计算机科学
COMPUTER SCIENCE

基于智能合约的流数据授权撤销方案研究

门蕊蕊，贾洪勇，都金如

引用本文

门蕊蕊，贾洪勇，都金如. 基于智能合约的流数据授权撤销方案研究[J]. 计算机科学，2024，51(10): 372-379.

MEN Ruirui, JIA Hongyong, DU Jinru. Study on Stream Data Authorization Revocation Scheme Based on Smart Contracts [J]. Computer Science, 2024, 51(10): 372-379.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.230700094

基于智能合约的流数据授权撤销方案研究

门蕊蕊　贾洪勇　都金如
郑州大学网络空间安全学院　郑州 450000
(mrr0623@163.com)

摘　要　物联网设备和服务将实时生成的流数据加密后进行外包存储,并通过访问控制对用户进行授权,当用户的身份或权限发生变更时,需要撤销用户的权限。现有撤销方案通常存在密钥频繁更新和重加密密文的问题,导致撤销效率低下,灵活度不足,难以实现实时撤销,面临数据泄露风险。为解决流数据外包存储场景下的实时授权撤销问题,提出了一种基于智能合约的去中心化授权撤销方案。在边缘计算和区块链相结合的物联网架构下,将流数据按照时间间隔分块,使用 HASH 树生成与块对应的大量且唯一的密钥,并对分块数据进行对称加密;树节点创建访问令牌并通过代理重加密技术进行共享,实现了可更改的访问策略和高效动态数据共享;利用智能合约技术创建访问控制列表和不当行为列表,对用户权限进行定时撤销和即时撤销操作,实现了去中心化的实时授权撤销。安全性分析和仿真实验证明所提方案与其他相关的研究方案相比,提供了更好的安全性、功能、通信和计算成本,更具有效性。

关键词:流数据;边缘计算;区块链;智能合约;访问控制;授权撤销
中图分类号　TP309

Study on Stream Data Authorization Revocation Scheme Based on Smart Contracts

MEN Ruirui,JIA Hongyong and DU Jinru
School of Cyber Science and Engineering,ZhengZhou University,Zhengzhou 450000,China

Abstract　IoT devices and services encrypt real-time generated stream data for outsourced storage,and authorize users through access control. When the user's identity or permissions change,authorization to the user needs to be revoked. Existing revocation schemes have problems of frequent key updates and re-encrypted ciphertext,resulting in low revocation efficiency,insufficient flexibility,difficulty in achieving real-time revocation,and the risk of data leakage. In order to solve the real-time authorization revocation in the outsourcing storage scenario of streaming data,a decentralized authorization revocation scheme based on smart contracts is proposed. Under the IoT architecture combined with edge computing and blockchain,the streaming data is divided into blocks according to time intervals,and a large number of unique keys corresponding to the blocks are generated using the HASH tree,and the partitioned data is symmetrically encrypted. The tree nodes create access tokens and share them through proxy re-encryption technology,to implement modifiable access policies and efficient dynamic data sharing. By utilizing smart contract technology to create access control lists and misconduct lists,users are subjected to scheduled and immediate revocation operations,achieving decentralized real-time authorization revocation. Through security analysis and simulation experiments,it has been proven that this scheme provides better security,functionality,communication,and computing costs compared to other related research schemes,and is more effective.

Keywords　Stream data,Edge computing,Blockchain,Smart contracts,Access control,Authorization revocation

1　引言

物联网流数据[1]指由物联网设备和服务产生的连续、实时的数据流,并通过传感器、控制器等组件将采集的数据实时传输到存储设备。在一些特定应用场景中,应缩短流数据的处理时延,如股票价值、信用卡交易、交通状况和时间敏感的患者数据等流数据,如果不能立即处理,其价值会快速降低。边缘计算[2]可以快速完成流数据的计算、存储、缓存等任务,为当前物联网速度受限的问题提供了新的解决思路。

如今随着物联网的发展,数据的采集、处理和存储变得

到稿日期:2023-07-13　返修日期:2024-01-15
基金项目:河南省重大科技专项(221100210900-01);2021 年中国高校产学研创新基金(2021ITA11021)
This work was supported by the Management of Major Science and Technology of Henan Province(221100210900-01) and 2021 China University Industry University Research Innovation Fund(2021ITA11021).
通信作者:贾洪勇(hyjia@zzu.edu.cn)

越来越简便，但是如何在物联网外包场景下安全地共享并保护隐私变得更加重要。之前已经提出相关的加密方案来实现保护隐私[3]，其中数据在用户设备上进行加密，并在存储提供程序上进行加密存储。只有获得授权的请求者可以执行加解密，并且不会向第三方披露任何加密密钥。端到端加密可以增强安全性，同时在共享加密数据的过程中需要对用户进行访问控制。目前采用的共享加密数据的访问控制解决方案[4-8]在流数据的安全存储、访问控制、信息更新和删除方面提供了一些思路，但在恶意用户撤销方面仍然存在许多问题。

针对流数据外包存储场景下权限难以实时撤销的问题，本文提出了一种基于智能合约的授权撤销方案。本文的主要研究工作及成果如下：

1）提出了一种加密方案，实现可更改的访问策略和高效动态数据共享。将流数据按照时间间隔分块，使用 HASH 树生成与块对应的密钥流，对每个块使用唯一的密钥进行加密。树节点创建访问令牌并通过代理重加密技术进行共享。

2）设计了一个基于智能合约的访问控制框架，实现去中心化的实时授权撤销。框架提供注册、更新和删除功能，利用智能合约进行“自主代理”检查用户行为，实现定时撤销和即时撤销。

3）使用 Fabric 进行仿真实验，对所提方案进行了详细的性能测试，对各模块的计算成本、执行效率以及智能合约开销等性能指标进行了分析。实验结果表明，所提方案相比目前现有的授权撤销方案，具有更高的效率和性能优势。

2 相关工作

数据传输到存储设备时存在数据被泄露或者滥用的问题。现有研究在数据上传到存储设备之前对其进行加密，同时通过访问控制对数据请求者授权。Sahai 提出了属性基加密（Attribute-based Encryption，ABE）的思想。ABE 作为一种新的密码原语，实现了细粒度的数据访问控制。ABE 主要包括两类，即密钥策略属性基加密（Key-policy Attribute-based Encryption，KP-ABE）[4] 和密文策略属性基加密（Ciphertext-policy Attribute-based Encryption，CP-ABE）[5]。Wang 等[6] 和 Saidi 等[7] 通过智能合约实现系统的去中心化访问控制功能并进行细粒度的授权。Burkhalter 等[8] 利用区块链进行细粒度的分散授权，授权方只能在其权限范围内解密和验证查询。但是这些访问控制模型更注重如何进行授权，忽视了对权限的撤销。

由于用户身份变化或者权限变更，需要对权限进行撤销，许多可撤销方案被提出。Choksy 等[9] 根据衡量用户可信度的 QF 值来控制权限级别并在属性有效期结束时撤销或按需自动撤销，但灵活度不足。Tao 等[10] 采用了向量承诺和群签名来支持用户撤销，生成群签名的计算成本和审计操作使该方案效率低下。Wu 等[11] 采用群签名来支持用户撤销，通过随机屏蔽技术来提供用户身份隐私和数据隐私。Höglund 等[12] 使用 Bloom 过滤器设计了一个压缩证书撤销列表协议，采用 Merkle 树和哈希链实现撤销操作。上述方案是基于公钥基础设施（PKI）设计的，而流数据的数据量大，对实时性要求较高，因此不适用于流数据的授权撤销。

目前众多学者对流数据授权撤销的研究已经取得了一定进展，并提出了一些可行的方案。Shafagh 等[13] 在文献[8]的基础上提出了 Droplet 方案，在撤销访问期间，必须下载、重新加密数据，并将其上传回云中，权限撤销过程较为复杂。Yang 等[14] 提出了多重访问控制结构，实现了用户撤销、属性撤销和关键字搜索，但采用外包 ABE，系统需要半信任云网络。Zhang 等[15] 引用组管理器来更新未撤销用户的组密钥，并生成重加密密钥实现用户撤销，但通信成本过高。Yu 等[16] 使用撤销列表实现数据的访问控制和恶意用户的跟踪和撤销，但未考虑数据隐私问题。Wiraatmaja 等[17] 和 LI 等[18] 利用智能合约技术实现对流数据的管理，Wiraatmaja 没有实现撤销，LI 实现了撤销但是灵活度不足。这些流数据授权撤销方案无法及时撤销用户的访问权限，难以实现实时撤销。

3 预备知识

3.1 AES-GCM

AES-GCM 算法融合了 AES 加密和 GCM 身份验证算法，提供了机密性和完整性保护。AES-GCM 算法的具体步骤如下：用一个密钥初始化算法，设置运行参数得到密钥 k_i，数据块进行 AES 加密，得到加密结果 C_i，计数器递增产生新的随机值 IV，与加密结果一起利用 GCM 算法产生身份验证标签，生成加密块；如果存在多个数据块需要加密，计数器需要递增，然后重复加密和解密操作，最终得到所有数据块加密后的结果，同时得到身份验证标签，此时加密完成。接收方接收到密文后，以同样方式生成身份验证标签，然后将密文与身份验证标签一起进行解密得到原始明文 M_i。加解密的过程如式(1)和式(2)所示：

$$Enc(k_i, IV, M_i) \rightarrow C_i \tag{1}$$

$$Dec(k_i, IV, C_i) \rightarrow M_i \tag{2}$$

3.2 代理重加密

代理重加密[19]（Proxy Re-encryption，PRE）是一种公钥加密机制，主要功能是将授权人公钥加密的密文转换为另一种密文，且对应明文不变，转换后的密文可以由被授权人的私钥进行解密。假设数据生成者 A 用公钥 pk_A 对一个消息 M 进行加密，从而产生密文 C_A。A 将对消息 M 的访问权限委托给 B，A 通过自己的私钥 sk_A 和 B 的公钥 pk_B 创建一个重新加密的密钥并发送给代理服务器。

$$rk_{A\rightarrow B} = rekey(sk_A, pk_B) \tag{3}$$

代理服务器利用重加密密钥 $rk_{A\rightarrow B}$ 重新加密密文 C_A，使其转换为密文 C_B 并发送给 B。

$$C_B = reencrypt(rk_{A\rightarrow B}, C_A) \tag{4}$$

B 使用私钥 sk_B 解密密文 C_B 获取消息 M。

$$M = decrypt(C_B, sk_B) \tag{5}$$

3.3 智能合约

智能合约[20-22] 存在于分布式、去中心化的区块链网络中。智能合约允许在匿名或不受信任的各方之间进行交易，

而无需中央机构。智能合约封装了一些预定义的状态和转移规则、触发合约执行的场景、特定场景下的响应等。智能合约具有3个特点,即自治、自给自足和去中心化。利用区块链智能合约技术可以很好地对时间、空间上的条件进行控制,进一步提高数据资源的安全性和可控性。

4 方案架构

本文的物联网架构结合了区块链和边缘计算,实现了流数据去中心化的授权撤销[23]。边缘计算系统由众多异构终端、边缘节点服务器和云服务器组成,如图1所示。根据上述各元素在访问控制中的作用,将其分为以下角色:

1)数据所有者:指拥有需要被处理或存储的数据的实体或组织。它们可能是终端设备的用户或组织,负责生成、收集和管理数据。

2)数据请求者:指需要访问和使用数据的实体或应用程序,是终端设备的用户、其他系统或应用程序。

3)边缘服务器:密钥共享时作为代理服务器,通常位于靠近用户的位置,用于存储和处理数据,执行计算任务并减轻云服务器的负载。

4)云服务器:位于云端的服务器,用于存储和处理大规模数据和计算任务。在边缘计算系统中起到支持和扩展边缘节点服务器的作用。

5)区块链网络:包含交易和智能合约的区块链,由边缘网络中所有设备共同维护。

当用户访问边缘节点上的资源时,向区块链发送身份数据并获取相关信息。存储节点接收到存储或检索数据的请求后会查询相应的访问权限。存储节点使用基于签名的身份验证技术来验证请求设备的身份,并检查设备是否具有所需的访问权限以及请求的操作是否与访问权限匹配。如果请求违反了访问权限,则存储节点将拒绝访问数据;如果设备获得了权限,则服务器将会把加密数据和重加密密文发送给设备。

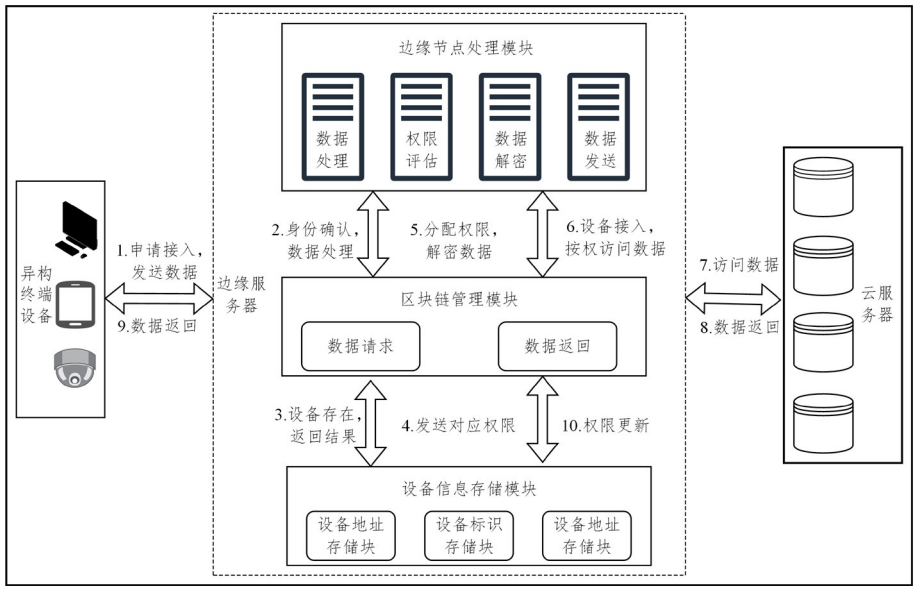

图1 访问控制流程图

Fig. 1 Access control flowchart

5 方案设计

本章详细介绍了流数据授权撤销方案的设计与实现。在数据初始化阶段,数据按照时间间隔分块并对每个数据块进行加密;在密钥共享阶段,树节点创建访问令牌并通过代理重加密技术进行共享;在访问控制阶段,提出了基于智能合约的访问控制框架;在访问权限撤销阶段,实现了定时撤销和即时撤销。

5.1 数据初始化

数据所有者将流数据序列化,以定义的时间间隔分块,并以密码方式链接在一起。流数据使用大量且唯一的随机对称密钥进行加密,随机对称密钥由密钥推导树导出,如图2所示。加密的最小单位是块,这样可以允许数据请求者在块级别上进行授权,与数据请求者共享密钥流中的对应范围来限制对数据流的访问,同时也可以在块级别上对授权过的用户进行撤销。

密钥推导树是基于平衡二叉树构成的 。密钥推导树的根节点由一个秘密且随机的种子形成,这个种子仅数据所有者拥有。树中的每一个子节点都由伪随机生成器PRG(Pseudo Random Generator)生成,PRG是一个不含随机性的函数 $G:\{0,1\}^{\lambda}\rightarrow\{0,1\}^{\lambda+\ell}$,$\ell>0$。假设有一个PRG,$G:\{0,1\}^{\lambda}\rightarrow\{0,1\}^{2\lambda}$,根据完全二叉树构造PRF(Pseudo Random Function),其中函数 $F:\{0,1\}^{\lambda}\times\{0,1\}^{n}\rightarrow\{0,1\}^{\lambda}$。将密钥 k 作为根节点PRG的输入得到 $G(k)$,然后根据输入的第一位是0(1)而保留前(后)λ 位 $G_L(G_R)$,作为下一个节点PRG的输入,后面以此类推构成PRF。密钥推导树叶节点中含有推导函数,字符串经过推导计算后形成密钥流 $\{k_0,k_1,k_2,\cdots,k_{2h-1}\}$,得到随机密钥流后,使用AES-GCM加密算法对数据进行对称加密,生成密文 C_i。通过这种构造,数据请求者可以有效地共享密钥流中的指定范围来限制数据请求者对数据流的访问,指定某位数据请求者可以访问数据流的某个时间段,实现细粒度的访问控制。

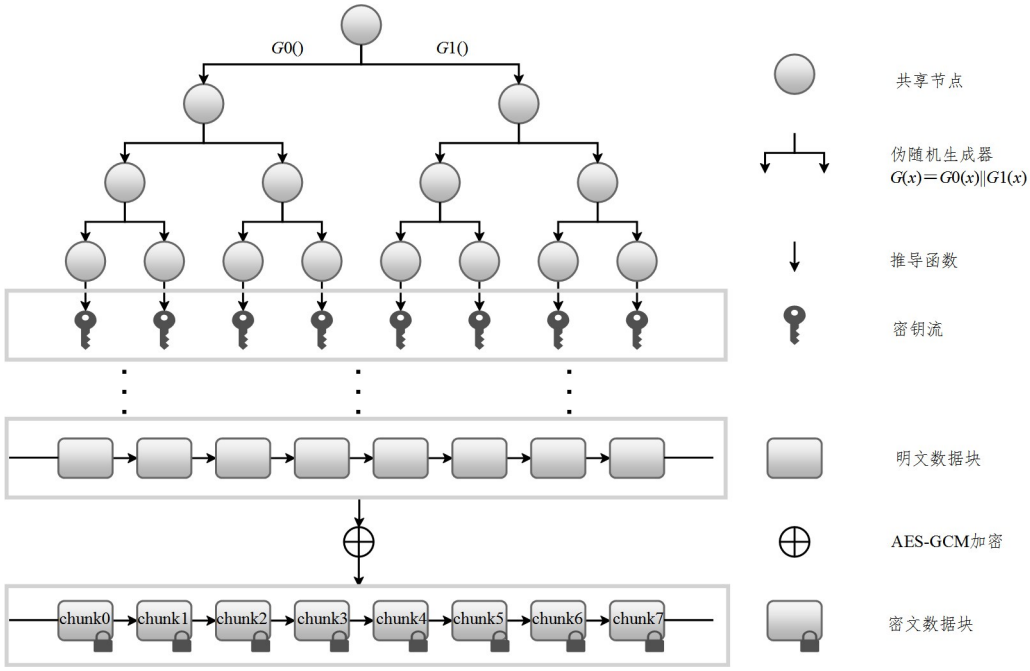

图 2 密钥派生及数据加密图

Fig. 2 Key derivation and data encryption

5.2 密钥共享

数据所有者共享密钥时并不是共享单个密钥，而是共享树内的几个节点，节点组合形成访问令牌 K，树中的单个节点可以用于数千个密钥的共享。访问令牌封装共享所需的树内部节点、子树高度和密钥标识符等信息，并通过代理重加密进行共享，以便数据请求者获取令牌，并计算所需的密钥材料。同时允许加密数据从一个密钥重新加密为另一个密钥，而无需解密数据本身，实现了高效解密和动态更新。访问令牌进行代理重加密的共享过程如图 3 所示。

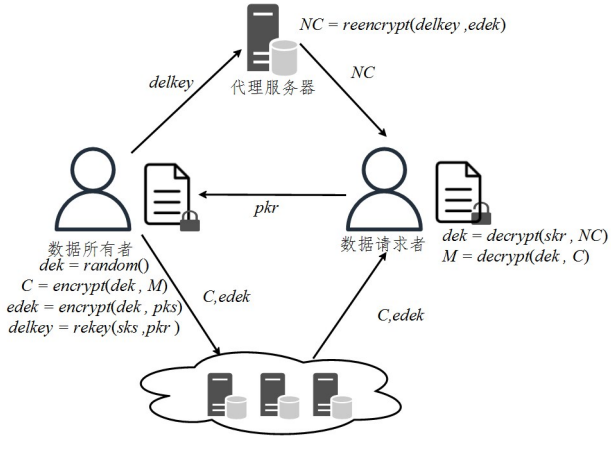

图 3 密钥共享

Fig. 3 Key sharing

密钥共享步骤如下：

1）各参与方通过椭圆曲线算法生成公私钥对，数据所有者生成密钥对 pk_s/sk_s，数据请求者生成密钥对 pk_r/sk_r。

2）数据所有者使用公钥 pk_s 对访问令牌 K 加密 $edek=encrypt(pk_s,K)$，并将原始加密数据 C 和 $edek$ 提供给代理服务器。

3）请求者请求数据时发送 pk_r 到所有者。

4）所有者根据私钥 sk_s 和请求者的公钥 pk_r 生成重加密密钥 $delkey=rekey(sk_s,pk_r)$，并将 $delkey$ 传到代理服务器。

5）代理服务器使用重加密密钥 $delkey$ 对 $edek$ 进行重加密，生成重加密密文 NC，$NC=reencrypt(delkey,edek)$，并将密文 C 和重加密密文 NC 发送给请求者。

6）请求者使用私钥 sk_r 解密 NC，得到访问令牌 $K=decrypt(NC,sk_r)$，通过访问令牌，请求者可以解密 C 得到明文 M。

通过这个过程，代理服务器可以在不获取数据内容的情况下，将加密数据由数据所有者的密钥重新加密为数据请求者的密钥，实现了数据转发的功能，同时保护了数据的隐私和安全。

5.3 基于智能合约的访问控制

为了更好地对用户进行访问控制，本文设计了一个基于智能合约的分布式访问控制框架，该框架包含 4 种不同的合约：注册合约（RC）、访问控制合约（ACC）、不当行为合约（MC）和判断合约（JC）。基于智能合约的访问控制框架，通过检查用户的行为来实现预定义的访问控制策略的静态访问权限验证和动态访问权限验证。

在访问控制过程中，根据数据请求者的操作和发送请求的时间，对请求者进行验证。ACC 和 MC 将消息发送至 JC，JC 进行判断并返回访问结果和处罚。关于智能合约的算法如算法 1 所示，其中 MI 表示两个连续请求之间允许的最小时间间隔。$NoFR$ 表示短时间内频繁请求的数量。ToLR 表示最后一次请求访问的时间。TH 表示 NoFR 上的阈值。如果 $NoFR$ 大于或等于阈值，则 MC 判断发生了不当行为，并将其加入不当行为列表。变量 UT 表示请求被阻止之前的时间，当请求解除阻止时，该时间设置为 0，阻塞时段被量化为 UT 和检测到不当行为时间之差。

算法 1 智能合约访问控制算法

输入：请求者 user，操作 operation，时间 time

```
输出:结果 result,惩罚 penalty
1. / * 智能合约访问控制方案 * /
2. 初始化:p ← RC[user][operation]
3. / * 若时间合规则继续验证,否则拒绝请求 * /
4. if UT≤ time
5. / * 返回 ACC 列表查询结果 ACC. result * /
6.     ACC. result ← ACC. list
7. / * 间隔时间不满足要求,频繁请求数+1 * /
8.     if time-p. ToLR≤ p. MI
9.         p. NoFR ← p. NoFR+1
10. / * 判断频繁请求数是否超过阈值 * /
11.        if p. NoFR ≥ p. TH
12. / * 返回 MC 和 JC 结的 MC. result 和 penalty * /
13.            MC. result ← false
14.            penalty ← JC. judge()
15. / * 设置阻塞时间、添加不当行为 * /
16.            UT ← time+penalty
17.            MC. list ← resourc
18.    end if
19. else
20. / * 间隔时间满足要求,频繁请求数为 0 * /
21.        p. NoFR ← 0
22.    end if
23. end if
24. / * 记录最后一次请求时间 * /
25. p. ToLR ← time
26. / * 根据 ACC 和 MC 返回最终结果 * /
27. result ← ACC. result and MC. result
28. / * 将最终结果和惩罚作为参数返回 * /
29. Trigger event returnResult(result,penalty)
```

5.3.1 注册合约 RC

注册合约 RC 存储 JC,MC 和 ACC 的信息并提供管理这些合同的功能。数据请求者可以调用 RC 部署新的 ACC 合约或者检索出所需的相关信息。注册合约(RC)包含多个函数,其中的主要函数如下:

1)*createContract*():可由 RC 合约调用,用于创建新的 ACC 合约。

2)*getDataInfo*():用于获取数据请求者所需的信息,如 ACC 地址、原始数据哈希摘要、密钥分发节点和第三方服务器等。

3)*updateDataInfo*():用于更新数据请求者所需的信息。

5.3.2 访问控制合约 ACC

访问控制合约 ACC 为区块链提供加密数据的访问控制方法,并且每种方法由一个 ACC 实现。一个用户可以与多个 ACC 相关联,ACC 通过检查预定义策略来实现静态访问权限验证。ACC 维护着一个访问控制列表(ACL),ACL 记录了数据请求者的访问权限,如表 1 所列。其中各字段的含义如下:流数据编号(Stream)、数据请求者(User)、数据请求者可访问的数据范围(Scope)、数字签名,用于验证请求者的许可信息(Permission)、授权时间(Grant-time)和权限过期时间(Expiry-time)。

表 1 访问控制

Table 1 Access control

Stream	User	Scope	Permission	Grant-time	Expiry-time
01	User1	[5,9]	0X52d65…	2023-03-05	2023-07-01
02	User2	[1,6]	0X37db5…	2023-03-01	2023-05-01
03	User3	[3,8]	0X4a55k…	2023-04-01	2023-06-01

ACC 中的主要函数如下:

1)*addPermission*():向访问控制列表中增加一个新请求者。

2)*removePermission*():数据所有者可以主动调用它来删除权限;如果发现某个策略已经过期,自动调用该方法来删除无用的权限。

3)*updatePermission*():更新请求者的许可信息。

4)*getPermission*():获取请求者的许可信息。

5.3.3 不当行为合约 MC

不当行为合约 MC 通过日志审计来记录不当行为,并形成不当行为列表(ML),如表 2 所列。通过检查用户的行为来实现动态访问权限验证。不当行为包括短时间内过于频繁地发送访问请求、尝试非法访问数据、未经授权修改数据等。对于已经记录的不当行为,可以采取相应的措施来保护系统的安全和正常运行。表 2 中的基本字段有:数据请求者(User)、遭受不当行为的设备(Object)、用户对数据具体的不当行为(Misbehavior)、出现不当行为时间(Time)和对不当行为实施的处罚措施(Penaity)。

表 2 不当行为

Table 2 Misbehaviors

Subject	Object	Misbehavior	Time	Penaity
user1	Sensor2	Too frequent requst	2023-03-05 11:40	Blocked for two hour
user1	Server3	Successive failure	2023-03-05 11:42	Blocked for one hour
user2	Server3	Attempt to access data illegally	2023-03-05 12:42	Revoke permissions

MC 中的主要函数如下:

1)*getmisbehavior*():用于检查用户是否已经被列入不当行为列表中。

2)*addmisbehavior*():用于将用户添加到不当行为列表中。

3)*determinePenaity*():根据不当行为历史来确定适当的处罚措施。

5.3.4 判断合约 JC

判断合约 JC 通过从 ACC 和 MC 接收返回的许可信息,来判断用户是否有访问权限。JC 会基于访问控制列表和不当行为列表来确定结果,一旦判断出用户具有访问权限,合约将执行相应的任务或操作;如果访问权限不正确或者不符合条件,合约将拒绝执行任务或操作,并向用户发送错误信息。

JC 的主要函数如下:

judgeRequest():验证请求者是否在访问控制列表中,若不在则拒绝访问,如果在列表中,则检查不当行为记录。若有记录,则根据处罚措施,通知 ACC 执行操作。

5.4 访问权限撤销

以往的方案在撤销访问期间,必须下载、重新加密数据,

将其上传回云中[13]，并将新密钥分发给有权访问的用户，对于只拥有旧密钥的用户，访问将被撤销。使用代理重加密进行用户撤销，无需解密数据内容，简化权限撤销过程，当撤销数据请求者的访问权限时，代理服务器只需撤销请求者的重加密密钥，将数据从一个密钥重新加密为另一个密钥，而不需要撤销已经与其他用户共享的密钥，如图 4 所示。由于区块链的不可篡改性，在更改或撤销数据请求者的访问权限时，并不是直接修改或删除原来的 ACL，而是添加新的 ACL 记录。新的 ACL 记录中包含对数据请求者的操作，如设置新的访问权限，或者撤销访问权限。旧的 ACL 记录被保留在区块链上，必要时可以对数据请求者的访问控制策略进行审计。当查询 ACL 获取请求者的访问权限时，将最新时间对应的 ACL 作为返回结果。

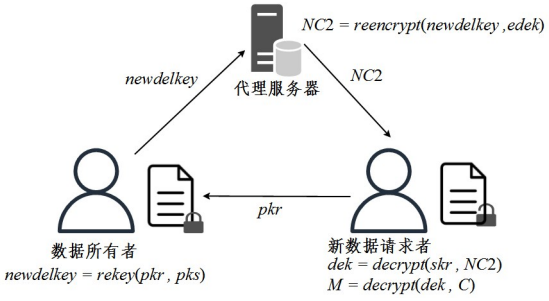

图 4 权限撤销

Fig. 4 Permission revocation

权限撤销的步骤如下：

1）数据所有者根据新数据请求者的公钥 pk_r 生成一个撤销消息 $newdelkey$，撤销消息包含有关要撤销的密钥的相关信息和新的重加密密钥，并将撤销消息发送给代理服务器。

2）代理服务器根据撤销内容，使用新的重加密密钥 $newdelkey$ 对数据进行重加密 $NC2=reencrypt(newdelkey, edek)$，并将重加密数据 $NC2$ 发送给新请求者。

3）新数据请求者使用私钥 sk_r 解密 $NC2$ 得到访问令牌 $K=decrypt(NC2, sk_r)$，通过访问令牌，解密 C 得到明文 M。

5.4.1 定时撤销

在访问控制列表 ACL 中，可以设定请求者的权限过期时间 Expiry-time，以定时撤销该请求者的访问权限。当权限到期时，请求者将不再被允许访问该资源或执行某些操作。设定权限过期时间可以帮助保护资源免受未经授权的访问，同时也可以确保权限的过期和更新得到及时处理，以便管理和控制资源的访问权限。

5.4.2 即时撤销

通过 MC 机制，审计系统日志来发现请求者的不当行为和恶意行为，形成不当行为列表 ML，对用户进行权限限制或即时撤销访问权限。例如，如果某个请求者在短时间内过于频繁地发送访问请求，将会暂停其访问权限；如果某个请求者涉嫌恶意攻击、诈骗、欺诈等行为，将撤销其相应的访问权限。

6 安全性分析

6.1 完整性

每个区块经过椭圆曲线算法 EDCSA 进行数字签名。使用数据所有者的私钥对加密消息 C_i 进行签名 $Sign(SK_s, C_i)\rightarrow SigC_i$，根据数据所有者的公钥 pk_s 对加密消息 C_i 进行签名验证 $Verify(pk_s, C_i, SigC_i)\rightarrow$(true, false)。该签名可以在给定公钥和密文的情况下，允许各方验证数据的真实性和完整性。

6.2 向前保密性

加密密钥是用 HASH 树构建的 PRF[24-25] 推导出来的，对于主密钥 k，第 i 个密钥推导为 $T(k,i)\rightarrow k_i$。共享密钥时共享树的内部节点，即 $T.con(k, S:=\{i,\cdots,j\})\rightarrow k_S$，此节点可派生 $T.eval(k_S,i),\cdots,T.eval(k_S,j)$，但不能派生为其他键。拥有令牌 k_S 的攻击者无法导出 $k_i,\cdots,k_j$ 之外的密钥。

6.3 不可伪造性

只要攻击者不破坏私钥，多项式时间内攻击者就不能伪造签名。中间人无法通过篡改公钥或伪造合法签名来通过验证，实现抗中间人攻击。

6.4 抗链接攻击

加密数据在链上都是密文，其安全性通过对称加密算法和 AES-GCM 加密算法保证，攻击者无法获得更多用户隐私信息，做到抗链接攻击。

7 实验评估

7.1 可用性分析

为评估本文方案在低功耗设备物联网上的可用性，本文总结了在 4 个不同平台上使用本文方案的加密方法的操作成本，如表 3 所列。每个块开销包括密钥计算、块加密、密钥加密和签名，计算开销最大的是加密操作。在物联网设备的软件中运行加密操作的成本最高，大多数物联网设备都配备了加密加速器，用于 AES 加密。在没有加速器的传统物联网设备上，尽管签名操作速度降低了 50%，但与 ABE 方案相比，依然是可行的。

表 3 加密操作成本

Table 3 Encryption operation cost

	加密时间/μs	加密/(op/s)	签名时间/ms	签名/(op/s)
传统 IoT	288	3.90×10^3	267.0	3.8
加速器 IoT	48	2.78×10^4	173.0	5.7
手机	47	2.50×10^4	4.7	257.0
电脑	5.5	1.97×10^5	1.5	752.0
云	2.5	3.91×10^5	1.3	895.0

本文将文献[14-16]中所提到的 3 种去中心化授权撤销方案与本文方案进行对比，比较结果如表 4 所列。

表 4 撤销方案对比

Table 4 Comparison of revocation plans

方案	去中心化	定时撤销	即时撤销	撤销依据	代理重加密	智能合约
文献[14]中的方案	Y	N	N	List	N	Y
文献[15]中的方案	Y	N	Y	GM	Y	N
文献[16]中的方案	Y	N	Y	RL	N	N
本文方案	Y	Y	Y	ACL	Y	Y

文献[14]中的用户列表由 MS 管理，其标记恶意用户并撤销他们的访问权限。在文献[15]中，当用户从系统中撤销时，组管理器 GM 更新未撤销用户的组密钥并生成重加密

密钥。文献[16]将用户的 ID 添加到撤销列表中，并重新加密数据进行即时撤销。本文方案通过 ACL 和 ML 进行定时和即时撤销，实现了实时撤销。

7.2 性能分析

为了证明本文方案的有效性，进行了仿真实验，重点是算法的性能比较和智能合约的性能分析。实验的硬件环境是两台台式电脑、两台笔记本电脑，具体硬件信息如表 5 所列。由 Java，Python 和 Go 等语言进行代码实现，使用多线程技术模拟多个验证节点，实验数据均为普通文档数据。

表 5 实验设备

Table 5 Experimental equipments

Device	CPU	OS	RAM/GB	ROM
Lenovo Legion T7	Intel Core i5 4.60GHz	Windows 11	16	1TB
HP 288 Pro G6	Intel Core i5 3.10GHz	Windows 10	8	1TB
Lenovo IdeaPad 15	AMD R5 5500U	Windows 11	8	256GB
Lenovo Legion T7	Intel Core i5 4.60GHz	Linux 7	4	256GB

图 5 给出了不同方案对同一数据进行 100 次加解密操作的平均时间的对比结果。文献[14]中的加密阶段可以分为本地加密阶段和雾节点加密阶段，加解密时间比文献[16]中的方案长。文献[15]采用外包加密，包含外包加解密算法和本地加解密算法，消耗的时间更长。文献[16]是在属性个数为 25 的情况下进行测试，虽然比本文方案的加密时间短，但随着属性个数的增加其加密时间呈线性增加。本文方案的总体效率是最高的。

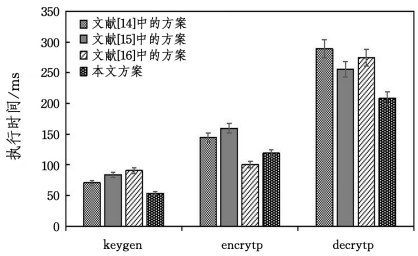

图 5 执行时间对比

Fig. 5 Execution time comparison

本文使用分布式网络测量本文方案的通信成本，网络大小为 8～512 个节点，如图 6 所示。

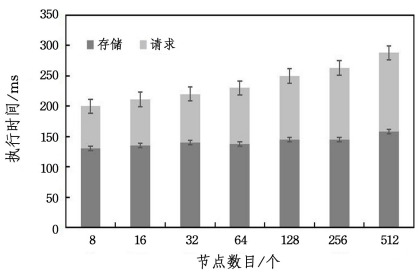

图 6 单个存储和获取请求的时间

Fig. 6 Time for single storage and getting requests

本文方案的大部分时间用于路由和检索，部分时间是签名操作造成的。随着节点数量的增长，总的获取时间从 7 ms 增加到 132 ms。本文方案的获取时间少于存储时间，原因是获取请求的路由开销低于存储请求。对于获取请求，一旦找到持有数据块的节点，路由过程就会中止。

本文区块链智能合约是基于 Hyperledger Fabric 搭建的。智能合约创建成本和功能执行成本如表 6 所列。可以看出，最高成本是智能合约部署，部署一条新的 ACC 所需的成本大约是 0.578 美元，但这种成本只是初始化系统的一次性成本，其他函数成本都相对较低，查询、增加和删除等函数所需成本不到 0.05 美元。

表 6 智能合约函数成本

Table 6 Cost of smart contract functions

Function	Gas used	Cost
creatContract	1813010	0.578
getDataInfo	39643	0.012
updateDataInfo	146578	0.047
addPermission	149432	0.048
removePermission	101819	0.033
updatePermission	164526	0.039
getPermission	64852	0.019
getmisbehavior	59186	0.018
addmisbehavior	183531	0.050
determinePenaity	165972	0.049
judgeRequest	84562	0.020

按模块分别测试 RC，ACC，MC 和 JC 在不同并发数下的吞吐量，如图 7 所示。本文方案的吞吐量保持在相对平滑的水平，并且没有一个超过 520 的阈值，各模块在密集请求下仍能保持较高的吞吐量和相对一致的性能输出。

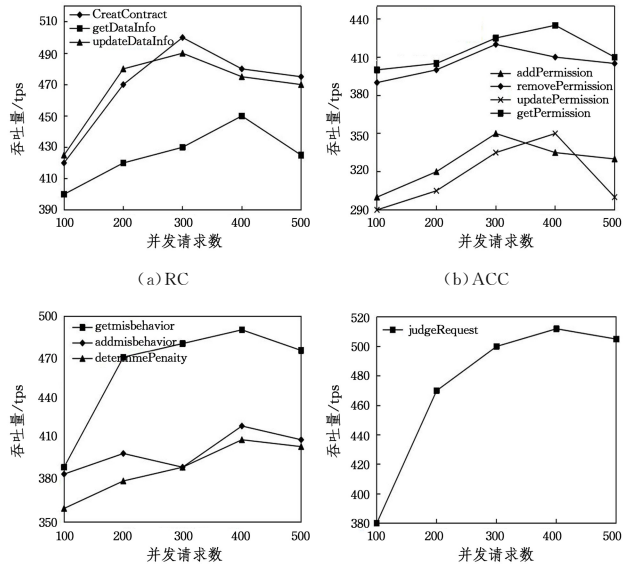

图 7 智能合约吞吐量

Fig. 7 Smart contract throughput

结束语 本文提出了一种流数据外包存储场景下的实时授权撤销方案，改进了流数据加密过程，在撤销访问期间，不再下载重新加密数据；并且提出了一种基于智能合约的授权撤销框架，确保只有经过授权的用户才能够访问特定资源，并实时撤销权限，保证数据的安全性和可控性。本文方案仅能在同一域内进行访问控制和权限撤销，存在跨域授权撤销的局限性。在下一步工作中将继续探讨并改进所提方案，解决跨域授权撤销问题。

参 考 文 献

[1] MACIEL L，BALLINI R，GOMIDE F. Adaptive fuzzy modeling of interval-valued stream data and application in cryptocurrencies prediction[J]. Neural Computing and Applications，2023，35(10)：7149-7159.

[2] SRIRAMG S. Edge computing vs. Cloud computing：an overview of big data challenges and opportunities for large enterprises [J]. International Research Journal of Modernization in Engineering Technology and Science，2022，4(1)：1331-1337.

[3] ZHANG T，SHEN J，LAI C F，et al. Multi-server assisted data sharing supporting secure deduplication for metaverse healthcare systems[J]. Future Generation Computer Systems，2023，140：299-310.

[4] RASORI M，PERAZZO P，DINI G，et al. Indirect revocable kp-abe with revocation undoing resistance[J]. IEEE Transactions on Services Computing，2021，15(5)：2854-2868.

[5] DAS S，NAMASUDRA S. Multiauthority CP-ABE-based Access Control Model for IoT-enabled Healthcare Infrastructure [J]. IEEE Transactions on Industrial Informatics，2022，19(1)：821-829.

[6] WANG W，HUANG H，YIN Z，et al. Smart contract token-based privacy-preserving access control system for industrial Internet of Things[J]. Digital Communications and Networks，2023，9(2)：337-346.

[7] SAIDI H，LABRAOUI N，ARI A A A，et al. DSMAC：Privacy-aware Decentralized Self-Management of data Access Control based on blockchain for health data[J]. IEEE Access，2022，10：101011-101028.

[8] BURKHALTER L，HITHNAWI A，VIAND A，et al. TimeCrypt：Encrypted Data Stream Processing at Scale with Cryptographic Access Control [C]// 17th USENIX Symposium on Networked Systems Design and Implementation. 2020：1053-1062.

[9] CHOKSY P，CHAURASIA A，RAO U P，et al. Attribute based access control(ABAC) scheme with a fully flexible delegation mechanism for IoT healthcare[J]. Peer-to-Peer Networking and Applications，2023，16(1)：1445-1467.

[10] TAO J，CHEN X，MA J. Public Integrity Auditing for Shared Dynamic Cloud Data with Group User Revocation[J]. IEEE Transactions on Computers，2016，65(8)：2363-2373.

[11] WU L，WANG J，ZEADALLY S，et al. Privacy-preserving auditing scheme for shared data in public clouds[J]. The Journal of Supercomputing，2018，74(11)：6156-6183.

[12] HÖGLUND J，FURUHED M，RAZA S. Lightweight certificate revocation for low-power IoT with end-to-end security[J]. Journal of Information Security and Applications，2023，73：103424.

[13] SHAFAGH H，BURKHALTER L，RATNASAMY S，et al. Droplet：Decentralized Authorization and Access Control for Encrypted Data Streams[C]// USENIX Security Symposium. USENIX Association. 2020：2469-2486.

[14] YANG Y，SHI R，LI K，et al. Multiple access control scheme for EHRs combining edge computing with smart contracts[J]. Future Generation Computer Systems，2022，129：453-463.

[15] ZHANG R，LI J，LU Y，et al. Key escrow-free attribute based encryption with user revocation[J]. Information Sciences，2022，600：59-72.

[16] YU K，TAN L，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.

[17] WIRAATMAJA C，ZHANG Y，SASABE M，et al. Cost-efficient blockchain-based access control for the internet of things[C]// 2021IEEE Global Communications Conference. IEEE，2021：1-6.

[18] LI D，HAN D，CRESPI N，et al. A blockchain-based secure storage and access control scheme for supply chain finance[J]. The Journal of Supercomputing，2023，79(1)：109-138.

[19] LIU J，LIU Z，SUN C，et al. A data transmission approach based on ant colony optimization and threshold proxy re-encryption in wsns[J]. Journal of Artificial Intelligence and Technology，2022，2(1)：23-31.

[20] LIN S Y，ZHANG L，LI J，et al. A survey of application research based on blockchain smart contract[J]. Wireless Networks，2022，28(2)：635-690.

[21] SHI J F，WU H，GAO H R，et al. Overview of parallel execution models for blockchain smart contract transactions [J]. Journal of Software，2022，33(11)：4084-4106.

[22] AGRAWAL T K，ANGELIS J，KHILJI W A，et al. Demonstration of a blockchain-based framework using smart contracts for supply chain collaboration[J]. International Journal of Production Research，2023，61(5)：1497-1516.

[23] YIN Y Y，YE B Y，LIANG T T，et al. Research on multi-layer blockchain network model in edge computing scenario [J]. Journal of Computer Science，2022，45(1)：115-134.

[24] CHEN D，WANG H，ZHANG N，et al. Privacy-preserving encrypted traffic inspection with symmetric cryptographic techniques in IoT[J]. IEEE Internet of Things Journal，2022，9(18)：17265-17279.

[25] CASACUBERTA S，HESSE J，LEHMANN A. SoK：Oblivious Pseudorandom Functions[C]// 2022 IEEE 7th European Symposium on Security and Privacy(EuroS&P). IEEE，2022：625-646.

MEN Ruirui，born in 1999，postgraduate. Her main research interests include cryptography and zero trust security.

JIA Hongyong，born in 1975，Ph.D，lecture. His main research interests include cloud computing security and zero trust security of the IoT system.

（责任编辑：喻藜）