1},R_{s_2}):C_2=sG_2+R_2H_2\wedge C_{s_x}=s_xG_1+R_{s_1}H_1\wedge C_{s_y}=s_yG_1+R_{s_2}H_1\wedge sG_2=(s_x,s_y)\},$$

$$\pi_b\leftarrow\text{ZKPoK2}.Prove\{(u_1,u_x,u_y,R_3,R_{u_1},R_{u_2}):C_3=u_1G_2+R_3H_2\wedge C_{u_x}=u_xG_1+R_{u_1}H_1\wedge C_{u_y}=u_yG_1+R_{u_2}H_1\wedge u_1y=(u_x,u_y)\},$$

$$\pi_c \leftarrow \text{ZKPoK3}.Prove\{(r, H_0(m), s_x, s_y, u_x, u_y, R_1, R_4, R_{u_1}, R_{u_2}, R_{s_1}, R_{s_2}):$$
$$C_{u_x} = u_x G_1 + R_{u_1} H_1 \wedge C_{u_y} = u_y G_1 + R_{u_2} H_1 \wedge C_{s_x} = s_x G_1 + R_{s_1} H_1 \wedge C_{s_y} = s_y G_1 + R_{s_2} H_1 \wedge C_1$$
$$= rG_2 + R_1 H_2 \wedge C_4 = H_0(m)G_2 + R_4 H_2 \wedge r = H_0(m) + ((u_x, u_y) + (s_x, s_y))_x\},$$

$$\pi_d \leftarrow \text{ZKPoK4}.Prove\{(r, s, u_1, R_1, R_2, R_3): C_1 = rG_2 + R_1 H_2 \wedge C_2 = sG_2 + R_2 H_2 \wedge C_3 = u_1 G_2 + R_3 H_2 \wedge r + s = u_1\}.$$

用户输出证明 $\theta = (\pi_a, \pi_b, \pi_c, \pi_d)$, 发送给验证者.

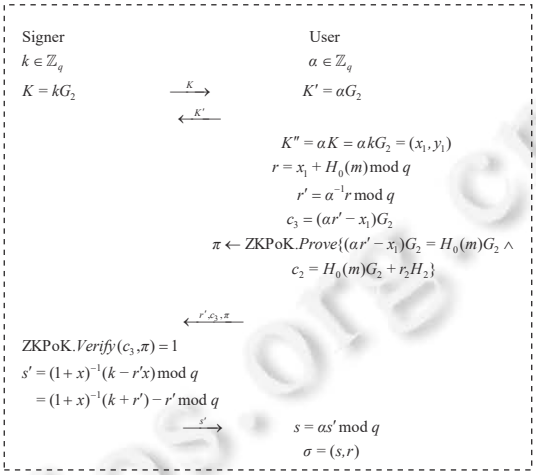


图 4 证书授权中心 (Signer) 与用户 (User) 进行交互过程

(5) $1/0 \leftarrow Verify(params, \theta)$: 输入公开参数和用户的证明, 验证证明是否有效. 验证者调用 ZKPoK.*Verify* 算法验证 $\theta$, 如果有效输出 1; 否则, 输出 0.

### 3.2 正确性

协议正确性: SM2 数字签名的正确性, ZKB++的完备性以及 ZKPoK 证明的完备性, 保证了基于 SM2 数字签名的匿名凭证协议的正确性. 具体过程如下.

初始化算法生成系统公开参数 $params = (p, F_p, q, \mathbb{G}_1, G_1, H_1, \mathbb{G}_2, G_2, H_2, H_0)$. 证书授权中心调用密钥生成算法生成公私钥 $(msk = x, mpk = y)$. 用户与证书授权中心进行交互, 申请证书. 针对用户持有的属性用户属性 $m \in \mathbb{Z}_q$, 随机选取 $r_1, r_2 \in \mathbb{Z}_q$, 计算承诺 $c_1 = mG_2 + r_1 H_2, c_2 = H_0(m)G_2 + r_2 H_2$, 调用 ZKB++, 证明持有 $(m, e)$, 同时 $e = H_0(m)$ 成立, ZKB++的正确性保证了证明 $e = H_0(m)$ 的正确性. 用户生成承诺的打开证明, 过程如图 5 所示.

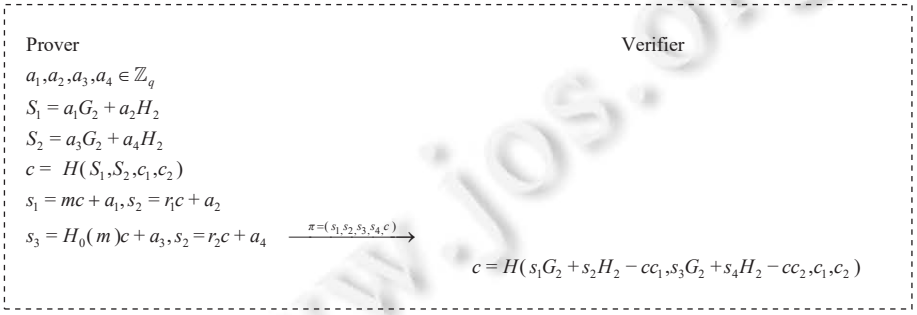


图 5 用户生成承诺打开证明

证明 $\pi$ 并发送给证书授权中心. 证书授权中心验证用户的请求证明. 与用户进行交互执行 SM2 签名的分发. 证书授权中心选取随机数, 产生盲化因子, 同时用户选取随机数, 随机化用户属性信息. 这一过程不会向证书授权中心泄露任何有关用户属性的信息, 同时需保证随机化操作的用户属性与 $c_2$ 承诺的消息相同. 用户调用知识证明算法 $\pi \leftarrow \text{ZKPoK}.Prove\{(H_0(m), r_2)(\alpha r' - x_1)G_2 = H_0(m)G_2 \wedge c_2 = H_0(m)G_2 + r_2 H_2\}$, 具体证明如图 6 所示.

证书授权中心验证用户的证明, 调用主密钥, 生成 SM2 签名, 发送给用户, 用户进行去随机化操作, 调用 SM2.*Verify* 算法验证接收的 SM2 签名正确性. 这一过程用户计算等式 $sG_2+(r+s)y=(x_1,y_1)$ 同时验证 $r=x_1+H_0(m) \bmod q$, 返回 1; 否则, 返回 0.

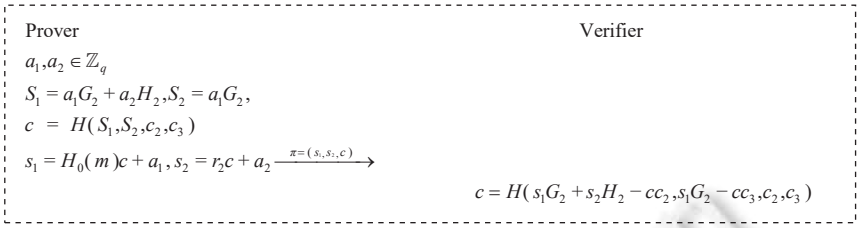


图 6 ZKPoK.*Prove* 证明生成 $\pi$ 的过程

用户计算:

$$\begin{aligned}s&=\alpha s'\\&=\alpha((1+x)^{-1}(k+r')-r') \bmod q\\&=\alpha((1+x)^{-1}(k+\alpha^{-1}r)-\alpha^{-1}r) \bmod q\\&=(1+x)^{-1}(\alpha k+r)-r \bmod q.\end{aligned}$$

计算:

$$\begin{aligned}sG_2+(r+s)y=sG_2+(r+s)xG_2&=s(1+x)G_2+rxG_2\\&=((1+x)^{-1}(\alpha k+r)-r)(1+x)G_2+rxG_2\\&=(\alpha k-rx)G_2+rxG_2\\&=\alpha kG_2=(x_1,y_1).\end{aligned}$$

用户获得证书 $ssk=(m,\sigma)=(H_0(m),(r,s))$.

在证明算法中, 用户生成 SM2 数字签名的知识证明, 证明持有 SM2 签名及对应的属性信息. 具体操作如下:

$$\pi_a\leftarrow\text{ZKPoK.}Prove\{(s,s_x,s_y,R_3,R_{s_1},R_{s_2}):$$
$$C_2=sG_2+R_2H_2\wedge C_{s_x}=s_xG_1+R_{s_1}H_1\wedge C_{s_y}=s_yG_1+R_{s_2}H_1\wedge sG_2=(s_x,s_y)\}$$

展开, 如图 7 所示.

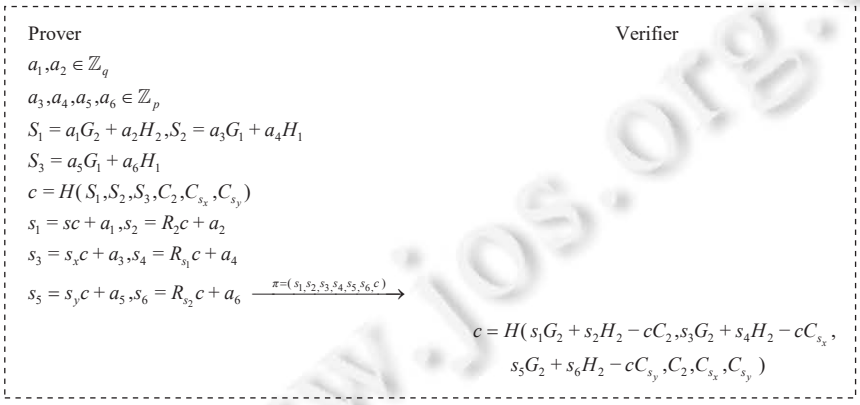


图 7 ZKPoK.*Prove* 证明生成 $\pi_a$ 的过程

$\pi_b,\pi_c,\pi_d$ 展开可采用相同方法, 只需调整相关参数即可, 这里不再重复展开. 用户输出证明 $\theta=(\pi_a,\pi_b,\pi_c,\pi_d)$, 发送给验证者. 验证者调用 ZKPoK.*Verify* 算法验证 $\theta$, 如果有效, 输出 1; 否则, 输出 0.

### 3.3 安全性分析

**定理 1**. 匿名性. 在随机谕言机模型下, 如果 ZKPoK 证明系统是零知识的, 则基于 SM2 数字签名的匿名凭证

协议满足匿名性.

证明: 模拟者 $\mathcal{S}$ 与敌手 $\mathcal{A}$ 进行交互, 执行基于 SM2 数字签名的匿名凭证协议, 这一过程, 模拟者 $\mathcal{S}$ 初始化系统并生成系统参数. 敌手 $\mathcal{A}$ 作为腐化的证书授权中心调用密钥生成算法, 生成公私钥对 $(msk, mpk)$. 敌手可进行如下谕言机问询.

$O.H(\cdot)$: 哈希谕言机问询, 敌手 $\mathcal{A}$ 选取字符串 $\{0,1\}^*$ 作为输入进行问询, 模拟者 $\mathcal{S}$ 查询 $Q_H$ 列表, 如果存在相应数值, 则反馈对应元素. 如果不存在, 则模拟者 $\mathcal{S}$ 进行模拟, 随机选取 $c \in \mathbb{Z}_q$ 并进行应答.

$O.Request(i)$: 模拟者与 $\mathcal{A}$ 进行交互, 执行<*Request, Issue*>协议. 添加 $i$ 到 $H\_User$, 模拟者作为诚实用户 $i$, 通过编辑谕言机 $H$ 模拟生成证明 $\pi$. 敌手持有 $msk$ 进行交互, 返回 $ssk[i] = (m,(r,s))$.

$O.Show(i)$: 敌手可以对标识 $i$ 的匿名凭证进行问询. 如果 $i \in H\_User$, 模拟者展开 $ssk[i] = (m_i,(r_i,s_i))$, 执行 *Show* 算法并生成 $\theta = (\pi_a,\pi_b,\pi_c,\pi_d)$ 进行应答, 添加 $(i,\theta)$ 到 $Q\_Show$. $O.List\_ssk(i)$: 敌手问询 $i$ 的证书, 模拟者查询列表并进行回答, 添加 $i$ 到 $C\_User$.

根据谕言机问询, 敌手输出 $i_0, i_1$ 标识. 如果 $i_0, i_1 \in H\_User$, $List\_ssk[i_0], List\_ssk[i_1] \notin \perp$, 模拟者选择 $i_b^*$, 并返回模拟的展示凭证 $\theta^* = (\pi_a,\pi_b,\pi_c,\pi_d)$ 给敌手, 敌手输出自己猜测 $b'$. 由 ZKPoK 的零知识性保证展示凭证不会向敌手泄露任何用户的标识信息, 因此, 敌手无法进行区分, 设计的基于 SM2 的匿名凭证协议满足匿名性, 证毕.

**定理 2**. 不可伪造性. 在随机谕言机模型下, 如果 SM2 数字签名是存在性不可伪造的, 则基于 SM2 数字签名的匿名凭证协议满足不可伪造性.

证明: 模拟者 $\mathcal{S}$ 与敌手 $\mathcal{A}$ 进行交互执行基于 SM2 的匿名凭证协议, 这一过程, 模拟者 $\mathcal{S}$ 收到 SM2 签名的公钥实例 $y = xG_2$. 协议具体证明如下: 模拟者 $\mathcal{S}$ 初始化系统并生成系统参数, 调用密钥生成算法, 将 $y$ 作为 $mpk$ 发送给敌手 $\mathcal{A}$ 作为腐化的用户与模拟者进行交互, 敌手可进行如下谕言机问询.

$O.H(\cdot)$: 哈希谕言机问询, 敌手选取字符串 $\{0,1\}^*$ 作为输入进行问询, 模拟者 $\mathcal{S}$ 查询 $Q_H$ 列表, 如果存在相应数值, 则反馈对应元素. 如果不存在, 则模拟者进行模拟, 随机选取 $c \in \mathbb{Z}_q$ 并进行应答.

$O.Issue(i)$: 模拟者与 $\mathcal{A}$ 进行交互, 执行 $<Request, Issue>$ 协议. 添加 $i$ 到 $C\_User$, 敌手进行问询, 模拟者作为诚实证书授权中心与敌手进行交互, 当接收到敌手问询, 模拟者调用 ZKPoK.*Verify* 算法验证, 如果验证有效, 模拟者重放这一过程, 调用 *Extract*() 算法, 提取出问询消息 $H(m)$. 模拟者内部问询消息 $H(m)$ 的 SM2 签名, 挑战者反馈 SM2 签名给模拟者, 模拟者发送签名给敌手. 记录 $reg[i] = (c_1,c_2,c_3,\pi,\text{ZKB++}.Prove)$.

$O.Show(i)$: 敌手可以对标识 $i$ 的匿名凭证进行问询. 如果 $i \in H\_User$, 模拟者执行 *Show* 算法并生成 $\theta = (\pi_a,\pi_b,\pi_c,\pi_d)$ 进行应答, 添加 $(i,\theta)$ 到 $Q\_Show$.

$O.List\_ssk(i)$: 敌手 $\mathcal{A}$ 问询 $i$ 的证书, 模拟者查询列表并进行回答, 添加 $i$ 到 $C\_User$.

根据谕言机问询, 敌手输出伪造 $\theta^*$. 如果验证算法无效, 中断与敌手交互. 如果敌手伪造凭证通过验证, 令 $i$ 标识第 $i$ 次哈希问询, 输出 $(i,\theta^*)$. 然后根据分叉引理[36], 通过回放, 可以输出两个伪造的凭证元组, 通过展开 $\theta^* = (\pi_a,\pi_b,\pi_c,\pi_d)$ 和 $\theta_1{}^* = (\pi_{1,a},\pi_{1,b},\pi_{1,c},\pi_{1,d})$, 依据 $(\pi_c,\pi_d),(\pi_{1,c},\pi_{1,d})$, 模拟者提取出 $H(m),(r,s)$ 作为 SM2 签名, 并发送给挑战者, 赢得游戏.

综上, 如果存在敌手 $\mathcal{A}$ 能够以不可忽略的概率 $\varepsilon(\lambda)$ 伪造基于 SM2 的匿名凭证, 那么一定存在模拟者 $S$ 能够伪造 SM2 签名, 进而攻破离散对数困难问题. 证毕.

## 4 性能分析

本节从协议的计算复杂度与执行效率对基于 SM2 数字签名的匿名凭证协议进行分析, 并与已有方案进行对比.

### 4.1 计算复杂度分析

本节对基于 SM2 数字签名的匿名凭证协议与已有的基于重随机化签名构造的匿名认证方案 [18] 和基于 ECDSA 签名构造的知识签名方案 [24] 进行比较分析. 本文方案与已有方案计算复杂度比较如表 1 所示. 参数说明如下: $\lambda$ 代表安全参数, 设置电路尺寸为$|F|$, 输入的秘密值尺寸为$|x|$. 令 pub 表示注册过程中公钥操作, sym 表示

对称密钥操作. *Prove* 表示 *Show* 算法用户执行的操作, *Verify* 表示 *Verify* 算法验证者执行的操作, 其中 $E$ 表示点积运算, $M$ 表示模乘运算, $P$ 表示 pairing 运算, $T$ 表示 $G_T$ 上的指数运算. 文献 [18] 在执行注册过程中采用密码学累加器技术, 在执行证明和验证过程中需要 O($\lambda$pub) 操作. 文献 [24] 的方案 (1) 执行证明和验证需要 O(|$x$|pub+|$F$|sym) 操作, 文献 [24] 的方案 (2) 执行证明和验证需要 O($\lambda$pub+(|$F$|+|$x$|$\lambda$)sym) 操作, 本文方案执行证明和验证需 O((|$x$|+$\lambda$)pub+(|$F$|×$\lambda$)sym) 操作. 文献 [18] 只支持一种模式的陈述证明即代数结构的陈述证明, 本文及文献 [24] 不仅支持代数结构陈述同时支持非代数结构陈述, 使用范围更广, 支持数据类型更丰富. 针对非代数结构的陈述 $H(m)=y$ 的证明, 文献 [24] 中采用 garbled circuits 及不经意传输协议, 文献 [24] 的最优方案 (2) 其通信代价与输入信息|$x$|大小相关, 而本文方案通信代价与电路尺寸|$F$|相关. 同时, 对方案的交互模式进行了对比. 文献 [18] 在用户注册证书过程中借助非交互零知识证明实现认证, 只需一轮通信, 为非交互模式. 文献 [24] 在这一过程需多轮交互, 为交互模式, 同时随着用户属性的增加, 也会增加通信轮数. 本文方案结合 ZKB++技术, 通过采用 Fiat-Shamir 转换实现了非交互式认证. 本文方案相较于文献 [18,24], 在计算证明和验证阶段的操作更少, 进一步压缩了计算开销.

表 1　本文方案与文献 [18]、文献 [24] 计算复杂度比较

| 方案 | Proof | | | Verification | | | 交互模式 |
|---|---|---|---|---|---|---|---|
| | pub | sym | *Prove* | pub | sym | *Verify* | |
| 文献[18] | $\lambda$ | — | 11$E$+7$M$+5$P$+2$T$ | $\lambda$ | — | 8$E$+7$M$+8$P$+4$T$ | 非交互式 |
| 文献[24](1) | \|$x$\| | \|$F$\| | 52$E$+26$M$ | \|$x$\| | \|$F$\| | 51$E$+34$M$ | 交互式 |
| 文献[24](2) | $\lambda$ | \|$F$\|+\|$x$\|$\lambda$ | 52$E$+26$M$ | $\lambda$ | \|$F$\|+\|$x$\| | 51$E$+34$M$ | 交互式 |
| 本文方案 | \|$x$\|+$\lambda$ | \|F\|$\lambda$ | 44$E$+22$M$ | \|$x$\|+$\lambda$ | \|F\|$\lambda$ | 45$E$+30$M$ | 非交互式 |

### 4.2 效率分析

本节对基于 SM2 数字签名的匿名凭证协议的计算开销进行测试. 采用 Windows 11 64 位操作系统进行效率测试, 实验环境为 AMD Ryzen9 5900HS with Radeon Graphics 处理器, 3.30 GHz, 机带 RAM 16.0 GB 电脑, 编译环境采用 Visual Studio 2019 IDE 版本, 基于 Miracl 大整数库, 编译语言采用 C++编译程序. 通过对匿名凭证证明算法和验证算法执行 1 000 次, 求得其时间开销的平均值. 基于 SM2 数字签名的匿名凭证协议的证明与验证算法时间开销如图 8 所示. 首先对基于 SM2 数字签名的匿名凭证协议的证明算法和验证算法中的 4 个零知识证明分别进行测试, ZKPoK1.*Prove* 生成证明用时 6.52 ms, ZKPoK2.*Prove* 生成证明消耗时间为 6.53 ms, ZKPoK3.*Prove* 生成证明需 13.13 ms, 而 ZKPoK4.*Prove* 生成证明需消耗 6.54 ms, 生成匿名凭证的证明算法的总时间消耗为 32.72 ms, 生成证明阶段, ZKPoK1.*Prove*、ZKPoK2.*Prove* 和 ZKPoK4.*Prove* 执行时间相近, ZKPoK3.*Prove* 生成证明大约是其他证明时间消耗的 2 倍. 针对本文方案的验证算法, 执行 ZKPoK1.*Verify* 消耗 9.77 ms, 执行 ZKPoK2.*Verify* 消耗 9.79 ms, 执行 ZKPoK3.*Verify* 消耗 19.80 ms, 执行 ZKPoK1.*Verify* 消耗 10.10 ms. 验证算法总时间消耗为 49.46 ms, ZKPoK3.*Verify* 消耗时间大约是其他证明时间消耗的 2 倍. 本文方案总体时间消耗均为毫秒级, ZKPoK3 的证明和验证有待进一步优化.

进一步与已有方案进行证明与验证算法的执行效率对比, 如图 9 所示. 文献 [18] 时间消耗最多, 因为文献 [18] 在证明和验证算法中需执行多个配对运算. 与文献 [24] 相比, 本文执行的点积运算和模乘运算更少, 时间消耗更少. 因此, 本文方案不仅实现了非交互式的证明, 同时时间开销更小, 与已有方案相比更具可用性.

## 5 应　用

在云计算、物联网、人工智能等多种新兴技术融合的开放环境下, 数据安全流转、数据安全利用为实现数据资源共享提供了安全保障. 数据安全流转和利用均需要实体来承载, 可信的数字身份是保证数据来源可信及用户开展各类数字活动的基础. 然而, 各种各样的网络攻击及复杂的网络环境使得用户在进行数字身份认证时消耗成本及信任代价过高, 同时, 存在泄露用户信息的风险, 因此, 亟需采用新技术手段解决传统数字身份认证方法的不足.

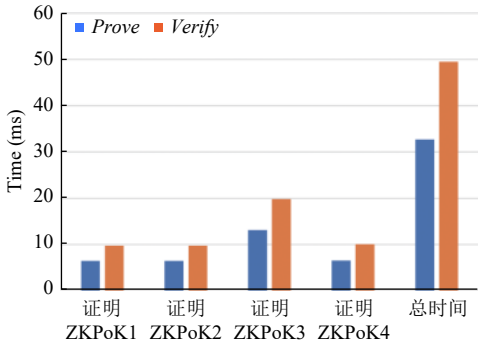


图 8 本文方案证明与验证算法的时间开销

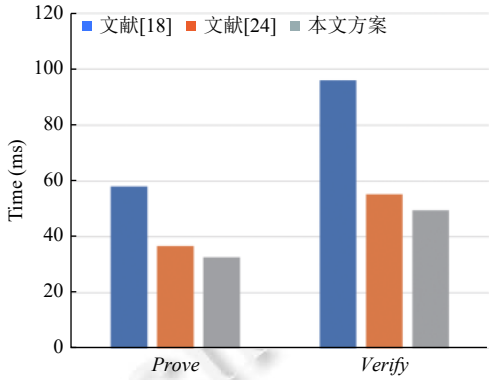


图 9 本文方案的证明与验证算法与已有方案在时间开销上的对比

针对以上问题, 本文结合基于 SM2 数字签名的匿名凭证协议和区块链技术, 设计了一种基于区块链的数字身份认证系统, 实现对用户的统一身份认证, 并保护身份隐私. 系统模型如图 10 所示.

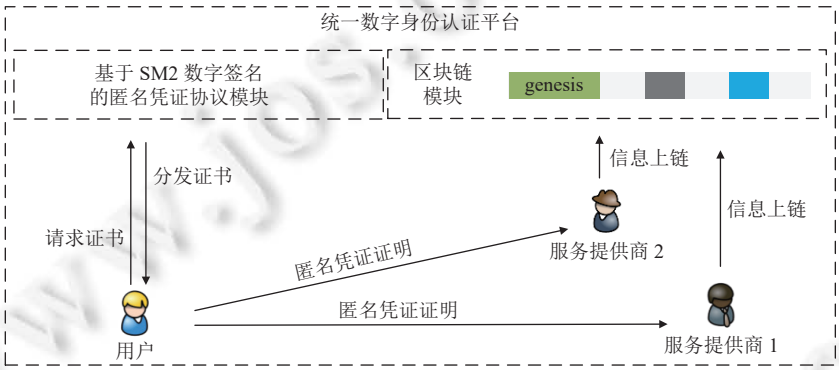


图 10 基于区块链的数字身份认证系统

在系统中主要包含: 用户、服务提供商、统一数字身份认证平台等模块. 统一数字身份认证平台集成了基于 SM2 数字签名的匿名凭证协议模块和区块链模块. 统一数字身份认证平台为用户提供认证服务. 基于 SM2 数字签名的匿名凭证协议模块提供用户属性证明, 证书分发, 用户根据应用需求生成匿名凭证证明, 同时不会泄露身份隐私. 区块链模块用于记录公开信息及用户的匿名凭证证明等信息.

## 6 总 结

本文探索基于 SM2 数字签名构造匿名凭证协议. 借鉴 ZKB++技术对承诺消息进行转化, 实现跨域证明, 并基于 SM2 数字签名分发用户证书. 本文给出基于 SM2 数字签名的匿名凭证协议的具体构造并进一步证明协议的安全性. 最后, 通过对协议计算复杂度分析与算法执行效率测试验证本文协议的有效性和可用性. 同时, 本文给出了协议的一个具体应用场景, 用于构建统一数字身份认证平台. 在现实应用中通常需要综合考虑多因素构建身份认证的方法, 如何借助基于 SM2 数字签名的匿名凭证协议构建匿名多因素身份认证协议的方法, 仍需进一步探索. 我们将其作为未来的研究工作, 将进一步探索.

**References:**

[1] First Research Institute of the Ministry of Public Security of P.R.C, China Academy of Information and Communication Technology, *et al*. Blockchain application service white paper based on trusted digital identity. 2020 (in Chinese) https://www.xdyanbao.com/doc/

h02arce83u?bd_vid=10125837072972166658

[2] Lindell Y. Fast secure two-party ECDSA signing. Journal of Cryptology, 2021, 34(4): 44. [doi: 10.1007/s00145-021-09409-9]

[3] Schnorr CP. Efficient identification and signatures for smart cards. In: Proc. of the 1989 Workshop on the Theory and Application of Cryptographic Techniques. Houthalen: Springer, 1989. 688–689. [doi: 10.1007/3-540-46885-4_68]

[4] Wang ZH, Zhang ZF. Overview on public key cryptographic algorithm SM2 based on elliptic curves. Journal of Information Security Research, 2016, 2(11): 972–982 (in Chinese with English abstract).

[5] Kurbatov O, Kravchenko P, Poluyanenko N, Demenko Y, Kuznetsova T. Global digital identity and public key infrastructure. In: Proc. of the 16th Int'l Conf. on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. ICTERI Workshops 2020. Kharkiv, 2020. 1–12.

[6] Information on German Identity Card. eID. 2022. https://www.germany.info/us-en/service/02-PassportsandIDCards/id-card/917860

[7] Chaum D. Security without Identification: Transaction systems to make big brother obsolete. Communications of the ACM, 1985, 28(10): 1030–1044. [doi: 10.1145/4372.4373]

[8] Camenisch J, Lysyanskaya A. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In: Proc. of the 2001 Int'l Conf. on the Theory and Application of Cryptographic Techniques. Innsbruck: Springer, 2001. 93–118. [doi: 10.1007/3-540-44987-6_7]

[9] Camenisch J, Lysyanskaya A. Signature schemes and anonymous credentials from bilinear maps. In: Proc. of the 24th Annual Int'l Cryptology Conf. on Advances in Cryptology. Santa Barbara: Springer, 2004. 56–72. [doi: 10.1007/978-3-540-28628-8_4]

[10] Muth R, Galal T, Heiss J, Tschorsch F. Towards smart contract-based verification of anonymous credentials. IACR Cryptology ePrint Archive, Paper 2022/492, 2022.

[11] Bitansky N, Canetti R, Chiesa A, Tromer E. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In: Proc. of the 3rd Innovations in Theoretical Computer Science Conf. Cambridge: ACM, 2012. 326–349. [doi: 10.1145/2090236.2090263]

[12] Rathee D, Vamsi Policharla G, Xie TC, Cottone R, Song D. ZEBRA: Anonymous credentials with practical on-chain verification and applications to KYC in DeFi. IACR Cryptology ePrint Archive, 2022.

[13] Groth J. On the size of pairing-based non-interactive arguments. In: Proc. of the 35th Annual Int'l Conf. on the Theory and Applications of Cryptographic Techniques. Vienna: Springer, 2016. 305–326. [doi: 10.1007/978-3-662-49896-5_11]

[14] Gabizon A, Williamson ZJ, Ciobotaru O. PLONK: Permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. IACR Cryptology ePrint Archive, Paper 2019/953, 2019.

[15] Pointcheval D, Sanders O. Short randomizable signatures. In: Proc. of the 2016 Cryptographers' Track at the RSA Conf. on Topics in Cryptology. San Francisco: Springer, 2016. 111–126. [doi: 10.1007/978-3-319-29485-8_7]

[16] Pointcheval D, Sanders O. Reassessing security of randomizable signatures. In: Proc. of the 2018 Cryptographers' Track at the RSA Conf. on Topics in Cryptology. San Francisco: Springer, 2018. 319–338. [doi: 10.1007/978-3-319-76953-0_17]

[17] Sonnino A, Al-Bassam M, Bano S, Meiklejohn S, Danezis G. Coconut: Threshold issuance selective disclosure credentials with applications to distributed ledgers. arXiv:1802.07344, 2020.

[18] Yu Y, Zhao YQ, Li YN, Du XJ, Wang LH, Guizani M. Blockchain-based anonymous authentication with selective revocation for smart industrial applications. IEEE Trans. on Industrial Informatics, 2020, 16(5): 3290–3300. [doi: 10.1109/TII.2019.2944678]

[19] Hébant C, Pointcheval D. Traceable constant-size multi-authority credentials. In: Proc. of the 13th Int'l Conf. on Security and Cryptography for Networks. Amalfi: Springer, 2022. 411–434. [doi: 10.1007/978-3-031-14791-3_18]

[20] Hanser C, Slamanig D. Structure-preserving signatures on equivalence classes and their application to anonymous credentials. In: Proc. of the 20th Int'l Conf. on the Theory and Application of Cryptology and Information Security. Kaoshiung: Springer, 2014. 491–511. [doi: 10.1007/978-3-662-45611-8_26]

[21] Connolly A, Lafourcade P, Perez-Kempner O. Improved constructions of anonymous credentials from structure-preserving signatures on equivalence classes. In: Proc. of the 25th IACR Int'l Conf. on Practice and Theory of Public-key Cryptography. Virtual Event: Springer, 2022. 409–438. [doi: 10.1007/978-3-030-97121-2_15]

[22] Mir O, Slamanig D, Bauer B, Mayrhofer R. Practical delegatable anonymous credentials from equivalence class signatures. IACR Cryptology ePrint Archive, Paper 2022/680, 2022.

[23] Connolly A, Deschamps J, Lafourcade P, Perez-Kempner O. Protego: Efficient, revocable and auditable anonymous credentials with applications to hyperledger fabric. In: Proc. of the 23rd Int'l Conf. on Cryptology in India. Kolkata: Springer, 2022. 249–271. [doi: 10.1007/978-3-031-22912-1_11]

[24] Chase M, Ganesh C, Mohassel P. Efficient zero-knowledge proof of algebraic and non-algebraic statements with applications to privacy

preserving credentials. In: Proc. of the 36th Annual Int'l Cryptology Conf. on Advances in Cryptology. Santa Barbara: Springer, 2016. 499–530. [doi: 10.1007/978-3-662-53015-3_18]

[25] He DB, Zhang YD, Zhang FG, Feng Q, Wang J. A lightweight SM2 blind signature generation method and system: 201910473354.0. 2019-09-06 (in Chinese).

[26] Fan Q, He DB, Luo M, Huang XY, Li DW. Ring signature schemes based on SM2 digital signature algorithm. Journal of Cryptologic Research, 2021, 8(4): 710–723 (in Chinese with English abstract). [doi: 10.13868/j.cnki.jcr.000472]

[27] He DB, Zhang J N, Feng Q, Wang J, Chen B W. A Lightweight SM2 two-party collaborative method to generate digital signatures. China, 201910147366.4. 2019-07-12 (in Chinese).

[28] Chase M, Derler D, Goldfeder S, Orlandi C, Ramacher S, Rechberger C, Slamanig D, Zaverucha G. Post-quantum zero-knowledge and signatures from symmetric-key primitives. In: Proc. of the 2017 ACM SIGSAC Conf. on Computer and Communications Security. Dallas: ACM, 2017. 1825–1842. [doi: 10.1145/3133956.3133997]

[29] Backes M, Hanzlik L, Herzberg A, Kate A, Pryvalov I. Efficient non-interactive zero-knowledge proofs in cross-domains without trusted setup. In: Proc. of the 22nd IACR Int'l Conf. on Practice and Theory of Public-key Cryptography. Beijing: Springer, 2019. 286–313. [doi: 10.1007/978-3-030-17253-4_10]

[30] Goldwasser S, Micali S, Rackoff C. The knowledge complexity of interactive proof-systems. In: Proc. of the 17th Annual ACM Symp. on Theory of Computing. Providence: ACM, 1985. 291–304. [doi: 10.1145/22145.22178]

[31] Derler D, Slamanig D. Key-homomorphic signatures: Definitions and applications to multiparty signatures and non-interactive zero-knowledge. Designs, Codes and Cryptography, 2019, 87(6): 1373–1413. [doi: 10.1007/s10623-018-0535-9]

[32] Camenisch J, Stadler M. Proof systems for general statements about discrete logarithms. ETH Zurich, 1997. [doi: 10.3929/ethz-a-006651937]

[33] Fiat A, Shamir A. How to prove yourself: Practical solutions to identification and signature problems. In: Proc. of the 1986 Advances in Cryptology (CRYPTO 1986). Berlin: Springer, 1986. 186–194. [doi: 10.1007/3-540-47721-7_12]

[34] Ishai Y, Kushilevitz E, Ostrovsky R, Sahai A. Zero-knowledge from secure multiparty computation. In: Proc. of the 39th Annual ACM Symp. on Theory of Computing. San Diego: ACM, 2007. 21–30. [doi: 10.1145/1250790.1250794]

[35] Giacomelli I, Madsen J, Orlandi C. ZKBoo: Faster zero-knowledge for Boolean circuits. In: Proc. of the 25th USENIX Conf. on Security Symp. Austin: USENIX Association, 2016. 1069–1083.

[36] Pointcheval D, Stern J. Security arguments for digital signatures and blind signatures. Journal of Cryptology, 2000, 13(3): 361–396. [doi: 10.1007/s001450010003]

## 附中文参考文献:

[1] 公安部第一研究所, 中国信息通信研究院等. 基于可信数字身份的区块链应用服务白皮书. 2020. https://www.xdyanbao.com/doc/h02arce83u?bd_vid=10125837072972166658

[4] 汪朝晖, 张振峰. SM2椭圆曲线公钥密码算法综述. 信息安全研究, 2016, 2(11): 972–982.

[25] 何德彪, 张语荻, 张方国, 冯琦, 王婧. 一种轻量级SM2盲签名生成方法及系统: 201910473354.0. 2019-09-06.

[26] 范青, 何德彪, 罗敏, 黄欣沂, 李大为. 基于SM2数字签名算法的环签名方案. 密码学报, 2021, 8(4): 710–723. [doi: 10.13868/j.cnki.jcr.000472]

[27] 何德彪, 张佳妮, 冯琦, 王婧, 陈泌文. 一种轻量级SM2两方协同生成数字签名的方法. 中国, 201910147366.4. 2019-07-12.

**赵艳琦**(1992－), 男, 博士, 副教授, CCF 专业会员, 主要研究领域为公钥密码学, 区块链安全.

**冯琦**(1994－), 女, 博士, 副研究员, CCF 专业会员, 主要研究领域为应用密码学, 安全协议, 隐私计算.

**杨晓艺**(1993－), 女, 博士, 主要研究领域为隐私计算, 安全多方计算.

**禹勇**(1980－), 男, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为密码学, 数据安全, 区块链安全.