

面向元宇宙的数字公民身份认证方案

吕秋云¹, 程绍鹏¹, 杨满智^{2,3}, 陈晓光^{2,3}, 王震^{1,4}

(1. 杭州电子科技大学网络空间安全学院, 浙江 杭州 310018; 2. 澳门科技大学, 澳门 999078;
3. 恒安嘉新(北京)科技股份有限公司, 北京 100086; 4. 杭州电子科技大学卓越学院, 浙江 杭州 310018)

摘要: 元宇宙作为虚实融生的三元世界, 为人们提供了海量多样化的网络服务, 但由于缺乏现实世界的面对面交流且无法打破地理与时间的限制, 现有方案无法在高效认证数字公民的同时实现隐私保护与监管。基于此, 提出一种面向元宇宙的数字公民身份认证方案。依据身份时效性设计分层认证方案, 基于 Paillier 同态加密和保持结构性签名等算法实现去中心化标识符与可验证凭证的双重不可链接, 并借助代码嵌入技术保障公民的被遗忘权, 保护公民隐私; 而且在认证过程中嵌入监管因子, 以供日后追责。最后, 安全性分析证明了所提方案满足不可链接性、可追溯性; 性能分析表明了所提方案具有较低计算成本和 Gas 开销, 可满足实际应用需求。

关键词: 元宇宙; 数字公民; 身份认证; 不可链接性; 可监管; 被遗忘权; 区块链

中图分类号: TP309.2

文献标志码: A

doi: 10.11959/j.issn.2096-6652.202239

A metaverse-oriented digital citizen authentication scheme

LYU Qiuyun¹, CHENG Shaopeng¹, YANG Manzhi^{2,3}, CHEN Xiaoguang^{2,3}, WANG Zhen^{1,4}

1. School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, China

2. Macau University of Science and Technology, Macao 999078, China

3. Eversec (Beijing) Technology Co., Ltd., Beijing 100086, China

4. ZhuoYue Honors College, Hangzhou Dianzi University, Hangzhou 310018, China

Abstract: As a surreal ternary world, the metaverse provides people with massive and diverse network services. However, due to the lack of face-to-face communication in reality and the broken geographical and time constraints, the existing schemes cannot effectively authenticate digital citizens while realizing privacy protection and accountability. Therefore, a metaverse-oriented digital citizen authentication scheme was proposed. Firstly, a layered authentication scheme based on the time-sensitive identity was constructed. Secondly, for protecting citizens' privacy, the Paillier homomorphic encryption algorithm and structure-preserving signature scheme were introduced to realize the double unlink ability of decentralized identifiers and verifiable credentials, in addition, the citizens' right to be forgotten was guaranteed by the code embedding technology. Thirdly, the regulatory factor during the authentication phase for accountability was designed. Finally, the security analysis proves that the proposed scheme satisfies the unlinkability and traceability. And, the performance analysis shows that the proposed scheme has lower computational cost and gas overhead, which can meet the needs of practical use.

Key words: metaverse, digital citizen, authentication, unlinkability, accountability, right to be forgotten, blockchain

0 引言

近年来, 被视为下一代互联网演变范式的元宇

宙^[1-2]吸引了各界人士的广泛关注, 学者们认为元宇宙的兴起昭示着智能产业的革命即将到来^[3]。在虚实融生的元宇宙中, 众多现实世界的自然人将通过数字

收稿日期: 2022-06-15; 修回日期: 2022-07-29

通信作者: 王震, wangzhen@hdu.edu.cn

基金项目: 浙江省重点研发计划基金资助项目 (No.2022C01018)

Foundation Item: The Key Research and Development Program of Zhejiang Province (No.2022C01018)

化身和海量数字人（分身以及假身等）共同存在于全球共享的虚拟空间中^[4]，进行虚拟空间位置的移动、各种服务访问以及与他人交互。鉴于自然人的社会属性，本文拟用公民来表示现实世界自然人本体，用数字公民（digital citizen, DC）表示自然人本体对应的数字化身。

虚实融生社会中公民与数字公民的关系如图1所示，在元宇宙中，作为公民的唯一映射，数字公民控制着分身和假身与他人交互以及访问海量的网络服务，例如网上购物、休闲娱乐、线上学习等，而在保证数据传输安全的过程中，身份认证是重中之重^[5]。然而，存在单点故障、隐私泄露等问题的传统身份管理方案^[6-7]无法满足元宇宙中的海量业务场景交互，且存在各种黑客攻击下可信身份认证的需求。因此，相比之下，允许用户控制自己身份的自我主权身份（self-sovereign identity, SSI）^[8]管理方案成为学者们公认的未来元宇宙中管理身份的主流选择^[1]。对于自我主权身份，万维网联盟（world wide web consortium, W3C）提出新型去中心化标识符（decentralized identifier, DID）^[9]和可验证凭证（verifiable credential, VC）^[10]，并将它们作为构建 SSI 的基础，诸多学者也基于此提出自我主权身份认证方案。

2021 年，Bandara E 等人^[11]提出通过点对点的方式将用户身份数据发送给验证者，并将其与物理证件（如身份证等）进行比对，以避免身份被冒充，

而被验证的用户 DID 与验证结果则被一同记录在区块链中，以防止篡改；但是，身份数据以明文传输给用户隐私保护带来了风险。为了避免用户个人隐私信息泄露，Zhou T 等人^[12]提出一种基于以太坊智能合约的可认证自我主权身份框架，其中引入一种密钥派生算法，使用不同的密钥对用户不同的属性值（如姓名、性别、年龄等）分别进行加密存储，以保护其隐私；此外，该框架将公钥视作每个用户的 DID，借助 solidity 内置函数，使用用户发送的数据签名还原出相应的公钥地址，继而通过智能合约将该公钥与区块链中记录的公钥进行对比，从而实现对其身份的认证。但与参考文献[11]相同，用户的 DID 同样被记录在区块链中，由于 DID 的不变性，每个用户的不同行为可以被攻击者链接在一起，从而导致用户的行为隐私暴露。

2020 年，Belchior R 等人^[13]基于访问控制提出一种自我主权身份管理方案，其中允许用户针对不同应用场景使用不同 DID 以实现身份的不可链接性，并且基于 W3C 的 VC 模型实现对用户身份的分布式认证。与参考文献[13]相似，Samir E 等人^[14]提出一种去中心化可信自我主权身份管理框架，使用 VC 模型实现对用户身份的认证，并采用秘密共享方案将 VC 分散存储在不同位置，以避免单点故障。同年，Stokkink Q 等人^[15]提出一种新型自我主权身份方案，使用公钥来表示 DID，而用于表示用户身份的 VC 则被依次链接在该公钥上，从而共同

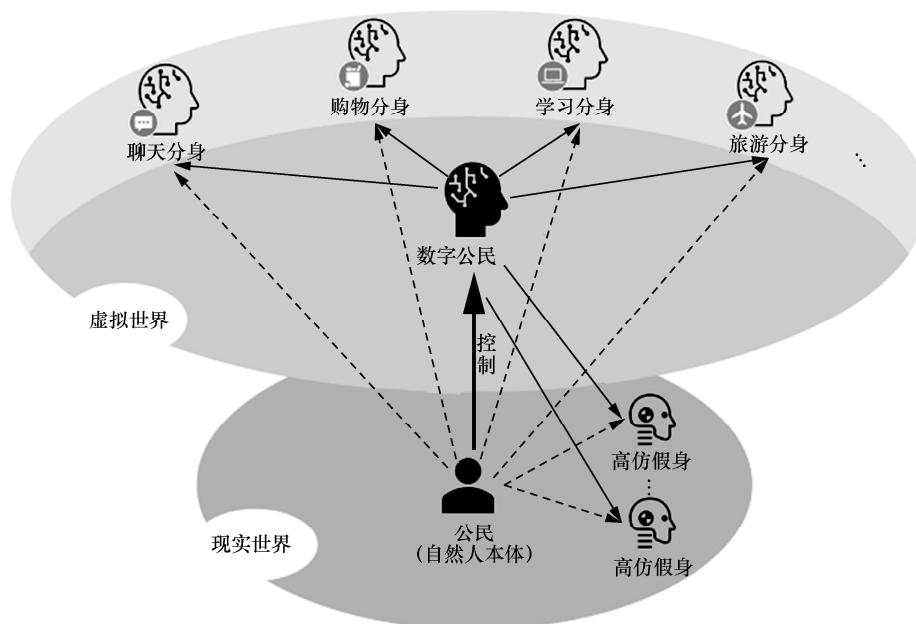


图1 虚实融生社会中公民与数字公民的关系

构成一条身份链，以此提高身份真实性；此外，Stokkink Q 等人^[15]引入单一监管机构来制裁恶意用户，但存在监管不足或监管不公正的情况。总体来讲，虽然在每次访问服务时改变用户 DID 可以避免用户身份被链接，却忽略了 VC 的不变性，使得用户在使用其认证身份时仍然存在身份被链接的可能，而对于虚实融生的元宇宙来说，数字世界身份的链接往往也意味着物理世界身份隐私的暴露。

而且，虽然现有的 SSI 方案致力于让用户控制自己的身份和数据，但他们却忽视了对用户被遗忘权^[16]的保护，不能保证用户在身份认证过程中发送的数据（例如 VC）在认证过程结束后被销毁，反而可能使数据存储在服务提供商（service provider, SP）手中，大大损害了元宇宙中用户的隐私和自主权。

综上所述，本文提出一种面向元宇宙的数字公民身份认证方案，在保证数字公民身份真实可靠的同时，实现身份的不可链接性和公民的被遗忘权，以保护公民在元宇宙中的个人隐私，并解决监管不公的问题，维护元宇宙的秩序稳定。值得一提的是，通过对元宇宙海量多样化的网络服务分析可知，大多数网络服务存在服务层级不同的两类身份：访客身份和会员身份。以网上购物为例，使用访客身份的数字公民，其行为只是简单的购买商品，只需证明其身份真实；而具备会员身份的数字公民可以享受更优惠的价格以及更优质的服务，所需要的不仅是对身份真实性的认证，还包括对其会员资格的认证。因此，为了使身份认证更加高效，本文基于 W3C 的 DID 标准以及 VC 模型提出分层认证方案，将 DID 用作数字公民访问网络服务的身份标识符，将 VC 用作数字公民的各类资格证明（如会员资格），通过 DID 与 VC 的灵活结合，分别实现对访客身份和会员身份的高效认证。此外，本文还引入椭圆曲线加密^[17]、Paillier 同态加密^[18]以及基于等价类的保持结构性签名^[19]等密码学组件，以实现 DID 与 VC 的双重不可链接，同时平衡隐私保护和公正监管；最后，通过在 VC 中嵌入具备自毁函数的可执行代码来保障公民的被遗忘权。

1 前置技术与理论

1.1 DID

本文方案将 DID 作为数字公民访问元宇宙网络服务的身份标识符。DID 是由 W3C 提出的一种新型全局唯一标识符，旨在使个人和组织控制自己的数

字身份。作为 SSI 模型的关键组成部分^[20]，DID 被设计为可以将 DID 主体与 DID 文档相关联的统一资源标识符（uniform resource identifier, URI），并且个人和组织可以使用分布式基础架构（如分布式账本技术（distributed ledger technology, DLT））来管理属于自己的 DID。此外，由于个人和组织等实体控制着 DID 的生成，每个实体都可以拥有尽可能多的 DID，以实现其身份的不可链接。本文则借助此特性来保护用户的行为隐私。DID 由 3 个部分构成：
<Scheme> :< Method> :< Method Specific Identifier>。

第一部分指明采用的是 DID 方案，第二部分则指定使用的 DID 方法，而第三部分表示按照 DID 方法的特定规则生成的标识符。对于第三部分，不同于常用的以公钥、智能合约地址等为标识符，本文设计了一种由用户生成的可信监管因子，并将其作为标识符，为 DID 增加一种新的可审计属性。

1.2 VC

本文引入 VC 表示数字公民在元宇宙中的各类资格证明。VC 是一种可互操作的数据结构，适用于表示加密可验证和防篡改的声明，与 DID 相同，VC 也是 SSI 模型的关键组成部分^[20]。VC 是物理世界中的凭证（如毕业证、驾照、护照等）在网络世界中的映射表现，其中数字签名等技术的引入使得 VC 比物理凭证具有更强的防篡改性。标准的 VC 由凭证元数据（credential metadata）、声明（claim）以及证明（proof）3 个部分组成，如图 2 所示。

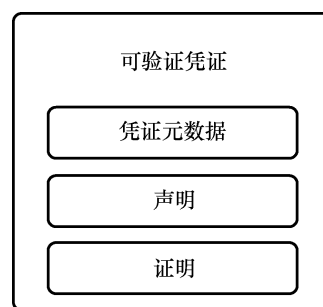


图2 可验证凭证结构

图2中的凭证元数据由多条描述凭证属性的数据组成，例如发行方、截止日期、公钥等；声明是 VC 主体对自身所具备属性的陈述，例如年龄大于 18 岁；而证明则是发行方对凭证元数据的数字签名。本文在标准 VC 的基础上进行改进，在数字化凭证的基础上构建新型 VC，该 VC 包含以下 3 个部分：
 $VC = \{credential, claims, EC\}$ 。其中凭证（credential）

是以数字化 JS 对象简谱 (JavaScript object notation, JSON) 结构表示的物理凭证, 声明 (claims) 是由用户构建的用于表示该 VC 所有者的属性构成, 可嵌入代码 (embeddable code, EC) 是一段可自动执行的代码, 位于 VC 最前端, 代码里可预设时间自毁函数, 当超过给定时间, 接收者无法打开 VC, 由此保障用户的被遗忘权。

1.3 Paillier 同态加密系统

本文使用 Paillier 加法同态特性设计认证因子。Paillier 同态加密系统^[18]是 Pascal Paillier 于 1999 年发明的概率公钥加密系统, 该系统基于复合剩余类的困难问题, 满足加法同态的性质。

1.3.1 Paillier 加密系统

Paillier 加密系统由以下 3 个算法构成。

(1) 密钥生成 KeyGen_p : 首先随机选择两个大素数 p 和 q , 使其满足 $\gcd(pq, (p-1) \times (q-1)) = 1$, 计算 $n = p \times q$ 和 $\lambda = \text{lcm}(p-1, q-1)$; 其次任选 $g \in Z_{n^2}^*$ 且使其满足 $\gcd(L(g^\lambda \bmod n^2), n) = 1$, 其中 $L(x) = \frac{x-1}{n}$, $Z_{n^2}^* \in \{1, 2, 3, \dots, n^2-1\}$; 最后选择公钥 $\text{PK} = (n, g)$ 和私钥 $\text{SK} = (\lambda)$, 返回 (PK, SK) 。

(2) 加密阶段 $\text{En}_p(m, r, \text{PK})$: 给定明文 $m \in Z_n^*$, 任选 $r \in Z_n^*$, 计算 $c = g^m \times r^n \bmod n^2$, 返回密文 c 。

(3) 解密阶段 $\text{De}_p(c, \text{SK})$: 给定密文 c , 计算 $m = \frac{L(c^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \bmod n^2$, 返回明文 m 。

1.3.2 Paillier 的加法同态特性

Paillier 加密系统满足加法同态特性, 即对两个密文的乘积进行解密后得到的结果等同于它们对应的明文之和, 证明如下。

对于 $\forall m_1, m_2 \in Z_n^*$, 任选 $r_1, r_2 \in Z_n^*$, 计算相应的密文 $c_1 = \text{En}_p(m_1, r_1, \text{PK})$ 和 $c_2 = \text{En}_p(m_2, r_2, \text{PK})$, 而 c_1 和 c_2 满足式 (1) 和式 (2)。

$$c_1 \times c_2 = g^{m_1+m_2} \times (r_1 \times r_2)^n \bmod n^2 \quad (1)$$

$$\text{De}_p(c_1 \times c_2, \text{SK}) = m_1 + m_2 \bmod n^2 \quad (2)$$

Paillier 同态加密系统具有语义安全性, 即当 $m_1, m_2 \in Z_n^*$, 任选 $r_1, r_2 \in Z_n^*$, 得到的 $c_1 = \text{En}_p(m_1, r_1, \text{PK})$ 和 $c_2 = \text{En}_p(m_2, r_2, \text{PK})$ 满足 $c_1 \neq c_2$ 。

1.4 基于等价类的保持结构性签名

本文基于等价类的保持结构性签名 (structure-preserving signatures on equivalence

classes, SPS) 方案构建 VC, 在保证签名真实可靠的同时, 实现 VC 的不可链接。Hanser C 等人^[19]于 2014 年提出一种基于等价类的保持结构性签名方案, 允许对组元素向量进行签名, 并且可以在不知道密钥的情况下随机化签名和消息。具体来说, 消息被看作组元素向量上的等价类的表示, 将原消息的每个分量与相同的标量相乘, 即可得到与原消息表示不同但等价的新消息, 继而通过相同的标量更新原签名, 得到与新消息对应的新签名。基于双线性配对的双线性特征 (如式 (3) 所示), 虽然新消息与新签名在表示上与原来不同, 但该签名仍是一个有效签名, 由此消息与数字签名实现表示上的变换, 避免数字签名因其不变性而被链接, 导致隐私泄露。

$$e(aP, bP') = e(P, P')^{ab} = e(bP, aP'), \forall a, b \in Z_p \quad (3)$$

该方案由以下 5 个算法构成。

(1) 系统建立 $\text{BGGen}_{\text{SPS}}(\chi)$: 给定安全参数 χ , 返回双线性群 BG。

(2) 密钥生成 $\text{KeyGen}_{\text{SPS}}(\text{BG}, \ell)$: 给定双线性群 BG 以及向量长度 ℓ , 随机选择 $x \in Z_p^*$ 和 $(x_i)_{i=1}^\ell \in (Z_p^*)^\ell$, 并将其作为私钥 $\text{SK} = (x, (x_i)_{i=1}^\ell)$, 计算相应的公钥 $\text{PK} = (X', (X'_i)_{i=1}^\ell) = (xP', (x_i x P')_{i=1}^\ell)$, 返回 (PK, SK) 。

(3) 签名阶段 $\text{Sign}_{\text{SPS}}(M, \text{SK})$: 输入等价类 $[M]$ 的一种表示 $M = (M_i)_{i=1}^\ell \in (G_1^*)^\ell$ 以及私钥 $\text{SK} = (x, (x_i)_{i=1}^\ell)$, 随机选择 $y \in Z_p^*$, 进行如下计算。

$$Z_1 = x \sum_{i=1}^\ell x_i M_i \quad (4)$$

$$Z_2 = y \sum_{i=1}^\ell x_i M_i \quad (5)$$

$$(Y, Y') = y \cdot (P, P') \quad (6)$$

返回签名 $\sigma = (Z_1, Z_2, Y, Y')$ 。

(4) 签名变换 $\text{ChgRep}_{\text{SPS}}(M, \sigma, \rho, \text{PK})$: 给定等价类 $[M]$ 的一种表示 $M = (M_i)_{i=1}^\ell \in (G_1^*)^\ell$, 相应的签名 $\sigma = (Z_1, Z_2, Y, Y')$, 标量 $\rho \in Z_p^*$ 以及公钥 $\text{PK} = (X', (X'_i)_{i=1}^\ell)$, 计算等价类 $[M]$ 的新表示 $\hat{M} = \rho \cdot (M_i)_{i=1}^\ell$, 随机选择 $\hat{y} \in Z_p^*$, 计算 $\hat{M}$ 对应的新签名 $\hat{\sigma} = (\rho Z_1, \hat{y} \rho Z_2, \hat{y} Y, \hat{y} Y')$, 返回 $(\hat{M}, \hat{\sigma})$ 。

(5) 验证阶段 $\text{Verify}_{\text{SPS}}(M, \sigma, \text{PK})$: 给定等价类 $[M]$ 的一种表示 $M = (M_i)_{i=1}^{\ell} \in (G_1^*)^{\ell}$, 相应的签名 $\sigma = (Z_1, Z_2, Y, Y')$ 以及公钥 $\text{PK} = (X', (X'_i)_{i=1}^{\ell})$, 依次判断下列等式是否成立。

$$\prod_{i=1}^{\ell} e(M_i, X'_i) = e(Z_1, P') \quad (7)$$

$$e(Z_1, Y') = e(Z_2, X') \quad (8)$$

$$e(P, Y') = e(Y, P') \quad (9)$$

若上述等式均成立, 则返回 1, 否则返回 0。

综上, SPS 方案具有完备性和可靠性, 具体定义如下。

(1) 完备性: 已知消息 M 和相应签名 σ , 任选 $\rho \in Z_p^*$, 得到的 $(\hat{M}, \hat{\sigma}) = \text{ChgRep}_{\text{SPS}}(M, \sigma, \rho, \text{PK})$ 可以通过验证。

(2) 可靠性: 只知签名 σ , 任选 M' 和 $\rho \in Z_p^*$, 得到的 $(\hat{M}', \hat{\sigma}) = \text{ChgRep}_{\text{SPS}}(M', \sigma, \rho, \text{PK})$ 无法通过验证。

2 前期工作与说明

在先前的工作中, 笔者提出了一种新式可问责

的自我主权身份方案 (new self-sovereign identity scheme with accountability, NSSIA)^[21], 以实现用户独立自主地体验元宇宙, 并在不侵犯合法用户隐私的同时, 实现对恶意用户公开、公平、公正的监管。在该方案中, 数字公民被设计为一段为用户量身打造的可执行程序, 使用这段仅位于本地的程序, 用户可以摆脱对第三方的依赖, 实现对身份存储、读取和验证的完全自控; 此外, 用户的生物特征也被嵌入该程序中, 以增强数字公民的唯一性和主权。为了避免监管部门权力的集中化, 同时保护用户隐私, 笔者设计了一种联合监管机制, 通过将 shamir(t, n) 门限签名算法与区块链技术结合来实现隐私和监管的平衡。

2.1 NSSIA

在 NSSIA 中, 笔者详细阐述了如何为每个自然人生成一个独一无二的数字公民以及在保护合法公民身份隐私的同时对恶意数字公民采取公正监管。具体来说, NSSIA 包含以下 4 个阶段, 分别是数字化、注册、认证和监管, 而受限于篇幅, NSSIA 只对数字化、注册以及监管 3 个阶段进行细节设计, 认证阶段的详细流程则是本文工作的重点。NSSIA 的系统模型如图 3 所示。

由图 3 可知, 公民首先通过信息采集与验证部

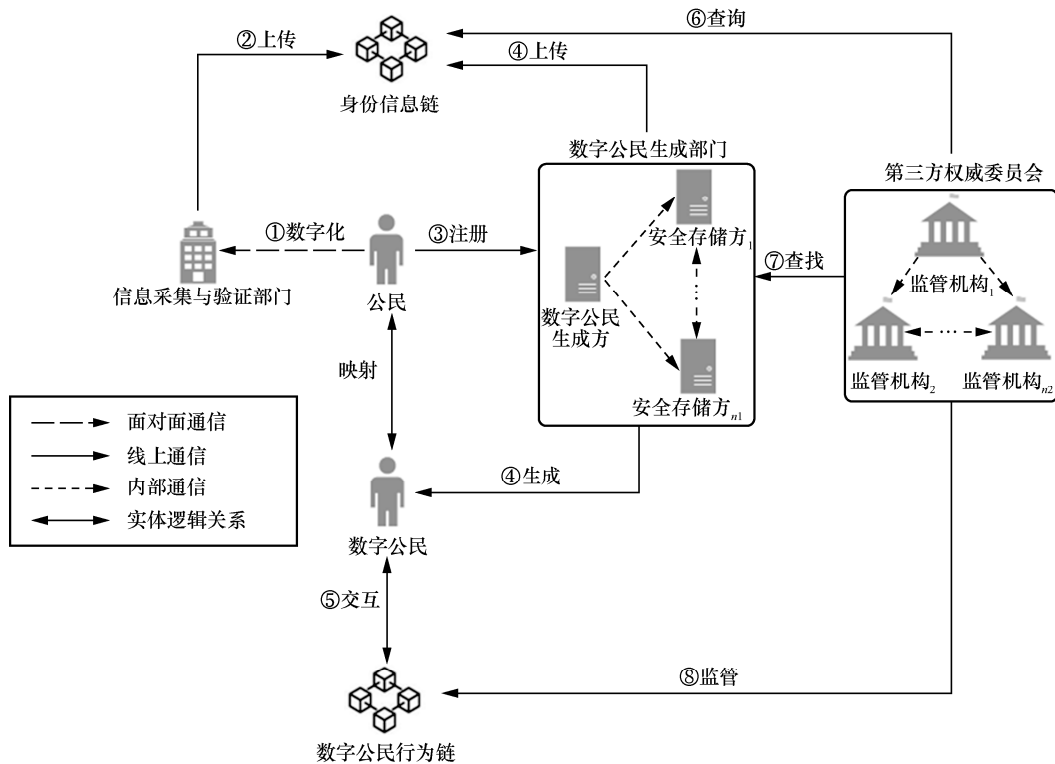


图 3 NSSIA 的系统模型

门完成基本的身份认证，对元数据（如姓名、住址）和生物特征（如虹膜、人脸）进行数字化，并接收由该部门颁发的身份认证证书；该部门则将公民相应的身份证明数据（元数据哈希、生物特征哈希）记录到身份信息链中留作存证。其次，通过身份认证证书，公民可以向数字公民生成部门注册数字公民；在验证公民活体生物特征准确无误后，数字公民生成方依据该生物特征从代码模块库中挑选不同功能模块（如文件传输、动态认证、转账交易等）来组建一段独一无二的可执行程序，该程序便是独属于该公民的数字公民；数字公民生成方将生成的数字公民发送给相应的公民并将证明（如数字公民哈希）记录到身份信息链中，此外，为了方便日后问责，同时保护用户隐私，公民的个人信息和数字公民哈希基于 $\text{shamir}(t, n)$ 门限签名算法被分割成多个毫无关联的片段，并分别存储在不同的安全存储方中。最后，通过数字公民，公民可以在元宇宙中与其他公民进行交互，并访问数字公民行为链（digital citizen behavior chain, DCBC）上的各类服务，而数字公民的所有行为则被完整记录在该链中；一旦某个监管机构在监控数字公民行为链时发现可疑行为，便可以联合其他多个监管机构发起对该可疑数字公民的正式监管程序，通过在身份信息链中查找相应的身份证明并还原存储在安全存储方中的公民身份信息，确定其在物理世界中的真实身份，并按照策略对其采取相应的制裁措施。

2.2 补充说明

本文在 NSSIA 工作的基础上完善数字公民在身份认证阶段的具体流程，由于对恶意用户的监管包含对数字公民认证阶段行为的分析，因此，本文在设计数字公民身份认证方案的同时将身份可监管考虑在内，并结合 NSSIA 中的联合监管机制实现对恶意用户的公正监管。出于对数字公民身份认证和监管的合理性以及保持 NSSIA 与本文方案整体性的考虑，笔者对 NSSIA 中的注册流程进行如下补充：数字公民生成方在生成数字公民的同时，构建可用于证明其身份真实性的主凭证。该凭证的作用可类比物理世界中由政府机关发行的居民身份证，既可以在身份认证阶段证明公民身份的真实性，又可以辅助监管机构确定恶意攻击者的具体身份信息。主凭证的构建细节以及本文采用的符号术语将在下文详细说明。

2.2.1 符号说明

为了便于阐释方案的构建细节，本节给出本文使用的各类术语，见表 1。其中，变量主要由首字母缩写表示，如数字公民哈希值（digital citizen information, DCI），实体由右下标表示；算法术语依照其功能进行设计，其中 alg 表示密码学算法；对于复杂术语，如公私钥对，分别使用右上标和右下标区分实体和密码学算法。

表 1 本文术语

术语	描述
DCI	数字公民哈希值
SPI	服务提供商公开信息
Cre	权威机构发行的原始凭证
VC	Cre 对应的可验证凭证
DID_e	实体 e 的身份标识符 DID
AF_e	实体 e 的认证因子
r_e	认证因子 AF_e 对应的校验码
sk	数字公民生成方与第三方权威委员会共享的私钥
$h()$	哈希函数
$(\text{PK}_{\text{alg}}^e, \text{SK}_{\text{alg}}^e)$	实体 e 拥有的 alg 方案公私钥
$\text{Setup}_{\text{ecc}}$	椭圆曲线加密方案的系统建立算法
$\text{BGGen}_{\text{SPS}}$	SPS 方案的系统建立算法
$\text{KeyGen}_{\text{alg}}$	alg 方案的密钥生成算法
En_{alg}	alg 方案的加密算法
De_{alg}	alg 方案的解密算法
Sign_{SPS}	SPS 方案的签名生成算法
$\text{ChgRep}_{\text{SPS}}$	SPS 方案的签名变换算法
$\text{Verify}_{\text{SPS}}$	SPS 方案的签名验证算法

2.2.2 主凭证设计

主凭证作为数字公民最基本、最重要的证明数据，主要由发行者(issuer)、类型(type)、声明(claims) 3 个部分组成，其中发行者是主凭证颁发者（即数字公民生成方）的 DID 标识符，通过该 DID，验证者可以查找其相应的公钥，从而对主凭证的真伪进行验证；类型表示该凭证所属类别，对于主凭证，该字段为“master”；声明则由多个属性-值对构成，主要包括认证因子（authentication factor, AF）、辅助因子等属性。主凭证生成流程如下。

步骤一：计算 $AF_{DC} = \text{En}_p(\text{DCI} \oplus h(\text{sk}), r_{DC}, \text{PK}_{\text{paillier}}^{\text{TAC}})$ 。其中 AF_{DC} 是数字公民的认证因子， $r_{DC} \in Z_n^*$ 是与 AF_{DC} 相对应的辅助因子， $\text{PK}_{\text{paillier}}^{\text{TAC}}$ 是第三方权威委员会的 Paillier 同态加密密钥。

步骤二：构建 $\text{Cre} = (\text{DID}_{\text{issuer}}, \text{master}, AF_{DC}, r)$ 。其中 Cre 表示主凭证， $\text{DID}_{\text{issuer}}$ 是发行者的 DID， $r \in Z_n^*$ 是随机数。

步骤三：将 Cre 转变为全局常量语句，并随机嵌入数字公民中的不同位置，在数字公民成功生成后，一同发送给相应的公民。

3 系统模型与安全模型

3.1 系统模型

本文方案的系统模型如图4所示，主要包括以下7个实体。

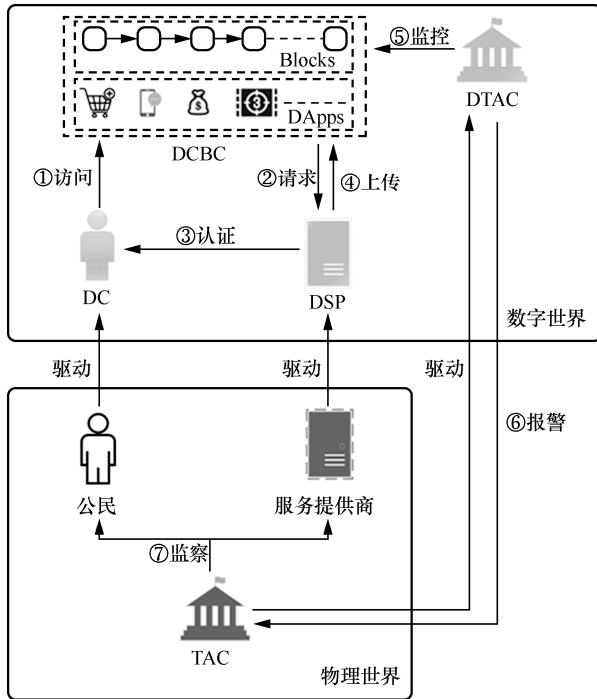


图4 本文方案的系统模型

(1) DCBC: DCBC 属于公有链，其上运行着多个提供各种服务（如网络购物、在线社交、理财等）的分布式应用（decentralized application, DApp），这些 DApp 本质上由代码构成，例如以太坊的智能合约。在元宇宙中，DC 可以直接访问 DApp 来请求所需服务，而 DApp 则会自动地将请求转给相应的 SP，并将 DC 的所有访问行为记录到 DCBC 中以备日后审计。

(2) 公民：公民是生活在物理世界中的自然人，可以驱动自己的 DC 来访问元宇宙中的各种服务。

(3) DC: DC 是公民在数字世界中的映射，由公民驱动访问 DCBC 上的各种服务；此外，DC 还可以按照公民提前设定的程序自主地在元宇宙中与他人交互。

(4) SP: SP 位于物理世界中，负责提供服务（如网络购物、在线社交、理财等），每个 SP 通过驱动 DSP 来管理运行在 DCBC 上的 DApp。

(5) 数字服务商 (digital service provider, DSP): DSP 是 SP 在数字世界中的映射，由 SP 驱动，负责处理 DC 发出的服务请求以及认证身份，并将认证结果记录在 DCBC 上。

(6) 第三方权威委员会 (third-party authoritative committee, TAC): TAC 由物理世界多个相互制约的机构（如公安局、教育局、工商局等）共同组成，负责监管系统内的其他实体，例如公民、SP 等。

(7) 数字委员会 (digital third-party authoritative committee, DTAC): DTAC 是 TAC 在数字世界中的映射，由 TAC 驱动，负责实时监控 DCBC 上记录的 DC 与 SP 之间的交互行为，并将其中的可疑行为报告给 TAC。

为了在元宇宙中如数字原生人一般生活，公民需要驱动自己的 DC 来访问 DCBC 上提供的海量多样化应用，如步骤①所示；为了避免身份欺诈，SP 则需通过驱动相应的 DSP 在数字世界中处理 DC 的访问服务请求，并验证其身份，而验证结果则由 DSP 上传到 DCBC 中，如步骤②~④所示；出于维护元宇宙秩序的需要，TAC 通过驱动 DTAC 对 DCBC 中所有的交互记录进行实时监控，一旦发现可疑行为，DTAC 立即报警，而 TAC 则依照策略对相应的实体（公民和/或 SP）实施监察，如步骤⑤~⑦所示。

3.2 形式化定义

本文方案包括以下算法。

(1) 系统建立 $\text{Setup}(\chi)$: 输入安全参数 χ ，返回系统公共参数 P 。

(2) 密钥生成 $\text{GenKey}(P)$: 输入系统公共参数 P ，DC、SP 以及 TAC 分别生成各自的密钥对。

(3) 临时身份生成 $\text{GenId}(AF_{DC}, \text{SPI}, r_{DC}, \text{PK}_{\text{ecc}}^{\text{SP}})$: DC 使用自己的认证因子 AF_{DC} 以及待访问 SP 的公开信息 SPI 生成临时标识符 DID_{DC} ，并使用辅助因子 r_{DC} 生成对应的身份校验码。

(4) 临时身份认证 $\text{VerId}(\text{DID}_{DC}, C, AF_{SP},$

$SPI, r_{SP}, SK_{ecc}^{SP}$): SP 使用 DC 的身份校验码 C 和自己的认证因子 AF_{SP} 、公开信息 SPI 以及相应的辅助因子 r_{SP} 验证 DID_{DC} , 若验证通过, 则返回 1, 否则返回 0。

(5) 凭证请求 $ReqCre(DID_{DC}, attr, t, PK_{ecc}^{SP})$: DC 使用 SP 的公钥 PK_{ecc}^{SP} 对需要验证的属性 $attr$ 和时间戳 t 进行加密, 并发送给 SP, 请求相应的可信凭证。

(6) 凭证生成 $GenCre(DID_{DC}, AC, SK_{ecc}^{SP}, SK_{SPS}^{SP})$: SP 使用私钥 SK_{ecc}^{SP} 和 SK_{SPS}^{SP} 验证密态属性值 AC 并为通过验证的 DC 颁发相应的凭证。

(7) 可验证凭证生成 $GenVC(Cre, claims)$: DC 使用凭证 Cre 以及声明集合 $claims$ 构造可验证凭证 VC 。

(8) 可验证凭证验证 $VerVC(VC)$: 输入 VC , SP 验证其真实性, 并返回 ω , 若 VC 真实可信, 则 $\omega=1$, 否则 $\omega=0$ 。

(9) 身份追溯 $TraceId(DID_{DC}, SK)$: 输入 DID_{DC} 以及 TAC 私钥 SK , 还原相应 DC 的身份。

3.3 安全模型

面向元宇宙的数字公民身份认证方案的安全性包括不可链接性和可追溯性, 下面通过挑战者 B 和攻击者 A 之间的博弈给出方案的安全模型。

3.3.1 不可链接性

定义 1: 不可链接性要求没有 TAC 的私钥, 攻击者在给予多个 DC 的 DID 和 VC 后不能将其归并到相应的公民身上。根据安全性模型, 即不存在任何多项式时间, 攻击者 A 在下面不可链接性模拟攻击博弈中以不可忽略的概率获得成功。

(1) 生成阶段: 挑战者 B 执行系统建立、密钥生成算法, 把生成的系统参数 P 、TAC 公钥发送给攻击者 A, 保留 TAC 私钥。

(2) 询问阶段: 攻击者 A 可以对挑战者 B 进行多项式有界次数的询问。

① DID 询问: 攻击者 A 将 SP 的公开信息 SPI 发送给挑战者 B, 挑战者 B 执行临时身份生成算法, 返回 DID 。

② VC 询问: 攻击者 A 指定挑战者 B 需提供的凭证类型, 挑战者 B 执行可验证凭证生成算法, 返回 VC 。

(3) 挑战阶段: 攻击者 A 指定凭证类型, 选择用户 DC_1 和 DC_2 以及 4 个不同的 SP 公开信息 $SPI_i (i=1,2,3,4)$, 并将其发送给挑战者 B, 挑战者

B 使用 SPI_1 和 SPI_2 执行临时身份生成算法以及可验证凭证生成算法, 为 DC_1 生成 $\{DID_{11}, VC_{11}\}$ 和 $\{DID_{12}, VC_{12}\}$, 使用 SPI_3 和 SPI_4 为 DC_2 生成 $\{DID_{21}, VC_{21}\}$ 和 $\{DID_{22}, VC_{22}\}$, 并将结果返回给攻击者 A。

(4) 猜测阶段: 攻击者 A 输出 DID 和 VC 的分组情况, 若输出为 $\{\{DID_{11}, VC_{11}\}, \{DID_{12}, VC_{12}\}\}$ 和 $\{\{DID_{21}, VC_{21}\}, \{DID_{22}, VC_{22}\}\}$, 则攻击者 A 攻击成功。

攻击者 A_1 攻击成功的概率为 $Adv_{A_1}^{unlink} = Pr[A_1 \text{ wins}]$ 。

3.3.2 可追溯性

定义 2: 可追溯性要求不诚信用户不能创建一个 TAC 无法追溯的有效临时 DID 。根据安全性模型, 即不存在任何多项式时间, 攻击者 A 在下面可追踪性模拟攻击博弈中以不可忽略的概率获得成功。

(1) 生成阶段: 挑战者 B 执行系统建立、密钥生成算法, 将生成的 TAC 公钥发送给攻击者 A, 保留 TAC 私钥。

(2) 询问阶段: 攻击者 A 可以对挑战者 B 进行多项式有界次数的询问。

① DID 询问: 攻击者 A 将 SP 的公开信息 SPI 发送给挑战者 B, 挑战者 B 执行临时身份生成算法, 返回 DID 。

② 验证询问: 攻击者 A 选择 DID 、密态校验码 C 请求询问, 挑战者 B 执行临时身份认证算法并返回结果给攻击者 A。

③ 追溯询问: 攻击者 A 选择一个 DID 请求询问, 挑战者 B 执行追溯算法并返回结果给攻击者 A。

(3) 挑战阶段: 攻击者 A 出示一个 DID , 若该 DID 通过临时身份认证算法, 但追溯算法还原出的身份信息不可读, 则攻击者 A 攻击成功。

攻击者 A_2 攻击成功的概率为 $Adv_{A_2}^{trace} = Pr[A_2 \text{ wins}]$ 。

4 具体方案

针对元宇宙网络服务存在的服务层级不同的两类身份: 访客身份和会员身份, 本文提出两种身份认证方案: 匿名可监管的访客身份认证 (anonymous guest authentication with accountability, AGAA) 方案和匿名可监管的会员身份认证

(anonymous membership authentication with accountability, AMAA) 方案。

4.1 AGAA 方案

AGAA 方案由 3 个部分组成, 首先是系统初始化, DC、DSP 以及 DTAC 分别生成各自的密钥对; 其次是访客身份认证, 在每次访问网络服务时, 公民驱动 DC 生成一个临时 DID 用作标识符, SP 则驱动 DSP 验证该 DID 的真实性, 以防止身份伪造; 最后, 当 TAC 收到对某可疑行为发出的报警时, TAC 使用其中包含的 DID 还原出相应的 DC 个人信息, 再结合 NSSIA 中的联合监管机制, 确定其相应的公民身份信息, 并依照策略对其实施监察。

4.1.1 系统初始化

(1) 系统建立 $\text{Setup}(\chi)$: 输入安全参数 χ , DCBC 执行以下操作。

① 执行 $\text{BGGen}_{\text{SPS}}(\chi)$ 算法, 生成一个双线性群 BG。

② 执行 $\text{Setup}_{\text{ecc}}$ 算法, 生成 ECC 加密公开参数 $\text{param} = \{p_0, E_p(a, b), G, n\}$ 。

③ 选择 $h: (0, 1)^* \rightarrow (0, 1)^n$, DCBC 将 $P = \{\text{BG}, \text{param}, h\}$ 写入创世块中, 并公开。

(2) 密钥生成 $\text{GenKey}(P)$ 。

① DC 执行 $\text{KeyGen}_{\text{ecc}}$ 算法, 生成 ECC 加密公私钥对 $(\text{PK}_{\text{ecc}}^{\text{DC}}, \text{SK}_{\text{ecc}}^{\text{DC}})$, DC 在本地保存 $\text{SK}_{\text{ecc}}^{\text{DC}}$, 公开 $\text{PK}_{\text{ecc}}^{\text{DC}}$ 。

② DSP 执行 $\text{KeyGen}_{\text{ecc}}$ 算法, 生成 ECC 加密公私钥对 $(\text{PK}_{\text{ecc}}^{\text{SP}}, \text{SK}_{\text{ecc}}^{\text{SP}})$, 然后执行 $\text{KeyGen}_{\text{SPS}}(\text{BG}, \zeta)$ 算法, 生成 SPS 公私钥对 $(\text{PK}_{\text{SPS}}^{\text{SP}}, \text{SK}_{\text{SPS}}^{\text{SP}})$, 将 $\{\text{PK}_{\text{ecc}}^{\text{SP}}, \text{PK}_{\text{SPS}}^{\text{SP}}\}$ 公开。

③ DTAC 执行 KeyGen_p 算法, 生成 Paillier 同态加密公私钥对 $(\text{PK}_{\text{paillier}}^{\text{TAC}}, \text{SK}_{\text{paillier}}^{\text{TAC}})$, TAC 将公钥 $\text{PK}_{\text{paillier}}^{\text{TAC}}$ 公开。

4.1.2 访客身份认证

访客身份认证由以下两个概率多项式时间算法组成, 具体如下。

(1) 临时身份生成 $\text{GenId}(\text{AF}_{\text{DC}}, \text{SPI}, r_{\text{DC}}, \text{PK}_{\text{ecc}}^{\text{SP}})$: 输入 DC 的认证因子 AF_{DC} 、待访问 SP 的公开信息 SPI、辅助因子 r_{DC} 以及待访问 SP 的公钥 $\text{PK}_{\text{ecc}}^{\text{SP}}$, DC 按以下步骤生成临时 DID_{DC}。

① 计算 $\text{AF}'_{\text{DC}} = \text{AF}_{\text{DC}} r_1^n \bmod(n^2)$, 其中 AF'_{DC} 是盲化后的 DC 认证因子, $r_1 \in Z_n^*$ 为随机数。

② 计算 $\text{DID}_{\text{DC}} = g^{\text{SPI}} \text{AF}'_{\text{DC}} \bmod(n^2)$ 。

③ 计算 $C = \text{En}_{\text{ecc}}(\text{En}_p(\text{DCI}, r_{\text{DC}} r_1, \text{PK}_{\text{paillier}}^{\text{TAC}}) \parallel \text{AF}'_{\text{DC}}, \text{PK}_{\text{ecc}}^{\text{SP}})$, 以 DID_{DC} 为身份标识符将密态校验码 C 发送给 SP, 并将此行为记录到 DCBC 中。

(2) 临时身份认证 $\text{VerId}(\text{DID}_{\text{DC}}, C, \text{AF}_{\text{SP}}, \text{SPI}, r_{\text{SP}}, \text{SK}_{\text{ecc}}^{\text{SP}})$: 输入 DC 的临时标识符 DID_{DC}、密态校验码 C、SP 的认证因子 AF_{SP} 和公开信息 SPI、辅助因子 r_{SP} 以及私钥 $\text{SK}_{\text{ecc}}^{\text{SP}}$, SP 按以下步骤验证 DID_{DC} 的真实性。

① 计算: $\{\text{ch} \parallel \text{AF}_{\text{DC}}''\} = \text{De}_{\text{ecc}}(C, \text{SK}_{\text{ecc}}^{\text{SP}})$, 其中 ch 为临时变量, 而 AF_{DC}'' 是将密态校验码 C 解密后得到的 DC 认证因子。

② 计算: $\text{ch}' = \text{En}_p(\text{SPI}, r_2, \text{PK}_{\text{paillier}}^{\text{TAC}})$, 其中 ch' 是临时变量, $r_2 \in Z_n^*$ 为随机数。

③ 验证等式 $\text{ch}' \text{AF}_{\text{DC}}'' = \text{ch} \text{AF}_{\text{SP}} = \text{DID}_{\text{DC}} r_{\text{SP}}^n \bmod(n^2)$ 是否成立, 若成立, 返回 1, 否则返回 0, SP 根据返回值将 DC 的验证结果记录到 DCBC 上。

4.1.3 监管

监管由一个概率多项式时间组成, 具体流程如下。

(1) TAC 分析链上数据, 获取可疑 DID_{DC}。

(2) 身份追溯 $\text{TraceId}(\text{DID}_{\text{DC}}, \text{SK}_{\text{paillier}}^{\text{TAC}})$ 。

① 计算 $c = \text{De}_p(\text{DID}_{\text{DC}}, \text{SK}_{\text{paillier}}^{\text{TAC}})$ 。

② 计算 $\text{DCI}' = c \oplus h(\text{sk})$, 其中 DCI' 是 TAC 还原得到的 DC 身份信息。

(3) 依据获得的 DCI', 通过 NSSIA 中的联合监管机制, 在身份信息链以及安全存储方中查找并还原可疑公民的真实身份, 依照策略对其实施监察。

4.2 AMAA 方案

AMAA 方案包括系统初始化、会员身份认证以及监管 3 个部分, 其中系统初始化和监管流程与第 4.1 节相同, 而会员身份认证不仅需要验证 DC 身份标识符的真实性, 还要求 DC 提供特定 SP 颁发的凭证。

4.2.1 系统初始化

系统初始化内容与第 4.1.1 节相同。

4.2.2 会员身份认证

以 SP₂ 要求 DC 提供 SP₁ 颁发的凭证为例, 会

员身份认证由不可链接的 DID 和可被遗忘的 VC 两部分共 6 个概率多项式时间算法组成，具体设计如下。

不可链接的 DID 由以下两个概率多项式时间算法组成，其具体流程与第 4.1.2 节一致。

(1) 临时身份生成 $\text{GenId}(\text{AF}_{\text{DC}}, \text{SPI}, r, \text{PK}_{\text{ecc}}^{\text{SP}})$ 。

(2) 临时身份认证 $\text{VerId}(\text{DID}_{\text{DC}}, C, \text{AF}_{\text{SP}}, \text{SPI}, r_{\text{SP}}'', \text{SK}_{\text{ecc}}^{\text{SP}})$ 。

可被遗忘的 VC 由以下 4 个概率多项式时间算法组成，具体设计如下。

(1) 凭证请求 $\text{ReqCre}(\text{DID}_{\text{DC}}, \text{attr}, t, \text{PK}_{\text{ecc}}^{\text{SP}_1})$ ：输入临时 DID_{DC} 、属性 attr 、时间戳 t 以及 SP_1 的公钥 $\text{PK}_{\text{ecc}}^{\text{SP}_1}$ ，DC 首先计算 $\text{AC} = \text{En}_{\text{ecc}}(\text{attr} \parallel t, \text{PK}_{\text{ecc}}^{\text{SP}_1})$ ，然后以 DID_{DC} 为身份标识符将密态属性值 AC 发送给 SP_1 ，并将此行为记录到 DCBC 中。

(2) 凭证生成 $\text{GenCre}(\text{DID}_{\text{DC}}, \text{AC}, \text{SK}_{\text{ecc}}^{\text{SP}_1}, \text{SK}_{\text{SPS}}^{\text{SP}_1})$ ：输入 DID_{DC} 、密态属性值 AC 以及 SP_1 的私钥 $\text{SK}_{\text{ecc}}^{\text{SP}_1}$ 和签名私钥 $\text{SK}_{\text{SPS}}^{\text{SP}_1}$ ， SP_1 按以下步骤为 DC 生成凭证。

① 计算 $\{\text{attr} \parallel t\} = \text{De}_{\text{ecc}}(\text{AC}, \text{SK}_{\text{ecc}}^{\text{SP}_1})$ ，若解密得到的时间戳 t 和属性 attr 无误，则进入下一步，否则返回 0，拒绝用户的请求。

② 计算 $\text{Sig} = \text{Sign}_{\text{SPS}}(\text{attr}, \text{SK}_{\text{SPS}}^{\text{SP}_1})$ ，其中 Sig 是 SP_1 使用私钥 $\text{SK}_{\text{SPS}}^{\text{SP}_1}$ 为 attr 生成的签名。

③ 构建 $\text{Cre} = \{\text{DID}_{\text{SP}_1}, \text{type}, \text{AC}, \text{Sig}\}$ ，将凭证 Cre 发送给 DC。DC 在接收到 Cre 后，计算 $\text{Verify}_{\text{SPS}}(\text{attr}, \text{Sig}, \text{PK}_{\text{SPS}}^{\text{SP}_1})$ 是否成立，若成立，返回 1，否则返回 0，拒绝接收此凭证 Cre。

(3) 可验证凭证生成 $\text{GenVC}(\text{Cre}, \text{claims})$ ：输入凭证 Cre 以及声明集合 claims，按以下步骤生成 VC。

① 执行可嵌入代码生成算法，如算法 1 所示，生成可嵌入代码 EC。

算法 1：可嵌入代码生成算法

输入：claims //声明集合

输出：EC //可嵌入代码

```

1   $V = \text{Query}(\text{claims}, "T_{\text{vaild}}")$  //查询有效期
2   $\text{EC} = \{$ 
3       $T = \text{getTime}()$  //获取当前时间
4      if  $T > V$  then
5           $\text{Destroy}()$  //自毁函数
```

6 **end if**

7 $\}$

8 **return EC;**

② 计算 $\{\text{attr}', \text{Sig}'\} = \text{ChgRep}_{\text{SPS}}(\text{attr}, \text{Sig}, \rho, \text{PK}_{\text{SPS}}^{\text{SP}_1})$ ，其中， attr' 和 Sig' 分别是经过变换后的属性值和签名。

③ 计算 $C' = \text{En}_{\text{ecc}}(\text{attr}', \text{PK}_{\text{ecc}}^{\text{SP}_2})$ ，其中 C' 是变换后属性值对应的密文，用 $\{C', \text{Sig}'\}$ 替换 Cre 中相应的字段。

④ 构建 $\text{VC} = \{\text{EC}, \text{Cre}, \text{claims}\}$ ，并将其发送给 SP_2 。

(4) 可验证凭证验证 $\text{VerVC}(\text{VC})$ ：输入 VC，按以下步骤验证其真伪。

① 首先自动运行 VC 中的代码段 EC，若正常运行，进入下一步；否则执行自毁操作，销毁 VC，终止算法。

② 获取 VC 中的 DID_{SP_1} ，并据此获取其签名公钥 $\text{PK}_{\text{SPS}}^{\text{SP}_1}$ 。

③ 获取 VC 中的密态属性值 C' ，计算 $\text{attr}'' = \text{De}_{\text{ecc}}(C', \text{SK}_{\text{ecc}}^{\text{SP}_2})$ 。

④ 计算 $\text{Verify}_{\text{SPS}}(\text{attr}'', \text{Sig}', \text{PK}_{\text{SPS}}^{\text{SP}_1})$ ，若验证通过，返回 1，否则返回 0， SP_2 根据返回值将验证结果记录到 DCBC 中。

4.2.3 监管

监管内容与第 4.1.3 节相同。

5 安全性分析

本文方案基于椭圆曲线加密方案、Paillier 同态加密系统以及 SPS 方案构造，根据各模块的安全特性，给出不可链接性和可追踪性的安全性分析。

5.1 不可链接性分析

本文方案的不可链接性满足以下结论。

结论 1：若椭圆曲线加密方案满足基本安全性质、Paillier 同态加密系统满足基本安全性质、SPS 方案满足完备性和可靠性，则本文方案满足不可链接性。

证明：定义 $\mathcal{A}_{\text{unlink}}$ 为攻击本文方案不可链接性模拟攻击博弈的敌手， $\mathcal{A}_{\text{ecc}}$ 为攻击椭圆曲线加密方案的敌手， $\mathcal{A}_p$ 为攻击同态加密系统的敌手， $\mathcal{A}_{\text{SPS}}$ 为攻击 SPS 方案的敌手。假设 $\mathcal{A}_{\text{unlink}}$ 成功攻击了该方案的不可链接性，定义一个多项式时间算法

$\mathcal{A}_\zeta \in (\mathcal{A}_{\text{ECC}}, \mathcal{A}_p, \mathcal{A}_{\text{SPS}})$ 包含攻击椭圆曲线加密方案、Paillier 同态加密系统、SPS 方案等攻击者的能力。在不可链接性模拟攻击博弈中 $\mathcal{A}_{\text{unlink}}$ 的查询和 $\mathcal{A}_\zeta$ 的响应交互来构造 $\mathcal{A}_\zeta$ ，使其能够攻击椭圆曲线加密方案、Paillier 同态加密系统和 SPS 方案，即若敌手 $\mathcal{A}_{\text{unlink}}$ 成功攻击本文方案的不可链接性，则 $\mathcal{A}_\zeta$ 就能以某一概率成功攻击椭圆曲线加密方案、Paillier 同态加密系统和 SPS 方案。

根据不可链接性定义，算法 $\mathcal{A}_\zeta$ 和敌手 $\mathcal{A}_{\text{unlink}}$ 的交互运行如下。

(1) 生成阶段：算法 $\mathcal{A}_\zeta$ 执行系统建立、密钥生成算法，通过分别运行 $\mathcal{A}_{\text{ECC}}$ 、 $\mathcal{A}_p$ 、 $\mathcal{A}_{\text{SPS}}$ 在椭圆曲线加密方案、Paillier 同态加密系统和 SPS 方案模拟攻击博弈中产生的公共参数，生成不可链接性模拟攻击博弈的公开参数 P 和 TAC 的公钥 $\text{PK}_{\text{paillier}}^{\text{TAC}}$ ，并将其发送给 $\mathcal{A}_{\text{unlink}}$ ，保留 TAC 私钥 $\text{SK}_{\text{paillier}}^{\text{TAC}}$ 。

(2) 询问阶段：敌手 $\mathcal{A}_{\text{unlink}}$ 可以对算法 $\mathcal{A}_\zeta$ 进行多项式有界次数的询问。

①DID 询问：敌手 $\mathcal{A}_{\text{unlink}}$ 指定 SP 的公开信息 SPI，请求询问用户 DC 对应的临时身份 DID，算法 $\mathcal{A}_\zeta$ 通过运行 $\mathcal{A}_{\text{ECC}}$ 和 $\mathcal{A}_p$ 在模拟攻击博弈中的 DID 询问，将得到的 DID 询问结果返回给 $\mathcal{A}_{\text{unlink}}$ 。

②VC 询问：敌手 $\mathcal{A}_{\text{unlink}}$ 指定提供的凭证类型，请求询问对应的 VC，算法 $\mathcal{A}_\zeta$ 通过运行 $\mathcal{A}_{\text{ECC}}$ 和 $\mathcal{A}_{\text{SPS}}$ 在模拟攻击博弈中的 VC 询问，将得到的 VC 询问结果返回给 $\mathcal{A}_{\text{unlink}}$ 。

(3) 挑战阶段：敌手 $\mathcal{A}_{\text{unlink}}$ 指定凭证类型，并选取用户 DC_1 和 DC_2 以及 4 个不同的 SP 的公开信息 $\text{SPI}_i (i=1,2,3,4)$ ，算法 $\mathcal{A}_\zeta$ 依据 $\{\text{SPI}_1, \text{SPI}_2\}$ 和 $\{\text{SPI}_3, \text{SPI}_4\}$ 分别为 DC_1 和 DC_2 运行 $\mathcal{A}_{\text{ECC}}$ 、 $\mathcal{A}_p$ 和 $\mathcal{A}_{\text{SPS}}$ 在模拟攻击博弈中的挑战阶段，得到 $\{\text{DID}_{11}, \text{VC}_{11}\}$ 、 $\{\text{DID}_{12}, \text{VC}_{12}\}$ 、 $\{\text{DID}_{21}, \text{VC}_{21}\}$ 、 $\{\text{DID}_{22}, \text{VC}_{22}\}$ ，并返回给敌手 $\mathcal{A}_{\text{unlink}}$ 。

(4) 猜测阶段：如果敌手 $\mathcal{A}_{\text{unlink}}$ 将 DID 和 VC 分组为 $\{\{\text{DID}_{11}, \text{VC}_{11}\}, \{\text{DID}_{12}, \text{VC}_{12}\}\}$ 和 $\{\{\text{DID}_{21}, \text{VC}_{21}\}, \{\text{DID}_{22}, \text{VC}_{22}\}\}$ ，说明敌手 $\mathcal{A}_{\text{unlink}}$ 成功进行了攻击。

敌手 $\mathcal{A}_{\text{unlink}}$ 攻击成功的概率为：

$$\begin{aligned} \text{Adv}_{\mathcal{A}_{\text{unlink}}}(k) &= \Pr[\text{Exp}_{\mathcal{A}_{\text{unlink}}}(k) = 1] = \\ &\Pr[\mathcal{A}_{\text{unlink}}(\text{guess} = \{\{\text{DID}_{11}, \text{VC}_{11}\}, \{\text{DID}_{12}, \text{VC}_{12}\}\})] = \end{aligned}$$

$$\begin{aligned} &\Pr[\{\{\text{DID}_{21}, \text{VC}_{21}\}, \{\text{DID}_{22}, \text{VC}_{22}\}\}] = \\ &\Pr[\mathcal{A}_{\text{unlink}}(\text{guess} = \{\{\text{DID}_{11}, \text{DID}_{12}\}, \{\text{DID}_{21}, \text{DID}_{22}\}\})] + \\ &\Pr[\mathcal{A}_{\text{unlink}}(\text{guess} = \{\{\text{VC}_{11}, \text{VC}_{12}\}, \{\text{VC}_{21}, \text{VC}_{22}\}\})] = \\ &\Pr[\text{Exp}_{\mathcal{A}_{\text{ECC}}}(k) = 1] \cdot \Pr[\text{Exp}_{\mathcal{A}_p}(k) = 1] + \\ &\Pr[\text{Exp}_{\mathcal{A}_{\text{ECC}}}(k) = 1] \cdot \Pr[\text{Exp}_{\mathcal{A}_{\text{SPS}}}(k) = 1] = \\ &\text{Adv}_{\mathcal{A}_{\text{ECC}}}(k) \cdot \text{Adv}_{\mathcal{A}_p}(k) + \text{Adv}_{\mathcal{A}_{\text{ECC}}}(k) \cdot \text{Adv}_{\mathcal{A}_{\text{SPS}}}(k) \end{aligned} \quad (10)$$

因此，如果攻击者 $\mathcal{A}_{\text{ECC}}$ 成功攻击椭圆曲线加密方案， $\mathcal{A}_p$ 成功攻击 Paillier 同态加密系统， $\mathcal{A}_{\text{SPS}}$ 成功攻击 SPS 方案，则 $\mathcal{A}_{\text{unlink}}$ 在不可链接性模拟攻击博弈中获得胜利。然而，根据对基本组件安全性的假设可知，敌手 $\mathcal{A}_{\text{unlink}}$ 攻击成功的概率可以忽略不计，因此，本文方案满足不可链接性。

5.2 可追溯性分析

本文方案的可追溯性满足以下结论。

结论 2：若椭圆曲线加密方案满足基本安全性质、Paillier 同态加密系统满足基本安全性质，则本文面向元宇宙的数字公民身份认证方案满足可追溯性。

证明：定义 $\mathcal{A}_{\text{trace}}$ 为攻击本文方案可追溯性模拟攻击博弈的敌手， $\mathcal{A}_{\text{ECC}}$ 为攻击椭圆曲线加密方案的敌手， $\mathcal{A}_p$ 为攻击同态加密系统的敌手。假设 $\mathcal{A}_{\text{trace}}$ 成功攻击了该方案的可追溯性，定义一个多项式时间算法 $\mathcal{A}_\psi \in (\mathcal{A}_{\text{ECC}}, \mathcal{A}_p)$ 包含攻击椭圆曲线加密方案、Paillier 同态加密系统等攻击者的能力。在可追溯性模拟攻击博弈中 $\mathcal{A}_{\text{trace}}$ 的查询和 $\mathcal{A}_\psi$ 的响应交互来构造 $\mathcal{A}_\psi$ ，使其能够攻击椭圆曲线加密方案和 Paillier 同态加密系统，即若敌手 $\mathcal{A}_{\text{trace}}$ 成功攻击本文方案的可追溯性，则 $\mathcal{A}_\psi$ 就能以某一概率成功攻击椭圆曲线加密方案和 Paillier 同态加密系统。

根据可追溯性定义的步骤，算法 $\mathcal{A}_\psi$ 和敌手 $\mathcal{A}_{\text{trace}}$ 的交互运行如下。

(1) 生成阶段：算法 $\mathcal{A}_\psi$ 执行系统建立、密钥生成算法，通过分别运行 $\mathcal{A}_{\text{ECC}}$ 、 $\mathcal{A}_p$ 在椭圆曲线加密方案和 Paillier 同态加密系统模拟攻击博弈中产生的公共参数，生成可追溯性模拟攻击博弈的公开参数 P 和 TAC 的公钥 $\text{PK}_{\text{paillier}}^{\text{TAC}}$ ，并将其发送给 $\mathcal{A}_{\text{trace}}$ ，保留 TAC 私钥 $\text{SK}_{\text{paillier}}^{\text{TAC}}$ 。

(2) 询问阶段：敌手 $\mathcal{A}_{\text{trace}}$ 可以对算法 $\mathcal{A}_\psi$ 进行

多项式有界次数的询问。

①DID 询问: 敌手 $\mathcal{A}_{\text{trace}}$ 指定 SP 的公开信息 SPI, 请求询问用户 DC 对应的临时身份 DID, 算法 $\mathcal{A}_{\mathcal{V}}$ 通过运行 $\mathcal{A}_{\text{ECC}}$ 和 $\mathcal{A}_p$ 在模拟攻击博弈中的 DID 询问, 将得到的 DID 询问结果返回给 $\mathcal{A}_{\text{trace}}$ 。

②验证询问: 敌手 $\mathcal{A}_{\text{trace}}$ 选择 DID 和校验码 C 请求询问, 算法 $\mathcal{A}_{\mathcal{V}}$ 通过运行 $\mathcal{A}_{\text{ECC}}$ 和 $\mathcal{A}_p$ 在模拟攻击博弈中的验证询问, 将得到的验证结果返回给 $\mathcal{A}_{\text{trace}}$ 。

③追溯询问: 敌手 $\mathcal{A}_{\text{trace}}$ 选择一个 DID 请求询问, 算法 $\mathcal{A}_{\mathcal{V}}$ 通过运行 $\mathcal{A}_{\text{ECC}}$ 和 $\mathcal{A}_p$ 在模拟攻击博弈中的追溯询问, 将得到的结果返回给 $\mathcal{A}_{\text{trace}}$ 。

(3) 挑战阶段: 最后, 敌手 $\mathcal{A}_{\text{trace}}$ 输出一个 DID, 若出示的 DID 有效, 但还原出的个人信息不可用, 则说明敌手成功进行了攻击。

敌手 $\mathcal{A}_{\text{trace}}$ 成功进行攻击的概率为:

$$\begin{aligned} \text{Adv}_{\mathcal{A}_{\text{trace}}}(k) &= \Pr[\text{Exp}_{\mathcal{A}_{\text{trace}}}(k) = 1] = \\ &= \Pr\left[\begin{array}{l} \text{VerId}(\text{DID}, C, \text{AF}_{\text{SP}}, \text{SPI}, r_{\text{SP}}, \text{SK}_{\text{SP}}) = 1, \\ \text{TraceId}(\text{DID}, \text{SK}) = \phi \end{array}\right] = \\ &= \Pr[\text{Exp}_{\mathcal{A}_{\text{ECC}}}(k) = 1] \cdot \Pr[\text{Exp}_{\mathcal{A}_p}(k) = 1] = \\ &= \text{Adv}_{\mathcal{A}_{\text{ECC}}}(k) \cdot \text{Adv}_{\mathcal{A}_p}(k) \end{aligned} \quad (11)$$

因此, 如果攻击者 $\mathcal{A}_{\text{ECC}}$ 成功攻击椭圆曲线加密方案, 攻击者 $\mathcal{A}_p$ 成功攻击 Paillier 同态加密系统, 则 $\mathcal{A}_{\text{trace}}$ 在可追溯性模拟攻击博弈中获得胜利。然而根据对基本组件安全性的假设可知, 敌手 $\mathcal{A}_{\text{trace}}$ 攻击成功的概率可以忽略不计, 因此, 本文方案满足可追溯性。

6 性能分析

本节从时间开销以及区块链 Gas 开销两方面将

本文所提两种匿名可监管身份认证方案与参考文献[12]和参考文献[14]的方案进行对比。其中时间开销仿真实验运行在 PC (Windows 11 系统, 3.7 GHz CPU, 16 GB 内存) 的 IntelliJ IDEA 2022.1.1 上, 而区块链 Gas 开销仿真实验则基于浏览器 IDE (Remix) 以及以太坊测试链 (Rinkeby), 通过 Chrome v102.0.5005.63 浏览器上的 metamask 插件连接 Remix 和 Rinkeby。

6.1 时间开销对比

本节通过仿真实验对本文所提 AGAA 方案和 AMAA 方案的时间性能进行量化统计, 并与参考文献[12]和参考文献[14]的方案进行对比。通过仿真实验得到的基本运算耗费时间见表 2, 其中 T_B 表示基于 Paillier 同态算法的盲化运算, T_M 是取模乘法运算, T_{AE} 是 ECC 加密运算, T_{AD} 是 ECC 解密运算, T_{PE} 是 Paillier 加密运算, T_{PD} 是 Paillier 解密运算, T_H 是 SHA256 哈希运算, T_{Sig} 、 T_{Ch} 以及 T_{Ver} 分别是 SPS 方案中的签名运算、签名修改运算以及签名验证运算, 实验结果是算法运行 500 次的平均值。

表 3 展示了 AGAA 方案和 AMAA 方案与参考文献[12]和参考文献[14]的方案在时间开销上的对比结果。

由表 3 可知, 在身份认证阶段, Zhou T 等人^[12]使用智能合约计算用户所提交签名对应的公钥地址, 并于以太坊记录一笔交易, 此过程耗费时间约为 6 700 ms^[22]; Samir E 等人^[14]采用 shamir 秘密共享方案、非对称加密算法以及 SHA256 算法实现 VC 的分发, 并验证用户身份, 根据本文对身份信息定义, 笔者参考当参与方为 3 个时参考文献[14]给出的时间消耗, 其值为 1 422 ms; 在身份认证阶段, AGAA 方案共执行一次盲化运算、两次取模乘

表 2

基本运算耗费时间

操作	T_B	T_M	T_{AE}	T_{AD}	T_{PE}	T_{PD}	T_H	T_{Sig}	T_{Ch}	T_{Ver}
时间/ms	0.082	0.072	1.386	0.213	0.141	0.172	0.037	10.002	10.000	26.676

表 3

时间开销

方案	身份认证/ms	监管/ms
参考文献[12]的方案	约 6 700	—
参考文献[14]的方案	1 422	—
AGAA 方案	$T_B + 2T_M + T_{\text{AE}} + T_{\text{AD}} + T_{\text{PE}} = 1.966$	$T_{\text{PD}} + T_H = 0.209$
AMAA 方案	$1.966 + 2T_{\text{AE}} + 2T_{\text{AD}} + T_{\text{Sig}} + T_{\text{Ch}} + T_{\text{Ver}} = 49.842$	$T_{\text{PD}} + T_H = 0.209$

法运算、一次 ECC 加密运算、一次 ECC 解密运算以及一次 Paillier 加密运算, 耗费的时间总和为 1.966 ms; 而 AMAA 方案在此基础上增加了两次 ECC 加密运算、两次 ECC 解密运算、一次 SPS 签名运算、一次 SPS 签名修改运算以及一次 SPS 签名验证运算, 共计耗费时间 49.842 ms。从表 3 中可以看出, 无论是 AGAA 方案还是 AMAA 方案, 耗费的时间均远小于其他方案。

由于参考文献[12]和参考文献[14]缺少监管机制, 因此, 本文只对 AGAA 和 AMAA 两个方案的监管阶段进行量化分析。由表 3 可知, AGAA 方案共执行一次 Paillier 解密运算和一次 SHA256 哈希运算, 耗费的时间为 0.209 ms; 而 AMAA 方案的监管流程与 AGAA 方案一致, 耗费的时间同为 0.209 ms。由表 3 可知, AGAA 和 AMAA 两个方案在监管阶段同样具有极低的时间开销。

6.2 区块链 Gas 开销对比

本节通过仿真实验将本文方案与参考文献[12]和参考文献[14]的方案在区块链 Gas 开销方面进行对比。由于实体与区块链之间的交互主要包括读和写两种, 因此, 笔者将智能合约部署到 Rinkeby 中以记录交易数据, 并统计智能合约在部署和调用时耗费的 Gas 值。

图 5 展示了本文 AGAA 和 AMAA 方案与参考文献[12]和参考文献[14]的方案在区块链 Gas 开销上的对比结果。

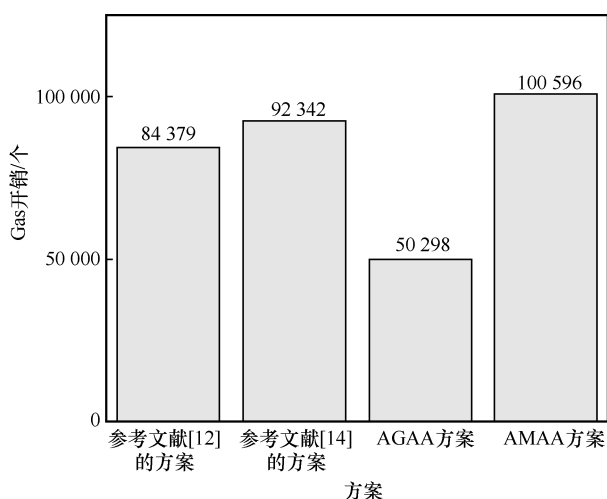


图 5 区块链 Gas 开销

由图 5 可知, 参考文献[12]调用合约来确定用户提交签名的公钥地址, 耗费的 Gas 值为 84 379; 参考文献[14]分别将 40 B 的 DID 和验证阈值、60 B

的 DID 文档、32 B 的哈希值以及约 50 B 的元组数据写入区块链, 耗费的 Gas 值为 92 342; 在系统初始化过程, 本文智能合约被写入区块链中, 耗费的 Gas 值为 176 311; 在 AGAA 方案中, 数字公民将 274 B 的 DID 以及密态校验码记录在区块链上, 而 SP 则将 65 B 的身份认证结果写入区块链, 耗费的 Gas 值为 26 811+23 487=50 298; 在此基础上, AMAA 方案要求用户将 274 B 的 DID 和密态属性值记录到区块链中, SP 将 65 B 的 VC 验证结果写入区块链, 该方案耗费的 Gas 值是 AGAA 的两倍, 即 100 596。由图 5 可知, AGAA 方案耗费的 Gas 值要低于其他方案, 而 AMAA 方案耗费的 Gas 值要高于其他方案, 但对于实际应用场景来说仍然是可以接受的。此外, 相较于其他方案, 实体交互行为的上链方便了对恶意用户的问责。

7 结束语

本文提出一种面向元宇宙的数字公民身份认证方案, 方案结合椭圆曲线加密方案、Paillier 同态加密系统以及基于等价类的保持结构性签名方案, 针对元宇宙海量网络服务普遍存在的两类身份分别设计身份认证方案。一方面, 本文解决了大数据分析下用户行为被链接的问题, 通过为用户在不同场景生成不同 DID 以及采用 SPS 方案构建包含可变签名的 VC, 保护用户的行为隐私; 同时, 使用代码嵌入实现 VC 的有限访问, 保障用户的被遗忘权。另一方面, 本文方案实现了隐私保护和公正监管的平衡, 基于 Paillier 同态加密系统构建 DID, 使其同时具备匿名、可认证以及可监管三重属性。此外, 引入联合监管机制, 有效避免了监管机构权力滥用。

参考文献:

- [1] GRIDER D, MAXIMO M. The metaverse: Web 3.0 virtual cloud economies[J]. Grayscale Research, 2021.
- [2] WANG Y T, SU Z, ZHANG N, et al. A survey on metaverse: fundamentals, security, and privacy[J]. arXiv preprint, 2022, arXiv:2203.02662.
- [3] WANG F Y. Parallel intelligence in metaverses: welcome to Hanoi![J]. IEEE Intelligent Systems, 2022, 37(1): 16-20.
- [4] 沈阳. 元宇宙发展研究报告 2.0 版[R]. 2022. SHEN Y. Metaverse development research report version 2.0[R]. 2022.
- [5] ALOMAR N, ALSALEH M, ALARIFI A. Social authentication applications, attacks, defense strategies and future research directions: a systematic review[J]. IEEE Communications Surveys & Tutorials, 2017, 19(2): 1080-1111.

- [6] DUNPHY P, PETITCOLAS F A P. A first look at identity management schemes on the blockchain[J]. IEEE Security & Privacy, 2018, 16(4): 20-29.
- [7] ALPÁR G, HOEPMAN J H, SILJEE J. The identity crisis. security, privacy and usability issues in identity management[J]. Journal of Information System Security, 2011:1-15.
- [8] CHRISTOPHER A. The path to self-sovereign identity[Z]. 2016.
- [9] W3C. Decentralized identifiers (DIDs) v1.0[Z]. 2021.
- [10] W3C. Verifiable credentials data model v1.1[Z]. 2022.
- [11] BANDARA E, LIANG X P, FOYTIK P, et al. A blockchain and self-sovereign identity empowered digital identity platform[C]//Proceedings of 2021 International Conference on Computer Communications and Networks. Piscataway: IEEE Press, 2021: 1-7.
- [12] ZHOU T, LI X F, ZHAO H. EverSSDI: blockchain-based framework for verification, authorisation and recovery of self-sovereign identity using smart contracts[J]. International Journal of Computer Applications in Technology, 2019, 60(3): 281.
- [13] BELCHIOR R, PUTZ B, PERNUL G, et al. SSIBAC: self-sovereign identity based access control[C]//Proceedings of 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications. Piscataway: IEEE Press, 2020: 1935-1943.
- [14] SAMIR E, WU H Y, AZAB M, et al. DT-SSIM: a decentralized trustworthy self-sovereign identity management framework[J]. IEEE Internet of Things Journal, 2022, 9(11): 7972-7988.
- [15] STOKKINK Q, ISHMAEV G, EPEMA D, et al. A truly self-sovereign identity system[C]//Proceedings of 2021 IEEE 46th Conference on Local Computer Networks. Piscataway: IEEE Press, 2021: 1-8.
- [16] VOIGT P, VON DEM BUSSCHE A. The EU general data protection regulation (GDPR)[M]. Switzerland: Springer, 2017.
- [17] KOBLITZ N, MENEZES A, VANSTONE S. The state of elliptic curve cryptography[J]. Designs, Codes and Cryptography, 2000, 19(2): 173-193.
- [18] PAILLIER P. Public-key cryptosystems based on composite degree residuosity classes[M]//Advances in cryptology - EUROCRYPT'99. Heidelberg: Springer, 1999: 223-238.
- [19] HANSER C, SLAMANIG D. Structure-preserving signatures on equivalence classes and their application to anonymous credentials[C]//Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security. Heidelberg: Springer, 2014: 491-511.
- [20] SOLTANI R, NGUYEN U T, AN A J. A survey of self-sovereign identity ecosystem[J]. Security and Communication Networks, 2021, 2021: 8873429.
- [21] LYU Q Y, CHENG S P, LI H, et al. NSSIA: a new self-sovereign identity scheme with accountability[J]. arXiv preprint, 2022 arXiv: 2206.04911.
- [22] CHEN J, YAO S X, YUAN Q, et al. CertChain: public and efficient

certificate audit based on blockchain for TLS connections[C]//Proceedings of IEEE INFOCOM 2018 - IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2018: 2060-2068.

[作者简介]



吕秋云（1978—），女，博士，杭州电子科技大学网络空间安全学院副教授、硕士生导师，主要研究方向为身份认证、区块链应用、隐私保护。



程绍鹏（1996—），男，杭州电子科技大学网络空间安全学院硕士生，主要研究方向为数字身份管理、区块链。



杨满智（1979—），男，澳门科技大学博士生，恒安嘉新（北京）科技股份有限公司首席技术官，主要研究方向为网络安全、机器学习。



陈晓光（1982—），男，澳门科技大学博士生，恒安嘉新（北京）科技股份有限公司首席执行官，主要研究方向为网络安全、机器学习。



王震（1984—），男，博士，杭州电子科技大学网络空间安全学院副教授、博士生导师，杭州电子科技大学卓越学院副院长，主要研究方向为人工智能安全、博弈论、多智能体系统和网络科学。