计 算 机 研 究 与 发 展
Journal of Computer Research and Development
https://crad.ict.ac.cn
62(10): 2416−2427, 2025

基于区块链的轻量级工业物联网跨域认证与数据共享方案

李哲成　张　波
（济南大学信息科学与工程学院　济南　250022）
（ise_zhangb@ujn.edu.cn）

A Lightweight Blockchain-Based Cross-Domain Authentication and Data Sharing Scheme for the Industrial Internet of Things

Li Zhecheng and Zhang Bo
（*School of Information Science and Engineering*, *University of Jinan*, *Jinan* 250022）

Abstract　In the current context of cross-domain identity authentication and data sharing for industrial Internet of things (IIoT) devices, edge servers and blockchain technology play vital roles in enabling rapid data forwarding and secure storage. Existing mechanisms for cross-domain authentication and data sharing are facing a range of security risks, and are further challenged by high computational and communication overhead at the device level, as well as by overly complex authentication structures. To address these problems, we propose a lightweight cross-domain authentication and data sharing scheme based on blockchain technology, specifically designed for IIoT scenarios. The proposed scheme enables anonymous cross-domain authentication and secure data sharing for industrial devices. Furthermore, we introduce a distributed key generation mechanism to construct service keys. Based on service keys, a lightweight message authentication algorithm is designed to ensure the confidentiality of service-related information at the device side, while minimizing resource consumption to the greatest extent. A detailed security analysis demonstrates that the proposed scheme satisfies the critical security properties of unforgeability, confidentiality, unlinkability, and anonymity. In addition, performance evaluation results show that the proposed scheme achieves superior performance compared with existing solutions in terms of computational overhead, communication cost, and the efficiency of blockchain query operations.

Key words　cross-domain; industrial Internet of things (IIoT); authentication; data sharing; distributed key generation; blockchain

摘　要　在当前工业物联网（industrial Internet of things, IIoT）设备的跨域身份认证和数据共享中，边缘服务器和区块链起到了快速转发和存储的作用. 现有的跨域认证与数据共享方案面临着多种安全风险，并且还存在着设备端计算和通信开销大、认证结构复杂的问题. 因此，提出了一种基于区块链的轻量级工业物联网跨域认证与数据共享方案，能够实现设备的跨域匿名认证与数据共享. 同时，引入了分布式密钥生成技术来构建服务密钥，并以此设计了一种轻量级的消息认证算法，能有效实现服务信息的设备端保密，同时极大限度地减少了资源消耗. 详细的安全性分析表明该方案满足不可伪造性、机密性、不可链接性和匿名性. 性能分析表明该方案在计算开销、通信开销和区块链查询效率上均优于相关方案.

关键词　跨域；工业物联网；认证；数据共享；分布式密钥生成；区块链

收稿日期：2025-06-06；**修回日期**：2025-07-28
基金项目：山东省自然科学基金项目（ZR2022MF264）
This work was supported by the Natural Science Foundation of Shangdong Province (ZR2022MF264).
通信作者：张波（ise_zhangb@ujn.edu.cn）

中图法分类号 TP309

DOI: 10.7544/issn1000-1239.202550402 CSTR: 32373.14.issn1000-1239.202550402

随着信息通信技术的持续发展，物联网（Internet of things, IoT）产业呈现出高速发展的态势[1]，各类具备不同功能与适配不同服务的物联网终端设备不断涌现. 在此背景下，工业物联网（industrial Internet of things, IIoT）[2-3]作为推动传统工业体系迈向智能制造与自动化的重要技术基础，正在各类工业场景中发挥着关键作用. IIoT 通过将众多工业设备、传感器及控制单元互联互通，实现了数据的感知、远程控制与协同调度，从而显著提升了工业系统的智能化水平和运行效率[4]. 在典型的工业物联网体系中，不同服务对设备的运行结构提出了差异化的功能需求与架构设计. 设备通过部署多类型传感器，可以实现对工业环境和生产流程的持续监测和实时采集[5-6]. 在较高复杂度的工业系统中，设备同时具备局部控制、分析和与其他设备的协同能力. 通过统一的通信协议及分布式调度，设备可实现信息共享、任务分配与行为协同，完成复杂的工业任务[7].

随着云计算、物联网和人工智能等技术的融合，边缘计算（edge computing）[8-9]成为传统 IIoT 体系架构的重要补充. 在 IIoT 体系中，边缘节点通常部署在靠近数据源的工厂内，可以有效解决在此之前网络云平台服务中存在的数据传输时延高、响应速度慢和系统可靠性不足等问题. 在工业场景中，边缘网关作为边缘计算的核心组件，具备协助认证、通信协议转换的功能，可以在无需频繁访问网络云平台的前提下完成初步的数据处理和消息转发.

然而，尽管 IIoT 架构在实际部署过程中存在着安全性与可扩展性方面的挑战. 一方面，IIoT 终端设备资源受限，容易成为网络攻击的薄弱环节，尤其在设备身份匿名的情况下，难以精准定位数据来源，影响了对异常行为的检测与溯源. 另一方面，大规模异构设备分布在各个工厂，不同工厂由于信任关系不同构成了一个个信任域. 正因如此，物联网设备间的数据共享与跨域访问存在着复杂的信任管理与认证机制，然而传统基于中心化架构的解决方案难以满足多方协作、高速响应以及隐私保护的实际需求. 此外，数据在传输与处理过程中容易遭受篡改和假冒攻击，严重威胁 IIoT 系统的安全性与稳定性. 因此，亟需一个同时兼顾安全性、隐私性和轻量化的数据管理与认证方案，以提升 IIoT 系统在开放网络环境下的抗攻击能力和协同效率.

为应对上述挑战，本文提出了一种基于区块链的轻量级工业物联网跨域认证与数据共享方案，该方案在保障数据隐私和系统安全的同时，实现了高效可信的跨域信息交互. 本文的主要贡献包括 3 个方面：

1）设计了一个基于区块链的 IIoT 跨域认证与数据共享框架，利用区块链的去中心化、不可篡改特性和边缘服务器的快速消息转发能力，实现各域之间的信任传递与数据互通. 本文方案将边缘服务器作为协同参与节点，引导服务信息在多个信任域间进行安全传输，并通过基于服务的访问控制机制实现了对服务共享权限的动态管理. 设备发起请求只需与边缘服务器交互，减轻了设备端的通信压力. 动态累加器的身份存储形式减轻了区块链的存储压力并提升了查询效率与身份验证的便捷性，降低了系统的部署成本和维护复杂度，适用于资源受限的物理网设备.

2）针对传统认证机制在 IIoT 中的计算与存储开销问题，本文结合分布式密钥生成技术，设计了一种基于服务的轻量级跨域消息认证算法. 本方案在不利用边缘服务器高计算能力以及保证设备端消息保密和身份匿名的前提下，相较以往方案实现了总体效率的显著提升.

3）对所提出的方案进行了安全性分析和性能分析. 安全性分析表明，该方案能够有效抵御选择明文攻击，具备保密性、匿名性、不可链接性和不可伪造性. 同时，性能分析表明方案具有较低的计算延迟和通信延迟，满足实际应用的需求.

1 相关工作

传统密码学认证协议作为构建安全通信系统的基础，其设计、分析与优化已被广泛研究. Schnorr[10]提出了一种面向资源受限设备的高效身份认证与签名方案，其核心思想是在传统离散对数问题基础上，引入“随机承诺—挑战—响应”机制，通过离线计算签名中与消息无关的部分，使得在实际应用时的签名生成过程中，设备仅需执行极少量运算，从而降低了认证过程中的计算和通信开销. 但该方案对伪随机数生成高度依赖，若预处理环节中的随机性不足，可能影响安全性. Ateniese 等人[11]提出了一种可证明安全的抗合谋群签名方案，该方案将群签名技术与

身份托管机制结合，通过签名算法实现了匿名性与可追踪性，设计了一种高效的成员加入协议，并支持群管理者追溯和公开签名者身份. Burrows 等人[12]提出了一种基于信任推理的 BAN 认证逻辑框架，通过形式化语言描述并分析认证协议中的交互过程，以此验证通信各方对密钥协商身份的信任建立等过程的合理性，并阐述了协议设计中的冗余与漏洞. Boneh 等人[13]提出了一种短签名长度的群签名方案，该方案结合双线性映射与非交互式零知识证明，允许群成员对消息进行匿名签名，并允许成员动态加入与撤销. 在效率方面，短签名的设计突破了传统群签名在签名大小与验证效率方面的性能瓶颈.

为了实现 IIoT 实体之间的安全认证和数据安全高效的传输，Karati 等人[14]提出了一种轻量级且可证明安全的无证书签名方案来解决传统基于证书的方案中身份认证效率低的问题. Cui 等人[15]提出了一种结合群签名与代理重加密技术的匿名消息认证机制，解决了 IIoT 系统中半可信边缘节点获取敏感数据的问题. Yang 等人[16]提出了一种基于动态认证凭证框架的轻量级 AKA 协议. 该方案中通过结合对称加密与哈希函数，在无需公钥算法的前提下实现了完美前向安全性. 然而，该方案依赖集中式网关作为的认证中介，存在单点故障的风险. Xiong 等人[17]提出了一种基于代理重签名的异构工业物联网设备间的认证协议，用以支持基于身份的加密与无证书加密之间的安全通信. Hou 等人[18]提出了一种基于证书的并行密钥隔离聚合签名方案，该方案中攻击者即便获得所有签名者的私钥，也无法伪造合法的聚合签名，并同时保证了较短的签名长度和较高的验证效率. Xu 等人[19]提出了一种融合物理层与密码层的跨层设备认证框架，该框架通过预编码和随机加密并利用离散量子行走生成一次性密钥来实现设备身份的隐私保护. Cui 等人[20]提出了一种基于边缘计算的高效批量认证方案，该方案中使用边缘服务器来代替 IIoT 设备进行认证运算，从而显著提升了认证过程的效率. 张应辉等人[21]提出了一种可验证的 IIoT 属性基签名方案，方案中半可信服务器仅生成部分签名，这使得方案可以有效抵抗伪造攻击和合谋攻击.

目前，越来越多的学者致力于利用区块链实现设备间的身份认证与数据共享，Tan 等人[22]提出了一种针对工业无人机网络的分布式轻量级认证方案，通过使用区块链智能合约实现了无人机身份信息的存储与管理，并设计了一种高度灵活的双边密钥协商和群组密钥协商算法，解决了工业无人机高动态网络中的通信安全与数据丢包问题. Wang 等人[23]提出了一种结合区块链与无配对无证书签名的 IIoT 消息认证方案，该方案使用了基于以太坊的密钥生成中心来取代传统的集中式密钥生成中心，有效解决了单点故障问题. Yu 等人[24]设计了一种基于区块链的安全访问控制方案，该方案通过区块链实现了恶意用户公钥的可追踪与撤销.

进一步地，针对 IIoT 中不同域之间的设备认证与数据共享问题，一些学者开始探索基于区块链的跨域安全机制，Shen 等人[25]提出一种基于联盟链的跨域安全认证机制，设计了一种基于身份的加密算法，实现了跨域设备匿名认证与安全密钥协商. Singh 等人[26]提出了一种基于区块链的多层次签名跨域数据共享安全框架，该方案在实现轻量级密钥协商与认证的同时，设计了一种基于智能合约的审计与惩罚机制，实现了对恶意设备的全局可追溯. Zhang 等人[27]结合物理不可克隆函数(PUF)与动态累加器技术，设计了一种 IIoT 多因子认证协议，实验证明该方案具有较高的效率和可扩展性. 在消息认证与边缘协同方面，Wang 等人[28]提出了一种基于区块链的轻量级跨域消息认证框架，实现了跨域多接收者认证场景下的设备认证. 然而，该方案中域内所有设备的服务公私钥由域内头节点设备负责生成并分发，存在单点故障与隐私泄露的风险. Zeng 等人[29]提出了一种区块链辅助的跨域数据共享框架，该框架引入零知识证明技术与密钥协商协议，解决了 IIoT 跨域环境下数据所有权不明和隐私泄露的问题. Wang 等人[30]将广播加密与代理重加密技术相结合，设计了一种边缘服务器作为数据拥有者和接收者的跨域数据共享协议，实现匿名智能设备的本地更新和数据的灵活共享，但该方案未考虑设备端的消息隐私保护.

2 预备知识

本节主要介绍文中用到的技术和相关知识，其中包括双线性映射、ECC 复合定理、离散对数困难问题假设和 CDH 困难问题假设.

2.1 双线性映射

设 2 个循环群G和G_T分别为加法群和乘法群，群的阶数q为根据安全参数λ生成的安全大素数. 双线性配对是一个映射$G\times G\to G_T$，其满足以下 3 个性质：

1)双线性. 对于$\forall P_1,P_2,P_3\in G$且$a,b\in Z_q^*$，有

$$e(P_1,P_2+P_3)=e(P_1,P_2)e(P_1,P_3),$$

$$e(aP_2,bP_3)=e(P_2,P_3)^{ab}.$$

2)非退化性. u,v为G的生成元, $e(u,v)\neq 1_G$.

3)可计算性. 对于所有的$u,v\in G$, 存在有效的算法可以计算出$e(u,v)$.

2.2 ECC 复合定理

当前已证, 给定多对公钥$(Q_1,Q_2,\cdots,Q_n)$和私钥$(s_1,s_2,\cdots,s_n)$, 公私钥关系为:

$$(Q_1=s_1\cdot P,Q_2=s_2\cdot P,\cdots,Q_n=s_n\cdot P).$$

多个公私钥各自的和可以组成一对新的公私钥, 例如, 私钥的和为$s=(s_1,s_2,\cdots,s_n)\bmod q$、公钥的和为$Q=(Q_1,Q_2,\cdots,Q_n)$. (s,Q)就构成了一对新的公私钥, 并且同样满足$Q=s\cdot P$.

2.3 困难问题假设

1)离散对数困难问题假设

设$a\in Z_q^*$, P是G的生成元, $Q=a\cdot P$. 给定P和Q, 计算出a即为离散对数困难问题. 如果离散对数困难问题的求解在计算上不可行, 则离散对数困难假设成立.

2)计算性 Diffie-Hellmana(CDH)困难问题假设

设$a,b\in Z_q^*$, P是G的生成元. 给定aP和bP, 计算出abP即为 CDH 困难问题. 如果 CDH 困难问题的求解在计算上不可行, 则 CDH 困难问题假设成立.

3 方案模型

本节主要介绍方案模型, 包括方案的系统模型、安全目标、形式化定义和安全模型.

3.1 系统模型

本文方案的系统模型如图 1 所示, 其中包括以下 4 个实体:

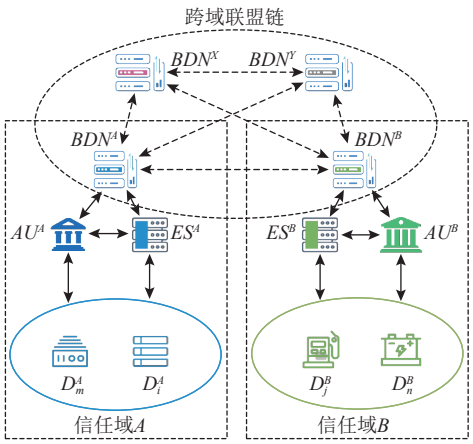


Fig. 1 System model

图 1 系统模型

1)权威机构(*AU*). 每个信任域内都存在一个权威机构, *AU* 为完全可信实体, 主要负责生成系统参数和为各实体注册, 同时参与为设备分布式生成服务密钥.

2)区块链域节点(*BDN*). 每个信任域都存在一个区块链域节点, 所有 *BDN* 共同构成跨域联盟链. 不同域的 *BDN* 可以通过共识发布智能合约. 在各个域内, *AU* 和边缘服务器(*ES*)可以通过 *BDN* 访问链上信息.

3)*ES*. 每个域内都存在一个 *ES*, *ES* 为诚实但好奇的半可信实体. 该系统中 *ES* 主要负责为其他实体转发服务的相关消息并协助 *AU* 为设备生成服务密钥.

4)设备(*D*). 在工业物联网中, *D* 主要负责定期收集工厂内各项服务数据并记录在本地, 需要在 *ES* 的协助下与其他实体通信.

3.2 安全目标

本文所提出的方案需要达成以下安全目标:

1)不可伪造性. 攻击者无法构造出有效的签名, 使得该签名能够通过验证过程, 进而假冒合法用户的身份或信息.

2)机密性. 传输的信息中不能泄露服务信息明文, 且攻击者无法从服务信息密文中计算出明文.

3)不可链接性. 攻击者无法确定拦截的 2 条消息是由同一设备生成的.

4)匿名性. 攻击者无法从任何截获的信息中获取或计算出设备的真实身份.

5)设备端保密. 系统中的半可信边缘服务器无法从转发的服务信息密文中获取或计算出服务信息明文.

6)基于服务的访问控制. 未加入服务列表的恶意设备或第三方实体无法进行数据共享.

3.3 形式化定义

方案由 5 个算法组成:

1)设置算法 $Setup(\kappa)\rightarrow\{pp,msk^X,mpk^X\}$. 该算法由$AU^X$执行. 输入安全参数$\kappa$, 输出系统公共参数$pp$、系统主私钥$msk^X$和系统主公钥$mpk^X$, 其中$msk^X$由$AU^X$秘密持有.

2)用户注册算法 $Register(pp,ID_i^X)\rightarrow\{sk_{ES}^X,\ pk_{ES}^X,\ PID_{i,j}^X,sk_{i,j}^X,pk_{i,j}^X\}$, 该算法由$AU^X$执行. 输入系统公共参数$pp$和$D_i^X$的真实身份$ID_i^X$, 输出$ES^X$私钥$sk_{ES}^X$、$ES^X$公钥$pk_{ES}^X$、$D_i^X$伪身份$PID_{i,j}^X$、$D_i^X$私钥$sk_{i,j}^X$和$D_i^X$公钥$pk_{i,j}^X$, 其中$sk_{ES}^X$由$ES^X$秘密持有, $sk_{i,j}^X$由D_i^X秘密持有.

3)分布式生成服务密钥算法包括 2 个子算法:

①生成部分服务密钥算法$DPKG(serv,pp)\rightarrow$

$\{ask^X_{j,n}, apk^X_{j,n}, ask^j_{j,n}, apk^j_{j,n}\}$，该算法由$D^B_j$、$AU^A$和$AU^B$执行. 输入服务$serv$和公共参数$pp$，输出部分服务私钥$ask^X_{j,n}$、$ask^j_{j,n}$和部分服务公钥$apk^X_{j,n}$、$apk^j_{j,n}$.

②生成服务密钥算法$DSKG(ask^X_{j,n}, apk^X_{j,n}, ask^j_{j,n}, apk^j_{j,n}) \rightarrow \{sk^j_{serv}, pk^j_{serv}\}$，该算法由$D^B_j$、$AU^A$和$AU^B$执行. 输入部分服务私钥$ask^X_{j,n}$、$ask^j_{j,n}$和部分服务公钥$apk^X_{j,n}$、$apk^j_{j,n}$，输出$D^B_j$服务私钥$sk^j_{serv}$和服务公钥$pk^j_{serv}$，其中$sk^j_{serv}$由$D^B_j$秘密持有.

③生成设备状态相关信息算法$DSG(Inf_{serv}, sk^j_{serv}) \rightarrow \{s^j_{serv}, S^j_{serv}\}$，该算法由$D^B_j$执行. 输入设备相关信息$Inf_{serv}$和$D^B_j$服务私钥$sk^j_{serv}$，输出设备状态相关信息$(s^j_{serv}, S^j_{serv})$，其中$s^j_{serv}$由$D^B_j$秘密持有.

4)服务信息产生算法$SIG(sk^A_{i,x}, s^i_{serv}, S^i_{serv}, m, PID^A_{i,x}, pk^j_{serv}, pk^B_{j,y}) \rightarrow \{\sigma^A_i, c, SRI\}$，该算法由$D^A_i$执行. 输入$D^A_i$私钥$sk^A_{i,x}$、设备状态相关信息$(s^i_{serv}, S^i_{serv})$、服务信息$m$、$D^A_i$伪身份$PID^A_{i,x}$、$D^B_j$服务公钥$pk^j_{serv}$和$D^B_j$公钥$pk^B_{j,y}$，输出签名$\sigma^A_i$、密文$c$和$SRI$.

5)服务信息接收算法$SIR(c, sk^j_{serv}, sk^B_{j,y}, SRI, S^i_{serv}, pk^A_{i,x}, pk^j_{serv}) \rightarrow \{m, \sigma^A_i, \text{true or false}\}$，该算法由$D^B_j$执行. 输入服务信息密文$c$、$D^B_j$服务私钥$sk^j_{serv}$、$D^B_j$私钥$sk^B_{j,y}$、$SRI$、设备状态相关信息$S^i_{serv}$、$D^A_i$公钥$pk^A_{i,x}$和$D^A_i$服务公钥$pk^i_{serv}$，输出服务信息$m$、签名$\sigma^A_i$和签名验证结果 true or false.

3.4 安全模型

定义 1. 如果不存在 PPT 敌手$\mathcal{A}$能够以不可忽略的概率解决 CDH 问题，则签名算法满足选择消息攻击下的存在不可伪造性(EUF-CMA). 具体的安全模型如下：

1)$Setup$. 输入安全参数κ，模拟器$\mathcal{B}$输出公共参数并发送给$\mathcal{A}$.

2)H查询. $\mathcal{A}$使用DS_i向$\mathcal{B}$发起查询，其中DS_i为二进制字符串，$\mathcal{B}$查询到对应的哈希值$h \in Z^*_q$并发送给$\mathcal{A}$.

3)H_1查询. $\mathcal{A}$使用Inf_{serv_α}向$\mathcal{B}$发起查询，其中Inf_{serv_α}为二进制字符串，$\mathcal{B}$查询到对应值$\left(S^i_{serv_\alpha}, \eta\right)$并发送给$\mathcal{A}$.

4)H_2查询. $\mathcal{A}$使用$\left(PID^A_{i,x}, k_1, T\right)$向$\mathcal{B}$发起查询，$\mathcal{B}$查询到对应值$h_2 \in Z^*_q$并发送给$\mathcal{A}$.

5)H_3查询. $\mathcal{A}$使用$\left(m, S^i_{serv_\alpha}\right)$向$\mathcal{B}$发起查询，$\mathcal{B}$查询到对应值$h_3 \in Z^*_q$并发送给$\mathcal{A}$.

5)签名密钥查询. $\mathcal{A}$使用$PID^A_{i,x}$发起查询，$\mathcal{B}$查询到对应的签名密钥并发送给$\mathcal{A}$.

6)验证公钥查询. $\mathcal{A}$使用$\left(serv_\alpha, PID^A_{i,x}\right)$发起查询，$\mathcal{B}$查询到对应的验证密钥并发送给$\mathcal{A}$.

7)签名查询. $\mathcal{A}$使用m向模拟器$\mathcal{B}$发起查询，$\mathcal{B}$返回给$\mathcal{A}$一个有效签名.

8)验证公钥替换查询. $\mathcal{A}$可以任意选择新的验证公钥来替换$PID^A_{i,x}$原本对应的验证公钥.

9)伪造. 在执行上述查询结束后，$\mathcal{A}$在不执行签名查询的前提下伪造了一个签名. 如果该签名有效，则$\mathcal{A}$赢得该游戏，否则$\mathcal{A}$失败.

4 方案构造

本节主要介绍方案的具体构造，系统流程如图 2 所示，具体流程如下.

1)系统初始化

在新的信任域X加入到 IIoT 系统中时，权威机构AU^X按照以下步骤执行域X的初始化：

输入安全参数κ，AU^X选择阶为素数q的加法群$(G,+)$和乘法群$(G_T,\cdot)$，P为群G的生成元. 定义双线性映射为$e: G \times G \rightarrow G_T$. 之后，$AU^X$定义哈希函数$H: \{0,1\}^* \rightarrow G_T$，$H_1: G_T \rightarrow \{0,1\}^\kappa$，$H_2: G_T \rightarrow \{0,1\}^*$，$H_3: \{0,1\}^* \rightarrow Z^*_q$.

AU^X选择一个随机数$msk^X \in Z^*_q$作为系统主私钥，并计算$mpk^X = msk^X \cdot P$作为系统主公钥. 最后，AU^X将系统公共参数$(G, G_T, P, mpk^X, H, H_1, H_2, H_3)$通过$BDN^X$发布到跨域联盟链上.

2)用户注册

①AU^X选择一个随机数$sk^X_{ES} \in Z^*_q$作为ES^X的私钥，计算ES^X公钥$pk^X_{ES} = sk^X_{ES} \cdot P$并将$\{sk^X_{ES}, pk^X_{ES}\}$通过安全信道发送给$ES^X$.

②当有新的设备D^X_i加入信任域X时，AU^X为D^X_i选择真实身份ID^X_i，选择随机数$\varepsilon^X_{i,j}, sk^X_{i,j} \in Z^*_q$并计算$PID^X_{i,j} = H(ID^X_i \| \varepsilon^X_{i,j} \| T_j)$作为$D^X_i$的伪身份，其中$sk^X_{i,j}$作为$D^X_i$的私钥. AU^X继续计算$pk^X_{i,j} = sk^X_{i,j} \cdot P$作为$D^X_i$的公钥，并会在本地存储$ID^X_i$和$PID^X_{i,j}$之间的对应列表. 最后，$AU^X$将$\{PID^X_{i,j}, sk^X_{i,j}, pk^X_{i,j}, VT^X_{i,j}\}$通过安全信道发送给$D^X_i$，其中$VT^X_{i,j}$为$PID^X_{i,j}$的有效期.

值得注意的是，AU^X通常会同时为D^X_i生成多个伪身份和私钥以避免多次通信产生不必要的时间消耗，并会在验证$PID^X_{i,j}$为唯一值后添加到动态累加器中. 每当生成一个$PID^X_{i,j}$，AU^X会通过BCN^X查询得到当前链上的动态累加器，AU^X会检查是否有重复项$\phi = \prod_{j=1}^{n} PID^X_{i,j}$. 如果未重复，则将$PID^X_{i,j}$添加到动态累

加器中，否则重新生成.

在将伪身份添加到动态累加器后，AU^X会将$\{PID_{i,1}^X, sk_{i,1}^X, pk_{i,1}^X, VT_{i,1}^X\}$，$\{PID_{i,2}^X, sk_{i,2}^X, pk_{i,2}^X, VT_{i,2}^X\}$，…，$\{PID_{i,n}^X, sk_{i,n}^X, pk_{i,n}^X, VT_{i,n}^X\}$通过安全信道发送给$D_i^X$.

最后，AU^X会通过BCN^X将$\{mpk^X, pk_{ES}^X, \phi\}$写入跨域联盟链的分布式账本中.

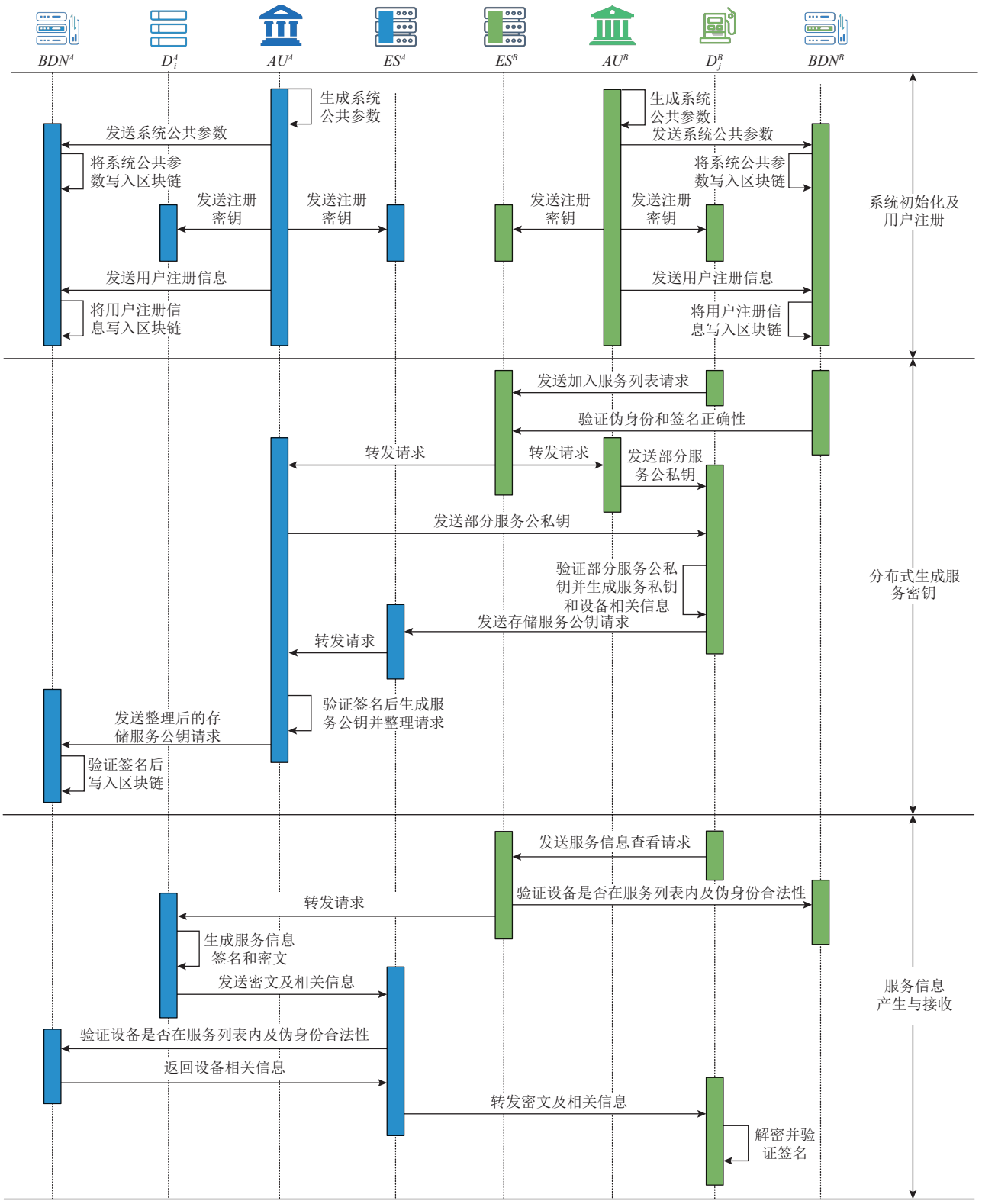


Fig. 2 System process

图 2 系统流程

3)分布式生成服务密钥

假设某 IIoT 服务 $serv$, 域 A 设备 D_i^A 为数据拥有者, 域 B 设备 D_j^B 为数据使用者. 在 D_j^B 想要加入跨域服务设备列表以获取服务信息时, 需要按照以下步骤申请服务密钥:

① D_j^B 使用 pk_{ES}^B 加密后向 ES^B 发送加入服务列表请求 $Req_{JS}=\{JS,serv,PID_{j,y}^B,pk_{j,y}^B,\sigma_{sk_{j,y}^B}\}$. 其中 JS 表示加入服务请求的消息类型标识符, $serv$ 表示服务名称, $\sigma_{sk_{j,y}^B}$ 表示使用 $sk_{j,y}^B$ 对消息进行的签名.

②收到请求后, ES^B 通过 BDN^B 在区块链上查询到当前的动态累加器并验证伪身份合法性, 之后使用 $pk_{j,y}^B$ 验证签名的正确性. 如果签名验证正确, ES^B 重新签名并将请求转发给 AU^A 和 AU^B.

③ AU^A 和 AU^B 收到请求后, 使用 pk_{ES}^B 验证签名的正确性. 如果签名验证通过, AU^A 和 AU^B 通过区块链共识从已加入系统的域中选择 n 个域的权威机构, 其中包括 AU^A 和 AU^B. AU^A 选择整数 t, 其中 $t<n$. n 个域的权威机构 AU^X 分别选择 $ask_{j,n}^X\in Z_q^*$ 并需要计算 $apk_{j,n}^X=ask_{j,n}^X\cdot P$.

之后, AU^X 会将部分服务公私钥通过安全信道发送给 D_j^B.

④ D_j^B 在收到 t 个部分服务公私钥对集合后首先逐一验证部分服务公私钥的正确性 $apk_{j,t}^X\overset{?}{=}ask_{j,t}^X\cdot P$. 验证通过后, D_j^B 选择 $ask_{j,1}^j,ask_{j,2}^j,ask_{j,3}^j\in Z_q^*$ 并计算 $apk_{j,1}^j=ask_{j,1}^j\cdot P, apk_{j,2}^j=ask_{j,2}^j\cdot P, apk_{j,3}^j=ask_{j,3}^j\cdot P$. D_j^B 调用随机函数对所有公私钥进行重新排序后得到服务公私钥集:

$$ask_{j,y}^B=\left(ask_{j,1},ask_{j,2},\cdots,ask_{j,t+3}\right),$$

$$apk_{j,y}^B=\left(apk_{j,1},apk_{j,2},\cdots,apk_{j,t+3}\right).$$

D_j^B 计算服务私钥:

$$sk_{serv}^j=\left(ask_{j,1}+ask_{j,2}+\cdots+ask_{j,t+3}\right)\bmod q.$$

得到服务私钥后, D_j^B 需要计算设备状态相关信息 $s_{serv}^j=H_1(Inf_{serv})$ 和 $S_{serv}^j=sk_{serv}^j\cdot s_{serv}^j$, 其中 Inf_{serv} 表示 D_j^B 的设备相关信息. 需要注意的是, Inf_{serv} 和 s_{serv}^j 仅在 D_j^B 本地保存.

⑤ d_j^B 使用 pk_{ES}^B 加密后将存储服务公钥请求 $Req_{st}=\left(st,PID_{j,y}^B,apk_{j,y}^B,S_{serv}^j,\sigma_{sk_{j,y}^B}\right)$ 发送给 ES^B. 其中 st 表示存储服务公钥请求的消息类型标识符, ES^B 在收到服务公钥集后验证签名, 如果验证签名通过, 则重新签名并转发给 AU^A.

⑥ AU^A 在收到请求后验证签名, 签名验证通过后计算

$$pk_{serv}^j=apk_{j,1}+apk_{j,2}+\cdots+apk_{j,t+3}.$$

得到 D_j^B 的服务公钥 pk_{serv}^j 后, AU^A 将请求格式整理为 $Req'_{st}=\left(st,PID_{j,y}^B,pk_{j,y}^B,pk_{serv}^j,S_{serv}^j,VT_{j,y}^X,\sigma_{msk^A}\right)$ 并发送给 BDN^A. BDN^A 收到后验证签名, 如果验证结果为正确则将 $\left(PID_{j,y}^B,pk_{j,y}^B,pk_{serv}^j,S_{serv}^j,VT_{j,y}^X\right)$ 存储在跨域联盟链上.

值得注意的是, AU^X 和 D_j^B 通常会在本地离线生成多组部分服务公私钥, 以便收到加入服务列表请求时可以快速做出响应, 以此来降低系统计算延迟. 设备 D_i^A 在初始加入该服务时同样需要执行分布式服务密钥生成流程. 与上述流程不同的是, D_i^A 需要通过 ES^A 向其他实体转发请求.

4)服务信息产生

数据使用者 D_j^B 通过 ES^B 向 D_i^A 发起服务信息查看请求. ES^B 在收到请求后, 通过 BDN^B 验证 $PID_{j,y}^B$ 的合法性, 包括 $PID_{j,y}^B$ 是否在服务列表内和 $PID_{j,y}^B$ 是否在有效期内, 验证 $PID_{j,y}^B$ 为合法伪身份后, ES^B 将请求重新签名并转发给 D_i^A. D_i^A 在收到请求后, 为保证服务信息在传输过程中的安全且不被篡改, D_i^A 按照以下步骤执行服务信息 m 的加密与签名:

D_i^A 首先选择一个随机数 $k_1\in Z_q^*$, 之后计算 $sri=H_2\left(PID_{i,x}^A,k_1,T\right)$, $SRI=sri\cdot P$, 其中 T 为当前时间戳. D_i^A 再计算

$$V_i^A=sk_{i,x}^A\cdot s_{serv}^i,R_i^A=sk_{i,x}^A\cdot S_{serv}^i,$$

$$U_i^A=H_3\left(m,S_{serv}^i\right),$$

$$W_i^A=R_i^A+U_i^A\cdot sk_{i,x}^A\cdot pk_{serv}^i.$$

最后, D_i^A 通过 ES^A 将 $\partial_i^A=\left(c,SRI,PID_{i,x}^A,T\right)$ 发送给 D_j^B. 其中密文 $c=e\left(pk_{serv}^j,sri\cdot pk_{j,y}^B\right)\cdot\left(m\|\sigma_i^A\right)$, 签名 $\sigma_i^A=\left(V_i^A,W_i^A\right)$.

5)服务信息接收

当 ES^A 收到 ∂_i^A 后, ES^A 首先通过 BDN^A 在跨域联盟链上查询 $PID_{i,x}^A$ 是否存在于服务列表内并根据 T 来验证 $PID_{i,x}^A$ 是否在有效期. 如果 $PID_{i,x}^A$ 存在且在有效期内, ES^A 通过 BDN^A 在跨域联盟链上获取到 S_{serv}^i 后将 $\partial_i^{A'}=\left(c,SRI,PID_{i,x}^A,T,S_{serv}^i\right)$ 发送给 D_j^B. D_j^B 收到 $\partial_i^{A'}$ 后, 根据如下步骤来解密并验证签名的有效性:

D_j^B 计算 $m'\|\sigma_i^{A'}=\dfrac{c}{e\left(sk_{serv}^j\cdot P,sk_{j,y}^B\cdot SRI\right)}$ 解密以得到服务信息 m' 和签名 $\sigma_i^{A'}$. 最后, D_j^B 需要计算出 $U_i^{A'}=H_3(m',S_{serv}^i)$ 并通过如下等式来验证签名 $\sigma_i^{A'}=\left(V_i^{A'},W_i^{A'}\right)$ 的正确性.

$$e\left(W_i^{A'},P\right)\overset{?}{=}e\left(V_i^{A'}+U_i^{A'}\cdot pk_{i,x}^A,pk_{serv}^i\right).$$

如果等式不成立，D_j^B拒绝接收该消息；否则，接收该消息，完成服务信息跨域数据共享.

6）身份撤销

当发现系统中有设备发起恶意行为时，实体可以报告给AU^X. AU^X可以根据伪身份和真实身份的对应关系找到疑似恶意的设备并根据报告的信息来监控设备行为. 当发现确有恶意行为时，会通过BDN^X更新跨域联盟链上的动态累加器和服务设备信息列表来撤销恶意设备的信息. 同时，BDN^X会定期更新动态累加器和服务设备信息列表来撤销到期的设备伪身份.

5 安全性证明与分析

5.1 安全性证明

定理 1. 如果 CDH 困难问题假设成立，则本文方案的签名算法在随机预言模型下对 PPT 敌手$\mathcal{A}$满足选择消息攻击下的存在不可伪造性.

证明. 假设一个 PPT 敌手$\mathcal{A}$能够以不可忽略的概率ε攻破签名的不可伪造安全，那么可以构造一个概率多项式时间的模拟器$\mathcal{B}$以不可忽略的概率解决 CDH 困难问题.

1）*Setup*. 在收到$\mathcal{A}$的初始化查询后，模拟器$\mathcal{B}$将公共参数$(G,G_T,P,mpk,H,H_1,H_2,H_3)$发送给$\mathcal{A}$.

2）H查询. $\mathcal{B}$预设置一个表$List_H$，$List_H$初始为空. 当$\mathcal{A}$使用DS_i发起H查询时，其中DS_i为二进制字符串，$\mathcal{B}$查询DS_i在$List_H$中是否有对应值，如果没有，$\mathcal{B}$选择随机数$h\in Z_q^*$，其中$h\notin List_H$. 之后$\mathcal{B}$设置$List_H(DS_i)=h$并将h发送给$\mathcal{A}$. 否则，$\mathcal{B}$查询到对应值并发送给$\mathcal{A}$.

3）H_1查询. $\mathcal{B}$预设置一个表$List_{H_1}$，$List_{H_1}$初始为空. 当$\mathcal{A}$使用Inf_{serv_α}发起H_1查询时，其中Inf_{serv_α}为二进制字符串，$\mathcal{B}$查询Inf_{serv_α}在$List_{H_1}$中是否有对应值，如果没有，$\mathcal{B}$选择随机数$h_1\in Z_q^*$并计算$\eta=h_1\cdot P$，其中$\left(S^i_{serv_\alpha},\eta\right)\notin List_{H_1}$. 之后，模拟器$\mathcal{B}$需要设置$List_{H_1}\left(Inf_{serv_\alpha}\right)=\left(S^i_{serv_\alpha},\eta\right)$并将$\left(S^i_{serv_\alpha},\eta\right)$发送给$\mathcal{A}$. 否则，$\mathcal{B}$查询到对应值并发送给$\mathcal{A}$.

4）H_2查询. $\mathcal{B}$预设置一个表$List_{H_2}$，$List_{H_2}$初始为空. 当$\mathcal{A}$使用$\left(PID^A_{i,x},k_1,T\right)$发起$H_2$查询时，$\mathcal{B}$查询$\left(PID^A_{i,x},k_1,T\right)$在$List_{H_2}$中是否有对应值，如果没有，$\mathcal{B}$选择随机数$h_2\in Z_q^*$，其中$h_2\notin List_{H_2}$. 之后$\mathcal{B}$设置$List_{H_2}\left(PID^A_{i,x},k_1,T\right)=h_2$并将$h_2$发送给$\mathcal{A}$. 否则，$\mathcal{B}$查询到对应值并发送给$\mathcal{A}$.

5）H_3查询. $\mathcal{B}$预设置一个表$List_{H_3}$，$List_{H_3}$初始为空. 当$\mathcal{A}$使用$\left(m,S^i_{serv_\alpha}\right)$发起$H_3$查询时，$\mathcal{B}$查询$\left(m,S^i_{serv_\alpha}\right)$在$List_{H_3}$中是否有对应值，如果没有，$\mathcal{B}$选择随机数$h_3\in Z_q^*$，其中$h_3\notin List_{H_3}$. 之后$\mathcal{B}$设置$List_{H_3}\left(m,S^i_{serv_\alpha}\right)=h_3$并将$h_3$发送给$\mathcal{A}$. 否则，$\mathcal{B}$查询到对应值并发送给$\mathcal{A}$.

6）签名密钥查询. $\mathcal{B}$预设置一个表$List_{\mathrm{SK}}$，$List_{\mathrm{SK}}$初始为空. 当$\mathcal{A}$使用$PID^A_{i,x}$发起签名密钥查询时，$\mathcal{B}$查询$PID^A_{i,x}$在$List_{\mathrm{SK}}$中是否有对应值，如果有，$\mathcal{B}$将对应值发送给$\mathcal{A}$. 否则，$\mathcal{B}$选择随机数$c_{i,x}\in\{0,1\}$且其中$c_{i,x}=1$的概率为β，其中$\beta=\dfrac{1}{q_{\mathrm{SK}}+q_{\mathrm{SSK}}+1}$，$q_{\mathrm{SK}}$为执行签名查询的最大次数，$q_{\mathrm{SSK}}$为执行签名密钥查询的最大次数. 如果$c_{i,x}=0$，$\mathcal{B}$选择随机数$sk^A_{i,x}\in Z_q^*$，其中$sk^A_{i,x}\notin List_{\mathrm{SK}}$. 最后，$\mathcal{B}$设置$List_{\mathrm{SK}}\left(PID^A_{i,x}\right)=sk^A_{i,x}$并将$sk^A_{i,x}$返回给$\mathcal{A}$. 相反，如果$c_{i,x}=1$，$\mathcal{B}$终止该过程并返回$\perp$.

7）验证公钥查询. $\mathcal{B}$预设置一个表$List_{\mathrm{VK}}$，$List_{\mathrm{VK}}$初始为空. 当$\mathcal{A}$使用$\left(serv_\alpha,PID^A_{i,x}\right)$发起验证密钥查询时，$\mathcal{B}$查询$\left(serv_\alpha,PID^A_{i,x}\right)$在$List_{\mathrm{VK}}$中是否有对应值，如果有，$\mathcal{B}$将对应值发送给$\mathcal{A}$. 如果没有且$c_{i,x}=0$，$\mathcal{B}$选择随机数$sk_{serv_\alpha}\in Z_q^*$并从$List_{\mathrm{SK}}$中使用$PID^A_{i,x}$查询到$sk^A_{i,x}$，$\mathcal{B}$计算$pk_{serv_\alpha}=sk_{serv_\alpha}\cdot P$，$pk^A_{i,x}=sk^A_{i,x}\cdot P$，其中$\left(pk_{serv_\alpha},pk^A_{i,x}\right)\notin List_{\mathrm{VK}}$. 最后，模拟器$\mathcal{B}$需要设置$List_{\mathrm{SK}}\left(serv_\alpha,PID^A_{i,x}\right)=\left(pk_{serv_\alpha},pk^A_{i,x}\right)$，并同时需要将$\left(pk_{serv_\alpha},pk^A_{i,x}\right)$返回给$\mathcal{A}$. 相反，如果$c_{i,x}=1$，$\mathcal{B}$选择一个随机数$s\in Z_q^*$并计算$pk^A_{i,x}=s\cdot P$，其中$\left(pk_{serv_\alpha},pk^A_{i,x}\right)\notin List_{\mathrm{VK}}$. 之后，$\mathcal{B}$需要设置$List_{\mathrm{SK}}\left(serv_\alpha,PID^A_{i,x}\right)=\left(pk_{serv_\alpha},pk^A_{i,x}\right)$，并同时需要将$\left(pk_{serv_\alpha},pk^A_{i,x}\right)$返回给$\mathcal{A}$.

8）签名查询. $\mathcal{A}$将m发送给模拟器$\mathcal{B}$，如果$c_{i,x}=1$，$\mathcal{B}$终止该过程. 否则，$\mathcal{B}$执行H_1查询和签名密钥查询，$\mathcal{B}$选择随机数$a_1,a_2\in Z_q^*$并计算$S^i_{serv_\alpha}=a_1\cdot\eta$，$V^A_i=a_2\cdot\eta$. 之后，$\mathcal{B}$执行$H_3$查询计算出$U^A_i=H_3(m,S^i_{serv_\alpha})$. 最后，$\mathcal{B}$选择随机数$w\in Z_q^*$，计算$W^A_i=w\cdot P$并将$(V^A_i,W^A_i,S^i_{serv_\alpha})$返回给$\mathcal{A}$.

9）验证公钥替换查询. 敌手$\mathcal{A}$可以任意选择新的验证公钥(pk_1,pk_2)来替换$PID^A_{i,x}$原本对应的验证公钥$\left(pk_{serv_\alpha},pk^A_{i,x}\right)$.

如果$c_{i,x}=1$时，$\mathcal{A}$伪造了一个有效签名$\sigma^A_i=\left(V^A_i,W^A_i\right)$，该签名满足式(1).

$$e\left(W^A_i,P\right)=e\left(V^A_i+U^A_i\cdot pk^A_{i,x},pk^i_{serv}\right). \tag{1}$$

根据签名的分叉引理[31-32]，$\mathcal{A}$能够以一个不可忽略的概率伪造另一个有效签名$\sigma^{A'}_i=\left(D^{A'}_i,W^{A'}_i\right)$，该签名满足式(2).

$$e\left(W^{A'}_i,P\right)=e\left(V^{A'}_i+U^{A'}_i\cdot pk^A_{i,x},pk^i_{serv}\right). \tag{2}$$

根据式(1)(2)，可以得到

$$e\left(W_i^A - W_i^{A'}, P\right) = e\left(\left(U_i^A - U_i^{A'}\right)\cdot\xi_1\cdot\xi_2\cdot P, P\right). \quad (3)$$

根据式(3)，$\mathcal{A}$只有在能够输出$\left(W_i^A - W_i^{A'}\right)$和$\left(U_i^A - U_i^{A'}\right)$的情况下才能解决CDH困难问题. 而$\mathcal{A}$能够输出$\left(W_i^A - W_i^{A'}\right)$和$\left(U_i^A - U_i^{A'}\right)$需要满足以下条件：

①$\mathcal{B}$的在查询阶段中未终止过程.

②$\mathcal{B}$的在伪造阶段中未终止过程.

③$\mathcal{A}$没有执行签名密钥查询.

条件①的概率为$(1-\beta)^{q_{\mathrm{SK}}+q_{\mathrm{SSK}}}$，条件②的概率为$\beta$，条件③的概率为$1-\frac{q_{\mathrm{SSK}}}{2^\kappa}$. 当执行签名密钥查询和签名查询的次数足够多时，满足上述的总体概率趋近于$\frac{\left(2^\kappa - q_{\mathrm{SSK}}\right)}{2^\kappa\cdot e\left(q_{\mathrm{SK}}+q_{\mathrm{SSK}}\right)}\varepsilon$.

综上所述，$\mathcal{A}$以不可忽略的概率解决了CDH困难问题. 因此，方案的签名算法满足选择消息攻击下的存在不可伪造性. 证毕.

5.2 安全性分析

1)机密性

D_i^A作为数据拥有者为D_j^B生成的服务信息密文为$c = e(pk_{serv}^j, sri\cdot pk_{j,y}^B)\cdot(m\|\sigma_i^A)$，其中$sri$为$D_i^A$选择的随机数. 设备$D_j^B$仅能够获得$SRI = sri\cdot P$. 除$D_j^B$外其他实体需要在给定$SRI$和$P$的前提下计算出$sri$，即解决了离散对数困难问题，才能从密文中获取到服务消息m和签名σ_i^A. 因此只要离散对数困难问题假设成立，除D_j^B外其他实体就无法获取到服务信息，从而证明了该方案满足机密性.

2)不可链接性

AU^X为D_i^X生成的一系列拥有不同有效期的伪身份$PID_{i,j}^X = H\left(ID_i^X\|\varepsilon_{i,j}^X\|T_j\right)$，其中$\varepsilon_{i,j}^X$为随机且不可验证的，因此攻击者无法确定2条消息请求来自同一个设备.

3)匿名性

AU^X为D_i^X生成伪身份$PID_{i,j}^X$并保存在动态累加器和服务设备列表中. 由于$\varepsilon_{i,j}^X$为仅AU^X持有的随机数，且在生成伪身份时会验证伪身份的唯一性，所有其他实体无法从$PID_{i,j}^X$获取到设备真实身份ID_i^X，因此该方案满足匿名性.

4)设备端保密

系统中边缘服务器作为半可信实体，在转发D_i^A的消息过程中由于无法获取到解密密钥对$\left(sk_{serv}^j, sk_{j,y}^B\right)$，因而无法从$D_i^A$生成的服务信息密文$c = e\left(pk_{serv}^j, sri\cdot pk_{j,y}^B\right)\cdot\left(m\|\sigma_i^A\right)$中获取服务消息$m$和签名$\sigma_i^A$，从而实现了方案的设备端保密.

5)基于服务的访问控制

设备在数据共享前，需要发起加入服务列表请求并执行分布式服务密钥生成过程. 服务列表需存储在跨域联盟链中，在查询到D_i^A的伪身份包含在服务列表内，边缘服务器才会协助转发共享消息，因此实现了基于服务的访问控制.

6 性能分析

本节主要介绍了本文方案与文献[29−30]在各阶段计算开销、通信开销和区块链并发查询的性能对比.

本文方案与文献[29−30]的理论时间开销对比如表1、表2所示. 阶段1分别代表文献[29]方案中的会话密钥生成与消息交换阶段；文献[30]方案中的消息加密、代理重加密与消息解密阶段以及本文方案中的分布式密钥生成阶段与服务信息加解密阶段. 阶段2分别代表文献[29]的零知识证明生成与验证阶段以及文献[30]和本文方案中的签名生成与验证阶段. 为更清晰地呈现性能比对结果，表中未纳入对系统计算开销影响较小的运算，具体包括：群G上的小因子乘法运算、哈希运算、哈希值映射到点运算和群G上的乘法运算. 从表1和表2中可以看出，随着消息数量的增多，本文方案中时间消耗较大的双线性配对运算C_{bp}和标量乘法运算C_{sm_1}总体数量相对文献[29−30]中时间消耗较大的双线性配对运算C_{bp}，标量乘法运算C_{sm_1}、C_{sm_2}以及指数运算C_{e}较少，且随着消息数量的增多，计算效率上的优势会越来越明显.

实验使用Java语言和JPBC(Java pairing-based cryptography)库进行代码编写，在32 GB内存、12th Gen

Table 1 Comparison of Computation Cost in Stage 1

表1 阶段1计算开销对比

方案	d_i^A	ES^A	ES^B	d_j^B
文献[29]方案	$3nC_{sm_1}+C_{sm_2}+2nC_{\mathrm{e}}+2nC_{bp}$	C_{sm_2}	C_{sm_2}	$3nC_{sm_1}+(n+1)C_{sm_2}+nC_{\mathrm{e}}+2nC_{bp}$
文献[30]方案		$2nC_{sm_1}+nC_{\mathrm{e}}$	$2nC_{sm_1}+2nC_{bp}$	nC_{bp}
本文方案	$(n+4)C_{sm_1}+nC_{bp}$			$(n+4)C_{sm_1}+nC_{bp}$

注：n代表消息数量，C_{sm_1}和C_{sm_2}分别为群G和G_2上的标量乘法运算，C_{e}为群G上的指数运算，C_{bp}为双线性配对运算.

Table 2 Comparison of Computation Cost in Stage 2

表 2 阶段 2 计算开销对比

方案	d_i^A	ES^A	ES^B	d_j^B
文献 [29] 方案	$4nC_{sm_1}+C_{sm_2}+2nC_{bp}$	C_{sm_2}	C_{sm_2}	$C_{sm_1}+C_{sm_2}+5nC_{sm_1}$
文献 [30] 方案			$2nC_{sm_1}+3nC_e$	$2nC_e$
本文方案	$2nC_{sm_1}$			$2nC_{bp}$

注：n 代表消息数量，C_{sm_1} 和 C_{sm_2} 分别为群 G 和 G_2 上的标量乘法运算，C_e 为群 G 上的指数运算，C_{bp} 为双线性配对运算.

Intel® Core™ i5-12600KF CPU、Windows11 操作系统环境下运行，软件环境为 IntelliJ IDEA 2023.3.8，使用 TypeA 型椭圆曲线，群的阶数为 160 b 长度的素数，哈希函数为 SHA 256.

为了确保结果的可靠性，本次实验中对每项数据进行了 100 次测试取平均值.

本文方案与文献 [29−30] 的各阶段及总体计算开销对比如图 3~5 所示. 从图 3~5 中可以看出，随着消息数量的增多，本文方案在各阶段的计算效率优势越来越大且始终优于文献 [29−30].

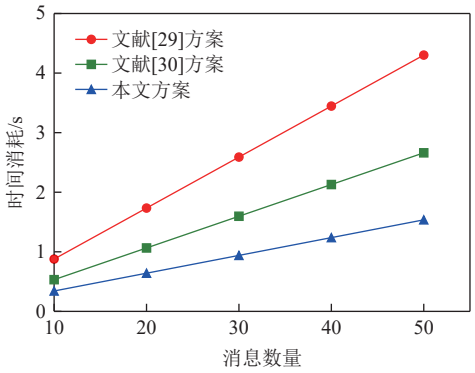


Fig. 3 Performance comparison of computation cost in stage 1

图 3 阶段 1 计算开销性能对比

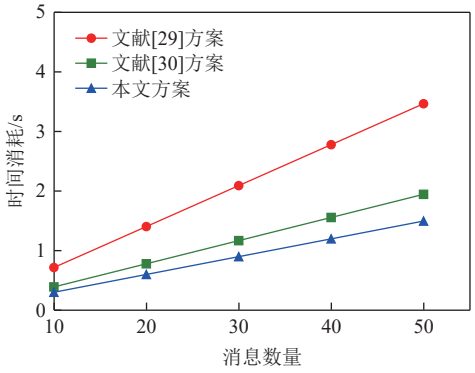


Fig. 4 Performance comparison of computation cost in stage 2

图 4 阶段 2 计算开销性能对比

本文方案与文献 [29−30] 的总体通信开销对比如图 6 所示. 根据实验设置的一系列参数，在保证各

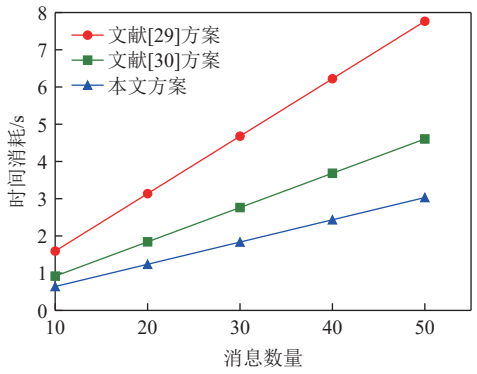


Fig. 5 Performance comparison of total computation cost

图 5 总体计算开销性能对比

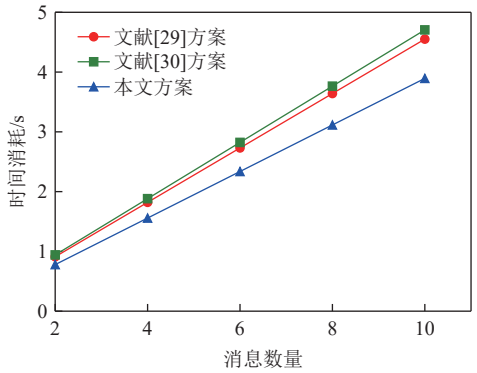


Fig. 6 Performance comparison of communication cost

图 6 通信开销性能对比

方案中的消息长度均为 20 B 的情况下，文献 [29] 的总通信带宽约为 3 034 B，文献 [30] 的总通信带宽约为 3 137 B，本文方案的总通信带宽约为 2 596 B. 从图 6 中可以看出，本文方案在各消息数量下的总体通信延迟均低于文献 [29−30].

为了测试链上操作的性能，实验中部署了 6 台 ubuntu-22.04.3 虚拟机作为 6 个区块链域节点并同时在每个测试域内部署了 20 个仿真设备. 在此节点基础了，我们搭建了 FISCO BCOS 2.0 联盟链，以此来进行链上操作的测试. 图 7 展示了文献 [29−30] 和本文

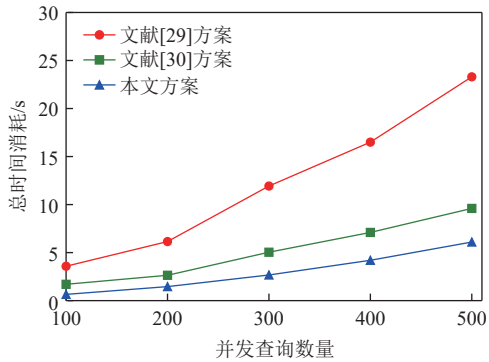


Fig. 7 Performance comparison of blockchain query

图 7 区块链查询性能对比

方案在并发查询数量不断增加下的总时间消耗. 其中文献 [30] 与本文方案的查询方式均为本地节点查询, 但文献 [30] 中所需查询条目较本文方案更为复杂, 结合实验结果可以看出本文方案的区块链查询效率更高且满足高并发性.

图 3~7 表明, 本文方案在各阶段计算效率、通信效率和区块链查询上均优于文献 [29−30]. 且计算效率和通信效率随着消息数量的增加性能优势尤其明显.

综上所述, 尽管本文方案通信延迟较对比方案提升有限, 但各阶段的计算效率和区块链并发查询效率有显著优势. 因此, 本文方案总体效率优于对比方案.

7 总 结

为了实现跨域环境下 IIoT 设备轻量级的消息认证与共享, 本文提出了一种基于区块链的轻量级 IIoT 跨域认证与数据共享方案, 该方案结合分布式密钥生成技术, 实现了在 IIoT 服务场景下设备端保密的匿名消息认证与共享. 本文详细介绍了所提出方案的具体构造, 并提供了安全性分析, 阐述了方案的安全性. 最后通过计算开销、通信开销和区块链查询的实验对比证明了该方案有着较高的效率, 满足轻量级特性.

作者贡献声明: 李哲成提出了初步方案, 进行了实验仿真、论文撰写和修改; 张波负责理论指导和方案分析.

参 考 文 献

[1] Bao Han, Zhao Youjian, Zhang Xiaoping, et al. A probabilistic and distributed validation framework based on blockchain for artificial intelligence of things[J]. IEEE Internet of Things Journal, 2024, 11(1): 17−28

[2] Yang Wei, He Jie, Wan Yadong, et al. Security countermeasures for time synchronization in IEEE802.15. 4e-based industrial IoT[J]. Journal of Computer Research and Development, 2017, 54(9): 2032−2043 (In Chinese)
(杨伟, 何杰, 万亚东, 等. 基于 IEEE802.15. 4e 标准的工业物联网安全时间同步策略[J]. 计算机研究与发展, 2017, 54(9): 2032−2043)

[3] Karati A, Islam S H, Biswas G P, et al. Provably secure identity-based signcryption scheme for crowdsourced industrial Internet of things environments[J]. IEEE Internet of Things Journal, 2018, 5(4): 2904−2914

[4] Yang Ting, Zhang Jiayuan, Huang Zaiqi, et al. Survey of industrial control systems security[J]. Journal of Computer Research and Development, 2022, 59(5): 1035−1053 (in Chinese)
(杨婷, 张嘉元, 黄在起, 等. 工业控制系统安全综述[J]. 计算机研究与发展, 2022, 59(5): 1035−1053)

[5] Yang Wenjie, Wang Shangpeng, Huang Xinyi, et al. On the security of an efficient and robust certificateless signature scheme for IIoT environments[J]. IEEE Access, 2019, 7: 91074−91079

[6] Liu Peng, Xian Youquan, Yao Chuanjian, et al. A trustworthy and consistent blockchain oracle scheme for industrial Internet of things[J]. IEEE Transactions on Network and Service Management, 2024, 21(5): 5135−5148

[7] Wang Hangyu, Lü Fei, Cheng Yuliang, et al. A review of zero trust security research in industrial Internet of things[J/OL]. Journal of Computer Research and Development. [2025-07-24]. http://kns.cnki.net/kcms/detail/11.1777.tp.20250407.1652.018.html (in Chinese)
(王航宇, 吕飞, 程裕亮, 等. 工业物联网零信任安全研究综述[J/OL]. 计算机研究与发展. [2025-07-24]. http://kns.cnki.net/kcms/detail/11.1777.tp.20250407.1652.018.html)

[8] Shim K A. A secure certificateless signature scheme for cloud-assisted industrial IoT[J]. IEEE Transactions on Industrial Informatics, 2024, 20(4): 6834−6843

[9] Zhang Shijie, Lee J H. A group signature and authentication scheme for blockchain-based mobile-edge computing[J]. IEEE Internet of Things Journal, 2020, 7(5): 4557−4565

[10] Schnorr C P. Efficient identification and signatures for smart cards[C]//Proc of Conf on the Theory and Application of Cryptology. Berlin: Springer, 1989: 239−252.

[11] Ateniese G, Camenisch J, Joye M, et al. A practical and provably secure coalition-resistant group signature scheme[C]//Advances in Cryptology−CRYPTO 2000: 20th Annual Int Cryptology Conf. Berlin: Springer, 2000: 255−270

[12] Burrows M, Abadi M, Needham R. A logic of authentication[J]. ACM Transactions on Computer Systems, 1990, 8(1): 18−36

[13] Boneh D, Boyen X, Shacham H. Short group signatures[C]//Proc of Annual Int Cryptology Conf. Berlin: Springer, 2004: 41−55

[14] Karati A, Islam S H, Karuppiah M. Provably secure and lightweight certificateless signature scheme for IIoT environments[J]. IEEE Transactions on Industrial Informatics, 2018, 14(8): 3701−3711

[15] Cui Jie, Wang Fengqun, Zhang Qingyang, et al. Anonymous message authentication scheme for semitrusted edge-enabled IIoT[J]. IEEE Transactions on Industrial Electronics, 2021, 68(12): 12921−12929

[16] Yang Zheng, He Jun, Tian Yangguang, et al. Faster authenticated key agreement with perfect forward secrecy for industrial Internet-of-things[J]. IEEE Transactions on Industrial Informatics, 2020, 16(10): 6584−6596

[17] Xiong Hu, Wu Yan, Jin Chuanjie, et al. Efficient and privacy-preserving authentication protocol for heterogeneous systems in IIoT[J]. IEEE Internet of Things Journal, 2020, 7(12): 11713−11724

[18] Hou Yingzhe, Xiong Hu, Huang Xin, et al. Certificate-based parallel key-insulated aggregate signature against fully chosen key attacks for industrial Internet of things[J]. IEEE Internet of Things Journal, 2021,

8(11): 8935−8948

[19] Xu Dongyang, Yu Keping, Ritcey J A. Cross-layer device authentication with quantum encryption for 5G enabled IIoT in industry 4.0[J]. IEEE Transactions on Industrial Informatics, 2022, 18(9): 6368−6378

[20] Cui Jie, Wang Fengqun, Zhang Qingyang, et al. Efficient batch authentication scheme based on edge computing in IIoT[J]. IEEE Transactions on Network and Service Management, 2023, 20(1): 357−368

[21] Zhang Yinghui, He Jiangyong, Guo Rui, et al. Server-aided and verifiable attribute-based signature for industrial Internet of things[J]. Journal of Computer Research and Development, 2020, 57(10): 2177−2187 (in Chinese)
(张应辉, 贺江勇, 郭瑞, 等. 工业物联网中服务器辅助且可验证的属性基签名方案[J]. 计算机研究与发展, 2020, 57(10): 2177−2187)

[22] Tan Yawen, Wang Jiadai, Liu Jiajia, et al. Blockchain-assisted distributed and lightweight authentication service for industrial unmanned aerial vehicles[J]. IEEE Internet of Things Journal, 2022, 9(18): 16928−16940

[23] Wang Weizheng, Xu Hao, Alazab M, et al. Blockchain-based reliable and efficient certificateless signature for IIoT devices[J]. IEEE Transactions on Industrial Informatics, 2022, 18(10): 7059−7067

[24] Yu Keping, Tan Liang, Aloqaily M, et al. Blockchain-enhanced data sharing with traceable and direct revocation in IIoT[J]. IEEE Transactions on Industrial Informatics, 2021, 17(11): 7669−7678

[25] Shen Meng, Liu Huisen, Zhu Liehuang, et al. Blockchain-assisted secure device authentication for cross-domain industrial IoT[J]. IEEE Journal on Selected Areas in Communications, 2020, 38(5): 942−954

[26] Singh P, Masud M, Hossain M S, et al. Cross-domain secure data sharing using blockchain for industrial IoT[J]. Journal of Parallel and Distributed Computing, 2021, 156: 176−184

[27] Zhang Yan, Li Bing, Wu Jiaxin, et al. Efficient and privacy-preserving blockchain-based multifactor device authentication protocol for cross-domain IIoT[J]. IEEE Internet of Things Journal, 2022, 9(22): 22501−22515

[28] Wang Fengqun, Cui Jie, Zhang Qingyang, et al. Blockchain-based lightweight message authentication for edge-assisted cross-domain industrial internet of things[J]. IEEE Transactions on Dependable and Secure Computing, 2024, 21(4): 1587−1604

[29] Zeng Shulei, Cao Bin, Sun Yao, et al. Blockchain-assisted cross-domain data sharing in industrial IoT[J]. IEEE Internet of Things Journal, 2024, 11(16): 26778−26792

[30] Wang Fengqun, Cui Jie, Zhang Qingyang, et al. Blockchain-based secure cross-domain data sharing for edge-assisted industrial Internet of things[J]. IEEE Transactions on Information Forensics and Security, 2024, 19: 3892−3905

[31] Pointcheval D, Stern J. Security arguments for digital signatures and blind signatures[J]. Journal of Cryptology, 2000, 13(3): 361−396

[32] Zhou Yanwei, Wang Zhaolong, Qiao Zirui, et al. An efficient and provably secure identity authentication scheme for VANET[J]. IEEE Internet of Things Journal, 2023, 10(19): 17170−17183



Li Zhecheng, born in 2001. Master candidate. His main research interests include cross-domain authentication and data sharing.
李哲成, 2001 年生. 硕士研究生. 主要研究方向为跨域认证、数据共享.



Zhang Bo, born in 1981. PhD, associate professor. Senior member of CCF. His main research interests include cryptography and information security.
张　波, 1981 年生. 博士, 副教授. CCF 高级会员. 主要研究方向为密码学、信息安全.