软件学报 ISSN 1000-9825, CODEN RUXUEW
[doi: 10.13328/j.cnki.jos.007502] [CSTR: 32375.14.jos.007502]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于 SM2 的匿名认证与密钥协商协议*

赵 璇[1], 赵艳琦[1], 孙铭鸿[1], 禹 勇[2]

[1](西安邮电大学 网络空间安全学院, 陕西 西安 710121)
[2](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: 随着 5G 技术的快速发展, 5G-AKA 协议作为 5G 技术的核心安全机制, 受到广泛关注. 5G-AKA 协议的部署推动了通信网络的高速互联, 但也带来了用户对隐私泄露的担忧. 运营商在协议交互过程将收集大量数据, 这些数据一旦泄露, 将给用户造成严重的威胁. 因此, 提出基于 SM2 的匿名认证与密钥协商协议, 实现用户认证过程的隐私增强, 达到用户信息的最小揭露. 扩展了国密 SM2 数字签名算法实现对多消息的签名, 结合 ElGamal 算法对用户的身份进行加密并利用零知识证明技术保证用户证书的匿名性, 有效实现对用户身份的匿名认证. 协议保护合法用户在网络活动中的身份隐私, 并有效阻断对用户信息的非法获取. 此外, 协议还具备对恶意用户的可追责性, 其允许经授权的监管机构在合法流程下还原出用户身份. 最后, 开展协议实验测评, 基于 Windows 及 Raspberry Pi 4B 平台上进行部署和实现. 测评结果显示, 匿名认证与密钥协商过程耗时均为毫秒级, 充分展示了所提协议的高效性与实用性.

关键词: 5G-AKA; 匿名认证; SM2 数字签名; 隐私保护
中图法分类号: TP309

中文引用格式: 赵璇, 赵艳琦, 孙铭鸿, 禹勇. 基于SM2的匿名认证与密钥协商协议. 软件学报. http://www.jos.org.cn/1000-9825/7502.htm

英文引用格式: Zhao X, Zhao YQ, Sun MH, Yu Y. Anonymous Authentication and Key Agreement Protocol Based on SM2. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/7502.htm

Anonymous Authentication and Key Agreement Protocol Based on SM2

ZHAO Xuan[1], ZHAO Yan-Qi[1], SUN Ming-Hong[1], YU Yong[2]

[1](School of Cyberspace Security, Xi’an University of Posts & Telecommunications, Xi’an 710121, China)
[2](School of Computer Science, Shaanxi Normal University, Xi’an 710119, China)

Abstract: With the rapid development of 5G technology, the 5G-AKA protocol, as the core security mechanism of 5G technology, has caught widespread attention. Although the deployment of the 5G-AKA protocol has promoted the high-speed interconnection of communication networks, it has also raised users’ concerns about privacy leakage. During the protocol interaction, operators will collect a large amount of data, and once the data is leaked, it will pose a serious threat to users. Therefore, this study proposes an anonymous authentication and key agreement protocol based on SM2 to enhance the privacy of the user authentication process and minimize the disclosure of user information. It extends the Chinese cryptographic SM2 digital signature algorithm to achieve the signature of multiple messages, combines the ElGamal algorithm to encrypt the user’s identity, and adopts zero-knowledge proof technology to ensure the anonymity of the user credentials, thereby achieving the anonymous authentication of the user’s identity. The protocol protects the identity privacy of legitimate users in network activities and effectively blocks the illegal acquisition of user information. Additionally, the protocol holds the accountability for malicious users, allowing authorized regulatory agencies to restore the user’s identity in a legal process.

* 基金项目: 国家重点研发计划 (2022YFB2701500); 国家自然科学基金 (62272385, 62202375); 陕西省杰出青年基金 (2022JC-47); 陕西省重点研发计划 (2024GX-ZDCYL-01-09, 2024GX-ZDCYL-01-15); 陕西省科协青年人才托举计划 (20220134); 陕西省高校青年创新团队 (24JP180)
收稿时间: 2025-03-26; 修改时间: 2025-05-17, 2025-06-24; 采用时间: 2025-06-30; jos 在线出版时间: 2026-01-07

Finally, protocol experimental evaluations are conducted, with deployment and implementation carried out on Windows and Raspberry Pi 4B platforms. The evaluation results show that the consumed time of the anonymous authentication and key agreement process is at the millisecond level, fully demonstrating the efficiency and practicality of this protocol.

Key words: 5G-AKA; anonymous authentication; SM2 digital signature; privacy protection

随着移动通信技术的发展, 2G、3G、4G 直至如今的 5G 网络已极大提升了用户的通信能力. 借助这些技术, 用户可以随时随地进行通话、上网、视频通话等一系列操作, 享受快速的连接体验和高速的数据传输[1]. 运营商为用户提供了便捷服务的同时也采集了大量的用户信息, 特别是用户在每次接入网络的过程中, 地理位置信息、使用习惯、通信记录等敏感数据都会被收集. 这些信息包含了日常行为轨迹和网络活动, 能详细勾勒出用户的生活方式、工作地点等. 由于移动设备与蜂窝网络连接, 运营商可以通过用户设备的国际移动用户识别码 (IMSI) 或类似的标识符, 追踪用户在不同位置的移动情况, 甚至在用户未主动使用设备时, 也能通过网络的定位功能确定用户的实时位置.

为了保护用户的隐私并增强网络安全性, AKA (authentication and key agreement) 协议[2]在 3G 网络中首次被引入, 用于用户 (UE) 与服务网络 (SN) 间身份认证和密钥协商. 其通过用户与归属网络 (HN) 共享的会话密钥生成认证响应, 确保用户身份的真实性与网络通信的安全性. AKA 协议的核心机制是通过挑战-应答的方式验证用户身份, 归属网络发送一个随机数给用户设备, 并通过共享密钥对该随机数进行加密, 生成预期响应 RES*发送给服务网络. 用户设备通过共享密钥对该随机数进行加密, 生成响应值 RES, 并发送给服务网络. 服务网络将 RES*与 RES 进行对比. 如果结果匹配, 认证成功, 服务网络和用户设备共同生成一个会话密钥, 用于加密通信数据, 确保通信的机密性和完整性. 但用户的永久标识符在无线信道传输时有可能以明文形式出现, 存在一定的隐私泄露风险. 为了保护用户的身份信息, 在 5G-AKA[3]协议中, 不允许用户的永久标识符 (IMSI) 通过无线网络以明文形式发送, 其使用公钥加密来加密用户身份信息, 增强了用户身份的保密性.

5G-AKA 协议通过 SUCI (加密永久标识符)[4]对用户真实身份进行加密, 理论上可阻止攻击者在传输过程中直接截获用户身份信息. SUCI 采用加密算法将 IMSI 转换为匿名标识符, 确保链路上的身份传输安全. 然而, 这一机制存在核心矛盾: 运营商作为解密密钥的唯一持有者, 可随时解密 SUCI 还原 IMSI, 从而获取用户的真实身份. 除此之外, 运营商还可以与服务网络合谋, 从而可以刻画出用户在网络上的所有活动轨迹. 5G-AKA 在技术层面上保护了用户身份的传输安全, 却忽视了数据控制者 (运营商) 本身可能成为隐私泄露的源头, 而用户在这一体系中始终处于被动地位[5].

为了解决上述移动通信过程中的隐私泄露问题, 本文提出了基于 SM2 的匿名认证与密钥协商协议, 确保了用户在身份认证中的匿名性, 保护了用户的隐私信息, 使除合法的追责机构外任何参与方都无法获取用户的隐私信息. 针对现有的通信协议大都使用国外的密码算法, 可能存在后门攻击甚至会威胁国家安全. 本协议通过使用国产 SM2 算法为 5G 通信网的安全防护与隐私保护提供了自主、可控、安全的解决方案.

本文提出的基于 SM2 的匿名认证与密钥协商协议, 实现了对用户身份的匿名认证. 通过引入追责机构, 确保用户发生违法行为时的可追责性, 实现隐私保护与有效监管. 具体而言, 在凭证生成阶段, 归属网络利用 SM2 数字签名对用户的属性信息进行签名; 凭证获取阶段, 用户利用 ElGamal 加密算法对身份信息进行加密, 并利用零知识证明技术证明自己持有归属网络签署的 SM2 数字签名的相关信息, 确保了用户身份的匿名性; 追责阶段, 当用户出现违法行为时, 追责机构可以利用追踪密钥还原用户的真实身份. 最后, 本文开展协议实验测评, 基于 Windows 及 Raspberry Pi 4B 平台上进行部署和实现. 测评结果显示, 匿名认证与密钥协商过程耗时均为毫秒级, 充分展示了本协议的高效性与实用性.

本文第 1 节讨论相关工作. 第 2 节介绍本文所使用的密码知识. 第 3 节给出协议的语法定义与安全模型. 第 4 节介绍协议的具体构造以及安全性证明. 第 5 节实验测评, 并进行效率分析. 第 6 节对本文内容进行总结.

1 相关工作

为实现身份认证的同时保护用户的身份信息, Lindell[6]对匿名认证的基础理论与构建匿名互联网的可能性进

行了深入探讨. Yu 等人[7]针对嵌入式移动设备提出一种基于硬件安全特性的匿名认证方案, 通过防止凭据复制与共享, 提高了移动设备认证的安全性. 张金花等人[8]基于区块链提出匿名跨域认证与密钥协商机制, 借助智能合约实现条件可控匿名和密钥的去中心化管理. 刘一丹等人[9]面向车联网, 提出一种改进的无证书匿名认证协议, 降低了对可信中心的依赖和密钥托管风险.

AKA 协议作为通信系统中典型的身份认证与密钥协商协议, 通过共享长期密钥进行身份验证与密钥生成, 虽然对用户唯一标识符进行了加密, 但运营商仍可解密并识别用户身份. 孙中岫等人[10]聚焦 5G 系统广播消息的身份认证问题, 提出了基于无证书签名的 5G 系统广播消息身份认证协议, 该协议能抵御常见攻击, 确保 5G 系统广播消息身份认证安全可靠. Yu 等人[11]进一步提出 AAKA 协议, 在实现服务访问的同时保障用户身份信息的完全匿名性, 从根本上解决了移动追踪问题. Narwal 等人[12]设计了 SAMAKA 协议, 实现无线局域网中的匿名相互认证与密钥协商, 能够有效抵御中间人攻击、重放攻击等, 并实现节点身份的隐藏.

刘志恩[13]将零知识证明与匿名凭证技术融合, 构建高效、安全的身份认证系统. 不仅分析了两者的体系结构与运作机制, 还提出针对融合过程中面临的挑战的解决思路. Zhu 等人[14]引入去中心化身份管理架构, 将身份控制权归还用户, 并引入范围证明技术以支持身份的全生命周期管理. 杨雪等人[15]关注用户体验, 提出双因素认证密钥协商协议, 结合生物特征与密码提升安全性, 同时优化协议流程, 降低用户操作复杂度和等待成本. Mir 等人[16]提出 DAMFA 方案, 结合区块链与多因素认证, 在确保透明性与不可篡改的同时, 保障用户身份隐私.

随着对数据安全和自主可控需求的提升, 越来越多身份认证方案开始引入国密算法, 黄旺旺等人[17]基于 SM9 算法设计了一种可重构密钥安全认证协议, 利用国密算法特性优化运算性能, 减轻设备负担. 赵艳琦等人[18]则在 SM2 数字签名的基础上提出匿名凭证方案, 有效地解决了 SM2 签名算法难以直接应用于匿名凭证场景中的兼容性问题.

2 基础知识

2.1 SM2 签名算法

SM2 数字签名[19]使用固定的有限域 F_p, 在 F_p 域中, 其特征是大素数 p; 对于满足 $y^2 = x^3 + ax + b \bmod p \in F_p$ 且 $4a^3 + 27b^2 \bmod p \neq 0$ 的椭圆曲线, 将其定义为 $E(F_p)$. 然后构建加法循环群 $\mathbb{G}_1 = \{P|P \in E\} \cup \{O\}$, 设 $G = (x_G, y_G)$ 为 $\mathbb{G}_1$ 群中阶为 q 的生成元. 此外, 签名和验证算法都需要使用密码杂凑函数 $H: \{0,1\}^* \times \{0,1\}^* \to Z_q$, SM2 签名算法中规定使用 SM3[20]杂凑函数. SM2 数字签名算法包括密钥生成算法 ($Keygen$)、签名算法 ($Sign$) 以及验证算法 ($Verify$).

(1) 密钥生成算法 $(sk, pk) \leftarrow Keygen(params)$: 输入系统公开参数 $params$, 生成签名密钥 $sk = x$, 验证公钥 $pk = y$. 其中, $x \in Z_q$ 为均匀选取的随机数, $y = xG$.

(2) 签名算法 $(\sigma) \leftarrow Sign(sk, m)$: 输入签名密钥 sk, 待签名的消息 m, 为了生成关于消息 m 的签名 σ. 签名者选取随机数 $k \in [1, q-1]$, 计算椭圆曲线上的点 $kG = (x_1, y_1)$, $r = (H(m) + x_1) \bmod q$, $s = (1 + sk)^{-1} \cdot (k - r \cdot sk) \bmod q$, 最后输出签名 $\sigma = (r, s)$.

(3) 验证算法 $1/0 \leftarrow Verify(pk, (m, \sigma))$: 输入验证密钥 pk, 待验证的消息签名对 (σ, m), 为了验证签名的真实性. 验证者首先计算 $sG + (r + s)\, pk = (x_1, y_1)$, 然后验证等式 $r = (x_1 + H(m)) \bmod q$ 是否成立, 成立则返回 1; 否则返回 0.

2.2 ElGamal 加密算法

ElGamal 密码体制[21]是一种重要的密码体制, 它的安全性基于有限域上的离散对数问题. 其参数定义如下: 含有 p 个元素的有限域 Z_p (其中 p 为素数), $\mathbb{Z}_p^*$ 是 $\mathbb{Z}_p$ 的乘法群, 选取 p 时必须确保 p 足够大以确保乘法群 $\mathbb{Z}_p^*$ 上的离散对数是难以计算的, 在 $\mathbb{Z}_p^*$ 中选择生成元 g, 公开参数 $params = \left(p, \mathbb{Z}_p, \mathbb{Z}_p^*, g\right)$, ElGamal 加密算法包括密钥生成算法 ($Keygen$)、加密算法 ($Enc$)、解密算法 ($Dec$).

(1) 密钥生成算法 $(sk, pk) \leftarrow Keygen(params)$: 输入系统参数 $params$, 生成主私钥和主公钥 (sk, pk). 选取随机数 $a \in Z_p$, 计算 $y = g^a \bmod p$, 令 $sk = a$, $pk = (y, g, p)$.

(2) 加密算法 $(c_1, c_2) \leftarrow Enc(pk, m)$: 输入加密密钥 pk, 消息 m, 生成密文 (c_1, c_2). 选取随机数 $k \in Z_{p-1}$, 计算

$c_1 = g^k \bmod p$, $c_2 = my^k \bmod p$, $c = (c_1, c_2)$.

(3) 解密算法 $m \leftarrow Dec(sk, c)$: 输入解密密钥 sk, 密文消息对 $c = (c_1, c_2)$, 计算 $m = c_2(c_1)^{-1} \bmod p$.

2.3 Diffie-Hellman 密钥协商协议

Diffie-Hellman 密钥协商协议[22]是一种允许双方在不安全的信道上, 安全地交换密钥, 以便后续进行加密通信的方法. 该协议的安全性基于有限域上离散对数问题. 首先, 双方需要共同选择一个大素数 p 和它的本原根 g, 作为公开参数, 然后整个密钥协商分两步完成.

(1) 首先通信双方 (记为 A 和 B) 分别挑选一个需要保密的随机数 X_A 和 X_B, 并分别计算 $Y_A = g^{X_A} \bmod p$ 和 $Y_B = g^{X_B} \bmod p$, 然后双方互相交换.

(2) 通信双方分别计算 $K = Y_B^{X_A} \bmod p$ 和 $K = Y_A^{X_B} \bmod p$, 得到双方共享密钥 K.

2.4 承诺方案

承诺协议[23]使得承诺方能够在不泄露消息 m 的前提下, 向接收方证明自己拥有消息 m, 并将该承诺值发送给接收方, 并在后续阶段不可抵赖地公开该消息. 承诺协议由以下 3 个算法组成.

(1) 初始化算法 $params \leftarrow Setup(1^\lambda)$: 输入安全参数 λ, 生成公开参数 $params$.

(2) 承诺算法 $c \leftarrow Com(params, m)$: 输入公开参数 $params$ 以及需要承诺的消息 m, 承诺方生成关于消息 m 的承诺值 c.

(3) 承诺打开算法 $1/0 \leftarrow Open(params, c, m)$: 以公开参数 $params$, 承诺的消息 m 以及关于消息 m 的承诺值 c 为输入, 验证方验证承诺值 c 是否为关于消息 m 的承诺值. 如果关于消息 m 的承诺有效则输出 1, 否则输出 0.

2.5 零知识证明

零知识证明系统最早由 Goldwasser 等人[24]提出, 证明者的目的是说服验证者相信某个论断是正确的, 证明的过程中不会向验证者透露除证明本身外任何的信息, 即除了验证该论断本身的正确性之外, 验证者无法从证明中获取任何知识. 零知识证明系统一般由以下 3 个算法组成.

(1) 初始化算法 $par \leftarrow Setup(1^\lambda)$: 以安全参数 λ 作为输入, 输出协议的公开参数 par.

(2) 证明算法 $\pi \leftarrow Prove(par, w, x)$: 以公开参数 par, 待证明的陈述 x 以及对应证据 w 为输入, 证明方生成相关证明 π.

(3) 验证算法 $1/0 \leftarrow Verify(par, x, \pi)$: 以公开参数 par, 待验证的陈述 x 以及证明 π 为输入, 验证方验证证明的有效性, 输出 1/0, 其中 1 表示有效; 0 表示无效.

零知识证明系统需满足以下 3 个性质.

- 可靠性: 任何试图伪造的证明者, 在不掌握真实证明知识的情况下, 几乎不可能让验证者接受其伪造的证明.
- 完备性: 如果证明者确实掌握了能够证明命题为真的知识, 且按照既定的零知识证明协议规范进行操作, 那么验证者就应当以极大概率接受证明者给出的证明.
- 零知识性: 验证者除了能够明确证明者所声称的命题是真还是假之外, 无法从证明过程中获取到其他信息.

3 匿名认证与密钥协商的语法与安全模型

3.1 语法定义

本节介绍基于 SM2 的匿名认证与密钥协商协议的算法构成, 其参与方包括归属网络 (HN)、用户 (UE)、服务网络 (SN)、追责机构 (LEA). 算法构成包括初始化算法、密钥生成算法、凭证生成算法、凭证获取算法、密钥协商算法、凭证验证算法、追责算法. 具体如下.

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

(2) $(sk_{\mathrm{HN}}, pk_{\mathrm{HN}}, sk_{\mathrm{UE}}, pk_{\mathrm{UE}}, sk_{\mathrm{SN}}, pk_{\mathrm{SN}}, sk_{\mathrm{LEA}}, pk_{\mathrm{LEA}}) \leftarrow Keygen(params)$ 密钥生成算法: 输入公开参数 $params$, 输出各参与方的公私钥对.

(3) $(r_1, s_1, r_2, s_2) \leftarrow Issue(params, sk, m)$ 凭证生成算法: 用户向归属网络申请凭证, 归属网络为用户生成凭证 (r_1, s_1, r_2, s_2).

(4) $(\pi, c_1, c_2) \leftarrow Get(params, pk_{\mathrm{LEA}}, r_1, s_1, r_2, s_2)$ 凭证获取: 用户输入公开参数 $params$, 追责机构 (LEA) 的公钥 pk_{LEA} 以及归属网络生成的凭证 (r_1, s_1, r_2, s_2), 生成相应的匿名凭证.

(5) $K \leftarrow KeyAgree(params, sk_{\mathrm{UE}}, pk_{\mathrm{UE}}, sk_{\mathrm{SN}}, pk_{\mathrm{SN}})$ 密钥协商算法: 用户与服务网络共同协商出会话密钥 K.

(6) $0/1 \leftarrow Verify(params, pk_{\mathrm{HN}}, \pi)$ 凭证验证算法: 输入公开参数 $params$, 用户的匿名凭证 π, 验证 π 的有效性. 有效, 输出 1; 否则输出 0.

(7) $m \leftarrow Account(sk_{\mathrm{LEA}}, c_1, c_2)$ 追责算法: 输入追责机构的私钥 sk_{LEA} 以及 (c_1, c_2), 输出为用户身份 m.

3.2 安全模型

基于 SM2 的匿名认证与密钥协商协议应该满足匿名性及不可伪造性. 匿名性: 对于任何合法途径生成的凭证, 敌手除了能验证凭证的正确性之外, 无法从凭证中获取任何信息. 不可伪造性: 敌手既不能构造出可通过验证的匿名凭证, 也无法伪造合法用户的有效签名. 为了对这两个安全目标进行形式化定义, 首先定义以下谕言机和列表.

用户列表 H_User: 记录已经询问过匿名凭证的用户标识.

凭证列表 Q_Issue: 记录敌手已经问询过的凭证.

匿名凭证列表 Q_Get: 记录敌手已经询问过的匿名凭证.

注册记录列表 reg: 记录用户的注册信息.

凭证颁发谕言机 $\mathcal{O}.Issue(\cdot)$: 敌手询问标识 i 用户的凭证, 谕言机调用 $Issue$ 算法生成用户 i 的凭证, 并对敌手进行应答, 然后将 i 添加到 H_User, 将凭证添加至 Q_Issue.

凭证获取谕言机 $\mathcal{O}.Get(\cdot)$: 敌手以 SM2 签名作为输入, 谕言机调用 Get 算法生成相应的匿名凭证, 对敌手进行应答, 并将匿名凭证添加至 Q_Get.

匿名凭证谕言机 $\mathcal{O}.Issue_Get(\cdot)$: 包括凭证生成算法 ($Issue$) 以及凭证获取算法 ($Get$). 敌手询问标识 i 用户的匿名凭证, 谕言机调用 $Issue$, Get 算法生成用户 i 的匿名凭证, 并对敌手进行应答, 然后将 i 添加到 H_User, 将匿名凭证添加至 Q_Get.

借助以上谕言机, 定义安全实验: 如图 1 不可伪造性实验所示, 敌手 $\mathcal{A}$ 可以访问谕言机来获取关于特定用户身份的匿名凭证, 其目的是生成一个新的匿名凭证, 且可以通过验证. 如图 2 匿名性实验所示, 敌手 $\mathcal{A}$ 可以访问谕言机来获取关于特定用户身份的匿名凭证, 其目的是区分一个新生成的匿名凭证是关于用户 i_0 还是 i_1 的.

$Exp_{\Pi,\mathcal{A}}^{\mathrm{forge}}(\lambda)$
$(params) \leftarrow Setup(1^{\lambda})$
$(sk_{\mathrm{HN}}, pk_{\mathrm{HN}}, sk_{\mathrm{UE}}, pk_{\mathrm{UE}}, pk_{\mathrm{LEA}}) \leftarrow Keygen(params)$
$\mathcal{O} \leftarrow \{Issue, Get, Issue_Get\}$
$\theta^* \leftarrow \mathcal{A}^{\mathcal{O}}(params, sk_{\mathrm{HN}}, pk_{\mathrm{HN}}, sk_{\mathrm{UE}}, pk_{\mathrm{UE}}, pk_{\mathrm{LEA}})$
如果$Verify(params, \theta^*) = 0$, 返回0; 如果$\exists i \notin H_User$, $\theta^* \notin Q_Get$,
$Verify(params, \theta^*) = 1$返回1; 否则返回0.

图 1 不可伪造性实验

$Exp_{\Pi,\mathcal{A}}^{\mathrm{ano}\text{-}b}(\lambda)$
$(params) \leftarrow Setup(1^{\lambda})$
$(sk_{\mathrm{HN}}, pk_{\mathrm{HN}}, sk_{\mathrm{UE}}, pk_{\mathrm{UE}}, pk_{\mathrm{LEA}}) \leftarrow Keygen(params)$
$\mathcal{O} \leftarrow \{Issue_Get\}$
$(i_0, i_1) \leftarrow \mathcal{A}^{\mathcal{O}}(params, sk_{\mathrm{HN}}, pk_{\mathrm{HN}}, sk_{\mathrm{UE}}, pk_{\mathrm{UE}}, pk_{\mathrm{LEA}})$
$(\theta^*, c_1, c_2) \leftarrow Issue_Get(i_b, params, sk_{\mathrm{HN}})$
$b_1 \leftarrow \mathcal{A}(params, \theta^*, c_1, c_2)$
如果$i_0, i_1 \in H_User$, $Q_Get(i_b) \neq \perp$, $Q_ElG(i_b) \neq \perp$, 返回b_1;
否则, 返回0.

图 2 匿名性实验

定义 1. 在基于 SM2 的匿名认证与密钥协商协议 Π 中, 若任意概率多项式时间敌手 $\mathcal{A}$, 执行实验 $Exp_{\Pi,\mathcal{A}}^{\mathrm{ano}}(\lambda)$ 成功的概率是可忽略的, 则该协议满足匿名性, 敌手的优势为:

$$Adv_{\Pi,\mathcal{A}}^{\mathrm{ano}}(\lambda) = \left|\Pr\left[Exp_{\Pi,\mathcal{A}}^{\mathrm{ano}\text{-}1}(\lambda) = 1\right] - \Pr\left[Exp_{\Pi,\mathcal{A}}^{\mathrm{ano}\text{-}0}(\lambda) = 1\right]\right| \tag{1}$$

定义 2. 在基于 SM2 的匿名认证与密钥协商协议 Π 中, 若任意概率多项式时间敌手 $\mathcal{A}$, 执行实验 $Exp_{\Pi,\mathcal{A}}^{\mathrm{forge}}(\lambda)$ 成

功的概率是可忽略的, 则该协议满足不可伪造性, 敌手的优势为:

$$Adv_{\Pi,\mathcal{A}}^{\text{forge}}(\lambda)=\Pr\left[Exp_{\Pi,\mathcal{A}}^{\text{forge}}(\lambda)=1\right] \tag{2}$$

4 基于 SM2 的匿名认证与密钥协商协议设计

本节构造基于 SM2 的匿名认证与密钥协商协议, 协议具体流程如图 3 所示. ① 生成凭证: 凭证申请阶段, 用户向归属网络发送凭证请求, 然后归属网络验证用户身份的合法性, 然后生成用户的属性信息, 并对属性信息进行签名; ② 凭证颁发: 归属网络向用户颁发凭证; ③ 凭证获取: 用户收到签名后先对签名进行验证, 然后通过零知识证明技术向服务网络证明持有归属网络签署的 SM2 签名; ④ 申请验证: 当用户需要访问服务网络提供的服务时, 将匿名凭证发送给服务网络; ⑤ 验证证明: 服务网络验证用户发送的匿名凭证及其有效期; ⑥ 密钥协商: 服务网络与用户协商用于后期通信的会话密钥; ⑦ 请求追责: 当用户出现违法行为时, 服务网络将用户的加密属性信息发送给追责机构; ⑧ 计算追踪值: 追责机构对用户身份进行恢复; ⑨ 追踪用户: 追责机构将用户身份信息发送给归属网络, 归属网络确定用户真实身份.

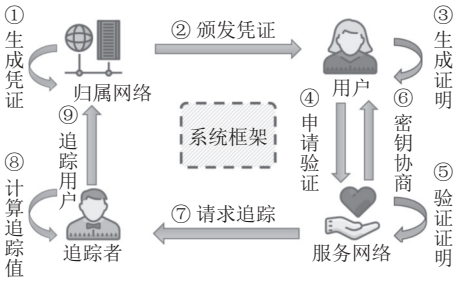


图 3 系统模型图

4.1 协议具体构造

协议由 *Setup* 初始化算法、*Keygen* 密钥生成算法、*Issue* 凭证生成算法、*Get* 凭证获取算法 (如图 4 所示)、*KeyAgree* 密钥协商算法、*Verify* 验证算法 (如图 5 所示)、*Account* 追责算法 (如图 6 所示) 组成, 具体算法如下.

HN		UE
$k_1,k_2\in\mathbb{Z}_q$		
$K_1=k_1G_2=(x_{11},y_{11})$		
$K_2=k_2G_2=(x_{22},y_{22})$		
$r_1=m_1+x_{11}$		
$s_1=(1+x_1)^{-1}(k-r_1x_1)$		
$r_2=m_2+s_1+x_{22}$		
$s_2=(1+x_1)^{-1}(k-r_2x_1)$		
	$\xrightarrow{(m_1,m_2,r_1,s_1,r_2,s_2)}$	
		$r_0\in\mathbb{Z}_q,\ c_1=r_0G_2$
		$c_2=m_1G_2+r_0pk_{\text{LEA}}$
		$\pi\leftarrow ZKPoK\{\pi_a,\pi_b,\pi_c,\pi_d\}$

图 4 凭证颁发与凭证获取

UE		SN
$k_{11}\in\mathbb{Z}_q,\ K_{11}=k_{11}G_2=(x_{12},y_{12})$		
$r_{11}=H_0(K_{11})+x_{12}$		
$s_{11}=(1+x)^{-1}(k_{11}-r_{11}x_2)$		
	$\xrightarrow{(K_{11},r_{11},s_{11},pk_{\text{UE}})}$	
		$Verify(K_{11},r_{11},s_{11},pk_{\text{UE}})$
		$k_{22}\in\mathbb{Z}_q,K_{22}=k_{22}G_2=(x_{22},y_{22})$
		$r_{22}=H_0(K_{22})+x_{22}$
		$s_{22}=(1+x)^{-1}(k_{22}-r_{22}x_3)$
		$K=k_{22}K_{11}$
	$\xleftarrow{(K_{22},r_{22},s_{22},pk_{\text{SN}})}$	
$Verify(K_{11},r_{11},s_{11},pk_{\text{UE}})$		
$K=k_{11}K_{22}$		
	$\xrightarrow{(m_2,c_1,c_2,\pi)}$	
		$Verify(\pi)$

图 5 密钥协商与验证算法

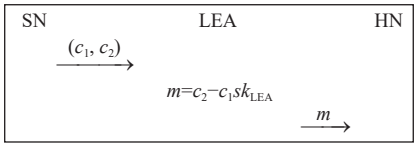


图 6 追责算法

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

1) 按照标准设定 SM2 数字签名算法所需要的参数 p, F_p, q.

2) 构建阶为素数 p 的循环群 $\mathbb{G}_1$, 其生成元为 G_1, $H_1 \in \mathbb{G}_1$, 构造阶为素数 q 的循环群 $\mathbb{G}_2$, 其生成元为 G_2, $H_2 \in \mathbb{G}_2$.

3) 选取 SM3 密码杂凑函数作为默认的密码杂凑函数.

(2) 密钥生成算法: $(pk, sk) \leftarrow Keygen(params)$: 输入公开参数 $params$, 输出各参与方的公私钥对.

1) 归属网络调用 SM2 密钥生成算法, 随机选取 $x_1 \in \mathbb{Z}_q$, 计算 $y_1 = x_1G_2$, 公私钥为 $(sk_{\mathrm{HN}} = x_1, pk_{\mathrm{HN}} = y_1)$.

2) 用户调用 SM2 密钥生成算法, 随机选取 $x_2 \in \mathbb{Z}_q$, 计算 $y_2 = x_2G_2$, 用户公私钥为 $(sk_{\mathrm{UE}} = x_2, pk_{\mathrm{UE}} = y_2)$.

3) 服务网络调用 SM2 密钥生成算法, 随机选取 $x_3 \in \mathbb{Z}_q$, $y_3 = x_3G_2$, 公私钥为 $(sk_{\mathrm{SN}} = x_3, pk_{\mathrm{SN}} = y_3)$.

4) 追责机构调用 ElGamal 密钥生成算法, 随机选取 $x_4 \in \mathbb{Z}_p$ 计算 $h = x_4G_1$, 追责机构公私钥为 $(sk_{\mathrm{LEA}} = x_4, pk_{\mathrm{LEA}} = h)$.

(3) 凭证生成算法 $(r_1, s_1, r_2, s_2) \leftarrow Issue(params, SK_{\mathrm{HN}}, m_1, m_2)$: 用户向归属网络申请凭证, 归属网络为用户生成凭证 (r_1, s_1, r_2, s_2).

1) 归属网络为用户生成属性信息 m_1, $m_2 \in \mathbb{Z}_q$, 其中 m_1 表示用户的身份等需要保密的信息, m_2 表示用户凭证有效期等可以公开的信息.

2) 归属网络生成凭证, 随机选取 $k_1 \in \mathbb{Z}_q$, $k_2 \in \mathbb{Z}_q$ 计算 $K_1 = k_1G_2 = (x_{11}, y_{11})$, $K_2 = k_2G_2 = (x_{22}, y_{22})$. 然后对于用户身份属性信息 m_1 计算 $r_1 = m_1 + x_{11}$, $s_1 = (1 + x_1)^{-1}(k_1 - r_1x_1)$, 对于用户公开属性信息 m_2 计算 $r_2 = m_2 + s_1 + x_{22}$, $s_2 = (1 + x_1)^{-1}(k_2 - r_2x_1)$.

3) 归属网络保存用户的属性信息 m_1, m_2 以及用户的真实身份信息.

(4) 凭证获取算法 $(\pi_a, \pi_b, \pi_c, \pi_d, c_1, c_2) \leftarrow Get(params, pk_{\mathrm{LEA}}, r_1, s_1, r_2, s_2, m_1, m_2)$: 用户首先验证凭证的真实性, 然后生成匿名凭证.

1) 用户利用归属网络的公钥验证凭证的真实性.

2) 用户随机选取 $r_0 \in \mathbb{Z}_q$, 计算 $c_1 = r_0G_2$, $c_2 = m_1G_2 + r_0pk_{\mathrm{LEA}}$.

3) 用户生成证明 π, 证明自己持有 SM2 签名, 以及加密结果的真实性.

4) 用户保存证明 π.

(5) 密钥协商算法 $K \leftarrow KeyAgree(params, sk_{\mathrm{UE}}, pk_{\mathrm{UE}}, sk_{\mathrm{SN}}, pk_{\mathrm{SN}})$: 用户与服务网络协商会话密钥 K.

1) 用户随机选取 $k_{11} \in \mathbb{Z}_q$, 计算 $K_{11} = k_{11}G_2 = (x_{12}, y_{12})$, 并计算 $s_{11} = (1 + x)^{-1}(k_{11} - r_{11}x_2)$, $r_{11} = H_0(K_{11}) + x_{12}$.

2) 服务网络以相同方法计算 K_{22}, r_{22}, s_{22}.

3) 用户与服务网络分别将上述信息发送给对方. 双方分别验证对方签名的真实性, 然后计算会话密钥 $K = k_{11}K_{22} = k_{22}K_{11}$.

(6) 凭证验证算法 $0/1 \leftarrow Verify(params, pk_{\mathrm{HN}}, r_2, s_2, r_3)$: 服务网络验证凭证, 服务网络验证凭证的真实性, 并验证凭证的有效期.

(7) 追责算法 $m_1G_2 \leftarrow Account(sk_{\mathrm{LEA}}, c_1, c_2)$: 输入追责机构的公钥 sk_{LEA} 以及 (c_1, c_2) 输出为用户身份. 追责机构计算 $m = c_2 - c_1SK_{\mathrm{LEA}}$, 然后将 m 发送给归属网络, 归属网络将 m 与保存的用户信息进行对比, 确定用户身份.

此外, 协议支持凭证的更新. 在实际应用场景下, 凭证的时效性至关重要, 当凭证过期时, 本文协议展现出独特的优越性. 归属网络在处理凭证更新过程中, 仅需向用户更新关于有效期 m_2 的签名 (r_2, s_2), 无需对用户标识符 m_1 的签名 (r_1, s_1) 进行重复更新. 在凭证获取阶段, 用户也只需要更新与有效期 m_2 相关部分的承诺与零知识证明. 用户端承诺与零知识证明具体算法在第 4.2 节中进行详细的说明, 与第 4.2 节中的算法相比, 在用户更新凭证过程中

关于消息 m_1 以及签名 (r_1, s_1) 的相关承诺 C_1, C_3, C_5, C_7, C_8, $C_{s_{x_1}}$, $C_{s_{y_1}}$, $C_{u_{x_1}}$, $C_{u_{y_1}}$ 无需再次生成, 只需要重新计算关于 m_2, (r_2, s_2) 相关承诺 C_2, C_4, C_6, $C_{s_{x_2}}$, $C_{s_{y_2}}$, $C_{u_{x_2}}$, $C_{u_{y_2}}$, 以 π_a 为例, 如图 7 所示, 在进行零知识证明时, 只需要计算 S_4, S_5, S_6 然后重新计算 C, 最后再重新计算 $s_7,\ldots,s_{12}$ 而无需重新计算 S_1, S_2, S_3 以及 $s_1,\ldots,s_6$, 其余零知识证明过程与 π_a 类似. 这大大简化了操作流程, 降低了系统的处理负担和资源消耗. 这一过程不仅提高了证明生成的效率, 而且减少了计算量, 使得整个凭证更新与证明生成过程更加简洁、快速, 为保障系统的高效运行和用户体验提供了坚实支撑.

$$
\begin{array}{ll}
\textit{Prover}: & \textit{Verifier}: \\
a_1,a_2,\ldots,a_{12}\in\mathbb{Z}_q & \\
S_1=a_1G_2+a_2H_2, S_2=a_3G_1+a_4H_1 & \\
S_3=a_9G_1+a_{10}H_1, S_6=a_{11}G_1+a_{12}H_1 & \\
S_4=a_5G_2+a_6H_2, S_5=a_7G_1+a_8H_1 & \\
C=H_0\left(S_1,S_2,S_3,S_4,S_5,S_6,C_3,C_4,C_{s_{x1}},C_{s_{y1}},C_{s_{x2}},C_{s_{y2}}\right) & \\
s_1=s_1C+a_1, s_2=R_3C+a_2 & \\
s_3=s_{x_1}C+a_3, s_4=R_{s_{11}}C+a_4 & \\
s_5=s_yC+a_5, s_6=R_{s_{12}}C+a_6 & \\
s_7=s_2C+a_7, s_8=R_4C+a_8 & \\
s_9=s_{x_{21}}C+a_9, s_{10}=R_{s_{21}}C+a_{10} & \\
s_{11}=s_{y_{21}}C+a_{11}, s_{12}=R_{s_{12}}C+a_{12} & \\
\xrightarrow{\pi_a=(s_1,s_2,\ldots,s_{12},C)} & \\
 & C=H_0\Big(s_1G_2+s_2H_2-CC_3, s_3G_1+s_4H_1-CC_{s_{x_1}} \\
 & s_5G_1+s_6H_1-CC_{s_{y1}}, s_7G_2+s_8H_2-CC_4 \\
 & s_9G_1+s_{10}H_1-CC_{s_{x2}}, s_{11}G_1+s_{12}H_1-CC_{s_{y2}} \\
 & C,C_{s_{y2}},C_3,C_4,C_{s_{x1}},C_{s_{y1}},C_{s_{x2}},C_{s_{y2}}\Big)
\end{array}
$$

图 7 π_a 计算方法

4.2 正确性分析

协议正确性: 基于 SM2 的匿名认证与密钥协商协议的正确性源于 SM2 签名的正确性、ElGamal 加密算法的正确性以及 *ZKPoK* 证明的完备性. 具体过程如下.

首先, 凭证获取阶段, 用户首先需要调用归属网络的公钥, 对接收到的 SM2 签名进行验证, 这一过程首先计算 $s_1G_2+(r_1+s_1)pk_{\mathrm{HN}}=(x_{11},y_{11})$, 同时验证 $r_1=m_1+x_{11} \bmod q$, 然后计算 $s_2G_2+(r_2+s_2)pk_{\mathrm{HN}}=(x_{22},y_{22})$, 同时验证 $r_2=m_2+s_1+x_{22} \bmod q$ 验证成功返回 1; 否则返回 0. 验证过程如下:

$$s_1G_2+(r_1+s_1)y=\left[\frac{k_1-r_1x_1}{1+x_1}+\frac{r_1x_1+r_1x_1^2}{1+x_1}+\frac{k_1x_1-r_1x_1^2}{1+x_1}\right]G_2=\frac{k_1+k_1x_1}{1+x_1}G_2=k_1G_2 \tag{3}$$

$$s_2G_2+(r_2+s_2)y=\left[\frac{k_2-r_2x_1}{1+x_1}+\frac{r_2x_1+r_2x_1^2}{1+x_1}+\frac{k_2x_1-r_2x_1^2}{1+x_1}\right]G_2=\frac{k_2+k_2x_1}{1+x_1}G_2=k_2G_2 \tag{4}$$

由此可验证 $r_1=m_1+x_{11} \bmod q$, $r_1=m_2+s_1+x_{22} \bmod q$.

其次, 在凭证获取阶段, 用户借助零知识证明手段对自身拥有归属网络所签署的 SM2 签名 r_1, s_1, r_2, s_2 以及 c_1, c_2 的正确性进行证明. 具体操作如下.

为了生成相关零知识证明, 首先对于 (r_1,s_1,r_2,s_2) 用户计算承诺 (r_1,s_1,r_2,s_2), $C_1=r_1G_2+R_1H_2$, $C_2=r_2G_2+R_2H_2$, $C_3=s_1G_2+R_3H_2$, $C_4=s_2G_2+R_4H_2$, 同时对于 $u_1=(r_1+s_1)$, $u_2=(r_2+s_2)$, 以及 $s_1G_2=(s_{x_1},s_{y_1})$, $s_2G_2=(s_{x_1},s_{y_1})$, $u_2y_1=(u_{x_2},u_{y_2})$, $u_1y_1=(u_{x_1},u_{y_1})$, y_1 用户需计算承诺 $C_5=u_1G_2+R_5H_2$, $C_6=u_2G_2+R_6H_2$, $C_7=r_0G_2$, $C_8=m_1G_2+r_0PK_{\mathrm{LEA}}$, $C_{s_{x_1}}=s_{x_1}G_1+R_{s_{11}}H_1$, $C_{s_{y_1}}=s_{y_1}G_1+R_{s_{12}}H_1$, $C_{s_{x_2}}=s_{x_2}G_1+R_{s_{21}}H_1$, $C_{s_{y_2}}=s_{y_2}G_1+R_{s_{22}}H_1$, $C_{u_{x_1}}=u_{x_1}G_1+R_{u_{11}}H_1$, $C_{u_{y_1}}=u_{y_1}G_1+R_{u_{12}}H_1$, $C_{u_{x_2}}=u_{x_2}G_1+R_{u_{21}}H_1$, $C_{u_{y_2}}=u_{y_2}G_1+R_{u_{22}}H_1$, $C_9=m_1G_2+R_9H_2$ 生成证明 π_a, π_b, π_c, π_d.

$$\begin{aligned}\pi_a \leftarrow ZKPoK1.Prove\{&(s_1,s_2,s_{x_1},s_{y_1},s_{x_2},s_{y_2},R_3,R_4,R_{s_{11}},R_{s_{12}},R_{s_{21}},R_{s_{22}}):\\ &C_3=s_1G_2+R_3H_2\wedge C_{s_{x_1}}=s_{x_1}G_1+R_{s_{11}}H_1\wedge C_{s_{y_1}}=s_{y_1}G_1+R_{s_{12}}H_1\wedge\\ &C_4=s_2G_2+R_4H_2\wedge C_{s_{x_2}}=s_{x_2}G_1+R_{s_{21}}H_1\wedge C_{s_{y_2}}=s_{y_2}G_1+R_{s_{22}}H_1)\}\end{aligned} \tag{5}$$

$$\begin{aligned}\pi_b \leftarrow ZKPoK2.Prove\{&(u_1,u_2,u_{x_1},u_{y_1},u_{x_2},u_{y_2},R_5,R_6,R_{u_{11}},R_{u_{12}},R_{u_{21}},R_{u_{22}}):\\ &C_5=u_1G_2+R_5H_2\wedge C_{u_{x_1}}=u_{x_1}G_1+R_{u_{11}}H_1\wedge C_{u_{y_1}}=u_{y_1}G_1+R_{u_{12}}H_1\wedge\\ &C_6=u_2G_2+R_6H_2\wedge C_{u_{x_2}}=u_{x_2}G_1+R_{u_{21}}H_1\wedge C_{u_{y_2}}=u_{y_2}G_1+R_{u_{22}}H_1\}\end{aligned} \tag{6}$$

$$\begin{aligned}\pi_c \leftarrow ZKPoK3.Prove\{&(r_1,r_2,s_1,s_2,u_1,u_2,s_{x_1},s_{y_1},s_{x_2},s_{y_2},u_{x_1},u_{y_1},u_{x_2},u_{y_2}):\\ &C_1=r_1G_2+R_1H_2\wedge C_2=r_2G_2+R_2H_2\wedge C_3=s_1G_1+R_3H_1\wedge C_7=r_0G_2\wedge\\ &C_8=m_1G_2+r_0PK_{LEA}\wedge C_9=m_1G_2+R_9H_2\wedge C_{s_{x_1}}=s_{x_1}G_1+R_{s_{11}}H_1\wedge\\ &C_{s_{y_1}}=s_{y_1}G_1+R_{s_{12}}H_1\wedge C_4=s_2G_2+R_4H_2\wedge C_{s_{x_2}}=s_{x_2}G_1+R_{s_{21}}H_1\wedge\\ &C_{s_{y_2}}=s_{y_2}G_1+R_{s_{22}}H_1\wedge C_{u_{x_1}}=u_{x_1}G_1+R_{u_{11}}H_1\wedge C_{u_{y_1}}=u_{y_1}G_1+R_{u_{12}}H_1\wedge\\ &C_6=u_2G_2+R_6H_2\wedge C_{u_{x_2}}=u_{x_2}G_1+R_{u_{21}}H_1\wedge C_{u_{y_2}}=u_{y_2}G_1+R_{u_{22}}H_1\wedge\\ &r_1=m_1+((u_{x_1},u_{y_1})+(s_{x_1},s_{y_1}))_x\wedge r_2=m_2+s_1+((u_{x_2},u_{y_2})+(s_{x_2},s_{y_2}))_x\}\end{aligned} \tag{7}$$

$$\begin{aligned}\pi_d \leftarrow ZKPoK4.Prove\{&(r_1r_2,s_1,s_2,u_1,u_2,R_1,R_2,R_3,R_4,R_5,R_6):\\ &C_1=r_1G_2+R_1H_2\wedge C_3=s_1G_2+R_3H_2\wedge C_5=u_1G_2+R_5H_2\\ &C_2=r_2G_2+R_2H_2\wedge C_4=s_1G_2+R_4H_2\wedge C_6=u_2G_2+R_6H_2\}\end{aligned} \tag{8}$$

图 7 展示了 π_a 的具体计算方法, π_b, π_c, π_d 的展开可采用相同方法, 用户计算证明 π, 发送给服务网络, 然后服务网络调用验证算法验证 π 的真实性, 如果验证有效, 输出 1; 否则, 输出 0.

4.3 安全性证明

定理 1. 匿名性. 如果 ElGamal 加密算法的密文满足不可区分性, 同时 *ZKPoK* 证明系统具备零知识性, 那么协议可以满足匿名性安全目标.

证明: 敌手 $\mathcal{A}$ 与模拟者 $\mathcal{S}$ 进行交互, $\mathcal{A}$ 可以访问 $\mathcal{O}.Issue(\cdot)$、$\mathcal{O}.Get(\cdot)$、$\mathcal{O}.Issue_Get(\cdot)$ 谕言机. $\mathcal{A}$ 的目的是从匿名凭证中猜测出用户的真实身份. $\mathcal{A}$ 可以进行如下谕言机问询.

$\mathcal{O}.Issue(i)$: $\mathcal{S}$ 与 $\mathcal{A}$ 进行交互, $\mathcal{A}$ 对标识 i 的凭证进行询问. 如果 $i\in H_User$, $\mathcal{S}$ 查询 Q_Issue 列表, 返回 (r_1,s_1,r_2,s_2). 如果 $i\notin H_User$, 则调用 *Issue* 算法生成 (r_1,s_1,r_2,s_2) 进行应答, 将结果添加到 Q_Issue 列表.

$\mathcal{O}.Get(i)$: $\mathcal{A}$ 对 (r_1,s_1,r_2,s_2) 的匿名凭证进行询问. $\mathcal{S}$ 调用 *Get* 算法生成关于 (r_1,s_1,r_2,s_2) 的零知识证明 $\theta=(\pi_a,\pi_b,\pi_c,\pi_d)$, 添加 θ 到 Q_Get, 并对 $\mathcal{A}$ 进行应答.

$\mathcal{O}.Issue_Get(i)$: $\mathcal{A}$ 对标识 i 的匿名凭证进行询问. 如果 $i\in H_User$, $\mathcal{S}$ 查询 $Q_Get(i)$ 列表, 返回 $\theta=(\pi_a,\pi_b,\pi_c,\pi_d)$. 如果 $i\notin H_User$, 则调用 *Issue* 与 *Get* 算法生成 $\theta=(\pi_a,\pi_b,\pi_c,\pi_d)$ 进行应答. 将 i 添加到 H_User 列表, 将 θ 添加到 Q_Get.

根据谕言机问询, $\mathcal{A}$ 输出 i_0, i_1. 如果 $Q_Issue(i), Q_Get(i)\notin\perp$, $\mathcal{S}$ 选择 i_b^*, 计算身份 i_b^* 的匿名凭证 $\theta^*=(\pi_a,\pi_b,\pi_c,\pi_d,c_1,c_2)$, $\mathcal{A}$ 根据 $\mathcal{S}$ 的应答输出猜测 b^*. 由于 *ZKPoK* 的零知识性, 并且 ElGamal 是 CPA 安全的[25], 因此不会向 $\mathcal{A}$ 泄露关于用户的身份信息. 综上所述, 设计的协议满足匿名性.

定理 2. 不可伪造性. 如果 SM2 数字签名算法具有不可伪造性, 那么协议满足不可伪造性.

证明: 敌手 $\mathcal{A}$ 与模拟者 $\mathcal{S}$ 进行交互, $\mathcal{S}$ 可以访问 $\mathcal{O}.Issue(\cdot)$、$\mathcal{O}.Get(\cdot)$、$\mathcal{O}.Issue_Get(\cdot)$ 谕言机. $\mathcal{A}$ 的目的是根据已知的匿名凭证伪造出新标识 i 上的匿名凭证. $\mathcal{A}$ 可以进行如下谕言机问询.

$\mathcal{O}.Issue(i)$: $\mathcal{S}$ 与 $\mathcal{A}$ 进行交互, $\mathcal{A}$ 对标识 i 的凭证进行询问. 如果 $i\in H_User$, 挑战者查询 Q_Issue 列表, 返回 (r_1,s_1,r_2,s_2). 如果 $i\notin H_User$, 则调用 *Issue* 算法生成 (r_1,s_1,r_2,s_2) 进行应答, 将结果添加到 Q_Issue 列表.

$\mathcal{O}.Get(i)$: $\mathcal{A}$ 询问 (r_1,s_1,r_2,s_2) 对应的零知识证明. $\mathcal{S}$ 调用 *Get* 算法生成 $\theta=(\pi_a,\pi_b,\pi_c,\pi_d)$, 添加 θ 到 Q_Get, 并对 $\mathcal{A}$ 进行应答.

$\mathcal{O}.Issue_Get(i)$: $\mathcal{A}$ 询问关于标识 i 的匿名凭证. 如果 $i \in H_User$, $\mathcal{S}$ 查询 $Q_Get(i)$ 列表, 返回 $\theta = (\pi_a, \pi_b, \pi_c, \pi_d)$. 如果 $i \notin H_User$, 则调用 $Issue$ 与 Get 算法生成 $\theta = (\pi_a, \pi_b, \pi_c, \pi_d)$ 进行应答. 将 i 添加到 H_User 列表, 将 θ 添加到 Q_Get.

根据谕言机问询结果, $\mathcal{A}$ 输出新标识 i 上的匿名凭证 $\theta^* = (\pi_a, \pi_b, \pi_c, \pi_d, c_1, c_2)$, 如果 $\exists i \notin H_User$, $\theta^* \notin Q_Get$, $Verify(params, \theta^*) = 1$, 那么则认为 $\mathcal{A}$ 伪造成功. 如果 $\mathcal{A}$ 能够伪造出可以通过验证的 $(\pi_a, \pi_b, \pi_c, \pi_d)$, 那么 $\mathcal{A}$ 可以伪造出 (r_1, s_1, r_2, s_2) 这样的 SM2 数字签名对. 由于 SM2 数字签名的安全性依赖于离散对数问题的困难性[26], 所以 $\mathcal{A}$ 无法伪造出新标识上的签名 (r_1, s_1, r_2, s_2), 因此也无法伪造出新标识上的匿名凭证 $\theta^* = (\pi_a, \pi_b, \pi_c, \pi_d, c_1, c_2)$. 综上所述, 设计的协议满足不可伪造性.

5 性能分析

本节对本文设计的协议进行实现, 并与现有方案文献 [13–16] 进行功能性, 通信开销以及计算开销的对比分析. 最后得出结论, 本文协议在综合性能上具有一定的优势.

5.1 计算开销

本节将基于 SM2 的匿名认证与密钥协商协议从功能性和计算开销方面与已有身份认证与密钥协商协议 (文献 [13–16]) 进行对比评估. 功能性对比结果如表 1 所示. 表 1 列出了各方案的功能特征. 文献 [13,14] 不能提供用户身份追踪功能. 与本协议相比, 文献 [13–16] 均不具备凭证更新能力. 本文协议在保证匿名性的同时, 实现有效凭证更新和非法用户的追责.

表 1 功能性对比

特性	本文方案	文献[13]	文献[14]	文献[15]	文献[16]
匿名性	√	√	√	√	√
凭证更新	√	×	×	×	×
非交互式	√	√	√	√	√
可追责性	√	×	×	√	√

接下来对凭证获取, 凭证验证两个步骤的计算开销进行对比分析, 考虑双线性配对运算、加法群上的标量乘法运算、乘法群上的乘法运算、幂运算和哈希操作对方案性能的影响, 忽略其他占比少且不影响对比结果的运算. 针对本文协议, 借助 Python 语言进行实现, 并分别将协议部署在 Windows 11 和 Raspberry Pi 4B 上, 其中 Windows 系统的配置为 12th Gen Intel(R) Core(TM) i5-12600KF 处理器, 3.7 GHz, 运行内存 32 GB. Raspberry Pi 4B 的配置为 Cortex-A72 处理器, 1.5 GHz, 运行内存 4 GB. 计算开销符号定义与运行时间如表 2 所示, 各步骤计算开销如表 3 所示.

表 2 计算开销符号定义与运行时间 (ms)

符号	定义	Raspberry Pi 4B	Windows 11
H_1	哈希函数 (SHA-1)	0.063	0.015
H_2	哈希函数 (SM3)	2.174	0.511
A	SM2椭圆曲线群上的点加法	0.184	0.043
M	SM2椭圆曲线群上的点乘法	6.18	1.645
E_T	群 G_T 上的指数运算	5.912	1.38
E_1	群 G_1 上的指数运算	22.368	12.86
M_T	群 G_T 上的乘法运算	0.064	0.021
M_1	群 G_1 上的乘法运算	23.236	12.012
P	双线性配对运算	37.072	7.451

从实验结果来看, 文献 [13–16] 所设计的方案在凭证的获取与验证阶段的时间消耗主要来源于双线性配对以及指数运算, 本文协议凭证获取与验证阶段的时间消耗主要来源于 SM2 椭圆曲线群上的加法与乘法运算, 因此本

文协议在运算效率及协议流程优化等方面展现出显著优势.

本文将归属网络、服务网络、追责机构均部署在 Windows 系统上, 将用户端分别部署在 Windows 和 Raspberry Pi 4B (Raspberry Pi 4B 的计算能力与移动设备较为相近) 系统上, 经过多次运行计算平均运行时间. 在 Windows 还是 Raspberry Pi 4B 上, 程序的运行时间均为毫秒级. 计算开销结果如表 4 所示, 其中 SAAKA1 表示本文协议在 Windows 系统上初次生成凭证的计算开销, SAAKA2 表示本文协议在 Windows 系统上更新凭证的计算开销, SAAKA3 表示本文协议在 Raspberry Pi 4B 系统上初次生成凭证的计算开销, SAAKA4 表示本文协议在 Raspberry Pi 4B 系统上更新凭证的计算开销.

表 3　计算复杂度对比

方案	凭证获取	凭证验证
文献[13]	$7E_1+3M_1+2H_1+3E_T+M_T+3P$	$3E_1+2M_1+H_1+6E_T+5P$
文献[14]	$4E_1+3M_1+6E_T+6M_T+5P$	$4E_1+3M_1+4E_T+4M_T+5P$
文献[15]	$2H_1+6E_1+3M_1$	$2H_1+4E_1+5E_T+4M_T+6P$
文献[16]	$3H_1+12E_1+4M_1+3P+2M_T+3E_T$	$7E_1+3M_1+5E_T+4M_T+5P$
本文协议	$53M+24A+4H_2$	$47M+31A+4H_2$

表 4　计算开销对比 (ms)

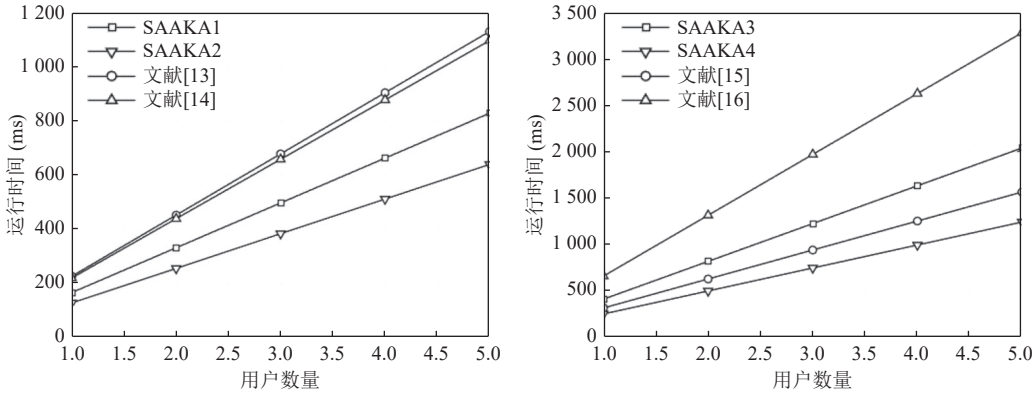
方案	凭证获取	凭证验证	总耗时
SAAKA1	92.3	73.0	165.3
SAAKA2	54.7	73.16	127.16
文献[13]	140.2	85.4	225.6
文献[14]	130.5	88.4	218.9
SAAKA3	337.57	70.32	407.89
SAAKA4	175.63	72.51	248.14
文献[15]	209.38	103.45	312.83
文献[16]	487.91	168.75	656.66

在系统性能测试过程中, 针对凭证相关操作的耗时情况进行了详细统计. 凭证生成阶段, 整体平均耗时为 7.2 ms. 在 Windows 系统环境下, 初次生成凭证耗时 92.3 ms, 而更新凭证操作则耗时 54.7 ms. 凭证验证操作耗时约为 73 ms. 同时, 在数据处理过程中, 追责机构解密操作耗时 0.51 ms, HN 追责操作耗时 1.37 ms. 为进一步测试系统在不同硬件平台上的性能表现, 将 UE 部署在 Raspberry Pi 4B 设备上. 结果显示, 在此设备上进行凭证获取操作所耗费的时间约为 Windows 系统的 4 倍. 具体而言, 初次凭证获取耗时达 337.57 ms, 更新凭证耗时 175.63 ms. 相较于 Windows 系统, Raspberry Pi 4B 在凭证获取等操作上耗时有所增多, 但其耗时依旧维持在毫秒级. 这表明即使是在性能相对较弱的硬件平台上, 系统仍能在可接受的时间范围内完成相关操作.

除此之外, 本文复现了文献 [13,14] 所设计的方案, 并将其部署在 Windows 系统上与 SAAKA1 和 SAAKA2 进行对比, 同时复现了文献 [15,16] 所设计的方案, 并将其部署在 Raspberry Pi 4B 系统上与 SAAKA3 和 SAAKA4 进行对比. 由后文图 8(a) 可知, 在 Windows 系统上部署本协议以及文献 [13,14] 时, 无论是初次获取凭证, 还是更新凭证并验证凭证, 本协议的耗时均小于文献 [13,14]. 由图 8(b) 可以看出, 在 Raspberry Pi 4B 系统上部署本协议以及文献 [15,16] 时, 无论是初次获取凭证, 还是更新凭证并验证凭证, 本协议的耗时均小于文献 [16]. 虽然在初次凭证获取并验证凭证时, 本协议的耗时略高于文献 [15]. 然而, 本协议支持凭证更新, 对于单一用户而言, 再次获取有效凭证的耗时小于文献 [15].

5.2 通信开销

为了方便比较, 设定文献 [13−16] 中循环群 G_1、G_2和G_T 中元素长度均为 1 024 bit, 本文协议中循环群 G_2 的元素长度也为 1 024 bit, 使用 $|G|$ 表示 1 024 bit 的数据, q和$Hash$ 的长度均为 160 bit, 使用 $|q|$ 表示长度为 160 bit 的数据.



(a) Windows系统 (b) Raspberry Pi 4B系统

图 8 不同系统计算开销对比

如表 5 所示, 本文协议出于对用户身份信息的匿名化处理, 以及减少交互次数的考虑进行了签名的零知识证明, 增加了通信开销, 但通信开销仍略低于文献 [13,16]. 虽然本文协议通信开销高于文献 [14,15], 但与文献 [14,15] 相比, 本文协议在凭证获取与展示阶段只需交互 2 次, 而文献 [14] 需要交互 5 次, 文献 [15] 则需交互 7 次.

表 5 通信开销

方案	凭证获取	凭证验证	总开销 (KB)
文献[13]	$12\|G\|+3\|q\|$	$12\|G\|+2\|q\|$	22.31
文献[14]	$6\|G\|$	$9\|G\|$	15.36
文献[15]	$4\|G\|+4\|q\|$	$6\|G\|+6\|q\|$	11.84
文献[16]	$12\|G\|+3\|q\|$	$10\|G\|+8\|q\|$	24.29
本文协议	$6\|G\|$	$4\|G\|+68\|q\|$	21.12

6 总 结

针对移动通信中的隐私泄露问题, 本文提出了基于 SM2 的匿名认证与密钥协商协议, 实现隐私保护与有效监管的平衡. 不仅为用户提供了高度安全和私密的通信环境, 确保了用户在移动通信过程中的匿名性, 并通过引入追责机构作为监管力量, 在用户发生违法行为时对用户的身份进行恢复. 此外, 协议还支持凭证的更新, 且凭证更新耗时显著减少. 最后, 将协议在 Windows 与 Raspberry Pi 4B 系统上进行部署, 耗时均在毫秒级, 体现了协议的适用性与高效性. 在未来工作中, 将进一步考虑协议在实际密码硬件设备中的应用需求, 逐步拓展至 FPGA、国密超级 SIM 卡等硬件设备开展性能评估与适配工作, 以测试协议在不同硬件环境下的表现, 为协议的实际应用提供更全面的技术保障.

References

[1] Zhong YY, Huang JQ. Research on user privacy protection strategies and network security in the big data environment. Digital Communication World, 2025(1): 46–48 (in Chinese with English abstract). [doi: 10.3969/J.ISSN.1672-7274.2025.01.014]

[2] Fei T, Wang WY. The vulnerability and enhancement of AKA protocol for mobile authentication in LTE/5G networks. Computer Networks, 2023, 228: 109685. [doi: 10.1016/J.COMNET.2023.109685]

[3] Xiao YL, Gao S. 5GAKA-LCCO: A secure 5G authentication and key agreement protocol with less communication and computation overhead. Information, 2022, 13(5): 257. [doi: 10.3390/info13050257]

[4] Han XX, Zhou WA, Han Z. Vulnerability analysis and verification of 5G-AKA authentication mechanism. Computer Engineering & Science, 2024, 46(12): 2149–2157 (in Chinese with English abstract). [doi: 10.3969/j.issn.1007-130X.2024.12.007]

[5] Zamani A, Skoglund M. Variable-length coding with zero and non-zero privacy leakage. Entropy, 2025, 27(2): 124. [doi: 10.3390/

e27020124]

[6] Lindell Y. Anonymous authentication. Journal of Privacy and Confidentiality, 2011, 2(2): 35–63. [doi: 10.29012/jpc.v2i2.590]

[7] Yu P, Ni W, Yu GS, Zhang H, Liu RP, Wen QY. Efficient anonymous data authentication for vehicular ad hoc networks. Security and Communication Networks, 2021, 2021: 6638453. [doi: 10.1155/2021/6638453]

[8] Zhang JH, Li XW, Zeng X, Zhao YQ, Duan R, Yang DQ. Cross domain authentication and key agreement protocol based on blockchain in edge computing environment. Journal of Cyber Security, 2021, 6(1): 54–61 (in Chinese with English abstract). [doi: 10.19363/J.cnki.cn10-1380/tn.2021.01.05]

[9] Liu YD, Ma YL, Du YB, Cheng QF. A certificateless anonymous authentication key agreement protocol for VANET. Netinfo Security, 2024, 24(7): 983–992 (in Chinese with English abstract). [doi: 10.3969/j.issn.1671-1122.2024.07.001]

[10] Sun ZX, Peng C, Fan W. System broadcast information authentication protocol based on certificateless signature for 5G network. Netinfo Security, 2024, 24(8): 1220–1230 (in Chinese with English abstract). [doi: 10.3969/j.issn.1671-1122.2024.08.008]

[11] Yu HX, Du CL, Xiao Y, Keromytis A, Wang CG, Gazda R, Hou YT, Lou WJ. AAKA: An anti-tracking cellular authentication scheme leveraging anonymous credentials. In: Proc. of the 2024 Network and Distributed System Security Symp. San Diego: Internet Society, 2024. [doi: 10.14722/ndss.2024.24617]

[12] Narwal B, Mohapatra AK. SAMAKA: Secure and anonymous mutual authentication and key agreement scheme for wireless body area networks. Arabian Journal for Science and Engineering, 2021, 46(9): 9197–9219. [doi: 10.1007/s13369-021-05707-3]

[13] Liu ZE. Research on identity authentication management system based on zero-knowledge proof and anonymous credentials [MS. Thesis]. Jinan: Shandong Normal University, 2024 (in Chinese with English abstract). [doi: 10.27280/d.cnki.gsdsu.2024.000879]

[14] Zhu XJ, He DB, Bao ZJ, Luo M, Peng C. An efficient decentralized identity management system based on range proof for social networks. IEEE Open Journal of the Computer Society, 2023, 4: 84–96. [doi: 10.1109/OJCS.2023.3258188]

[15] Yang X, Liu YJ, Jiang Q, Wang JH, Li XH. User-centric two-factor authentication key agreement protocol. Ruan Jian Xue Bao/Journal of Software, 2024, 35(10): 4859–4875 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6966.htm [doi: 10.13328/j.cnki.jos.006966]

[16] Mir O, Roland M, Mayrhofer R. DAMFA: Decentralized anonymous multi-factor authentication. In: Proc. of the 2nd ACM Int'l Symp. on Blockchain and Secure Critical Infrastructure. Taipei: Association for Computing Machinery, 2020. 10–19. [doi: 10.1145/3384943.3409417]

[17] Huang WW, Zhou H, Wang DQ, Zhao Q. Design of reconfigurable key security authentication protocol for IoT based on national cryptography SM9. Netinfo Security, 2024, 24(7): 1006–1014 (in Chinese with English abstract). [doi: 10.3969/j.issn.1671-1122.2024.07.003]

[18] 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 with English abstract). http://www.jos.org.cn/1000-9825/6929.htm [doi: 10.13328/j.cnki.jos.006929]

[19] 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).

[20] National Cryptography Administration. GM/T 0004-2012 SM3 cryptographic hash algorithm. Beijing: China Standard Press, 2012 (in Chinese).

[21] ElGamal T. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. on Information Theory, 1985, 31(4): 469–472. [doi: 10.1109/TIT.1985.1057074]

[22] Diffie W, Hellman M. New directions in cryptography. IEEE Trans. on Information Theory, 1976, 22(6): 644–654. [doi: 10.1109/TIT.1976.1055638]

[23] 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]

[24] 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: Association for Computing Machinery, 1985. 291–304. [doi: 10.1145/22145.22178]

[25] Sow D, Camara GM. Provable security of the generalized ElGamal signature scheme. Journal of Mathematics Research, 2019, 11(6): 77. [doi: 10.5539/jmr.v11n6p77]

[26] Yang A, Nam J, Kim M, Choo KKR. Provably-secure (Chinese Government) SM2 and simplified SM2 key exchange protocols. The Scientific World Journal, 2014, 2014: 825984. [doi: 10.1155/2014/825984]

附中文参考文献

[1] 钟煜焱, 黄鉴泉. 大数据环境下用户隐私保护与网络安全策略研究. 数字通信世界, 2025(1): 46–48. [doi: 10.3969/J.ISSN.1672-7274.2025.01.014]

[4] 韩晓璇, 周文安, 韩震. 5G-AKA 认证机制脆弱性分析与验证. 计算机工程与科学, 2024, 46(12): 2149–2157. [doi: 10.3969/j.issn.1007-130X.2024.12.007]

[8] 张金花, 李晓伟, 曾新, 赵榆琴, 段燃, 杨邓奇. 边缘计算环境下基于区块链的跨域认证与密钥协商协议. 信息安全学报, 2021, 6(1): 54–61. [doi: 10.19363/J.cnki.cn10-1380/tn.2021.01.05]

[9] 刘一丹, 马永柳, 杜宜宾, 程庆丰. 一种车联网中的无证书匿名认证密钥协商协议. 信息网络安全, 2024, 24(7): 983–992. [doi: 10.3969/j.issn.1671-1122.2024.07.001]

[10] 孙中岫, 彭诚, 范伟. 基于无证书签名的 5G 系统广播消息身份认证协议. 信息网络安全, 2024, 24(8): 1220–1230. [doi: 10.3969/j.issn.1671-1122.2024.08.008]

[13] 刘志恩. 基于零知识证明和匿名凭证的身份认证管理系统研究 [硕士学位论文]. 济南: 山东师范大学, 2024. [doi: 10.27280/d.cnki.gsdsu.2024.000879]

[15] 杨雪, 刘怡静, 姜奇, 王金花, 李兴华. 以用户为中心的双因子认证密钥协商协议. 软件学报, 2024, 35(10): 4859–4875. http://www.jos.org.cn/1000-9825/6966.htm [doi: 10.13328/j.cnki.jos.006966]

[17] 黄旺旺, 周骅, 王代强, 赵麒. 基于国密 SM9 的物联网可重构密钥安全认证协议设计. 信息网络安全, 2024, 24(7): 1006–1014. [doi: 10.3969/j.issn.1671-1122.2024.07.003]

[18] 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 [doi: 10.13328/j.cnki.jos.006929]

[19] 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.

[20] 国家密码管理局. GM/T 0004-2012 SM3 密码杂凑算法. 北京: 中国标准出版社, 2012.

作者简介

赵璇, 硕士生, 主要研究领域为公钥密码学, 国密算法.

赵艳琦, 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.

孙铭鸿, 硕士生, 主要研究领域为公钥密码学, 隐私增强技术.

禹勇, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为密码学, 数据安全, 区块链安全.