上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

学士学位论文

BACHELOR'S THESIS

论文题目：边缘计算架构下基于区块链技术的隐私保护方案

学生姓名：崔晏哲

学生学号：516030910378

专　　业：自动化

指导教师：龙承念 教授

学院(系)：电子信息与电气工程学院

上海交通大学

本科生毕业设计（论文）任务书

课题名称：边缘计算架构下基于区块链技术的隐私保护方案

执行时间：____年____月 至 ____年____月

教师姓名：龙承念　　职称：教授

学生姓名：崔晏哲　　学号：516030910378

专业名称：自动化

学院(系)：电子信息与电气工程学院

毕业设计（论文）基本内容和要求：

调研边缘计算与区块链技术，了解网络边缘端存在的隐私风险以及目前基于区块链技术的解决方案。面向边缘计算环境，深入分析隐私风险类别和常见的攻击模型，提出基于区块链技术的隐私保护方案。基于移动边缘计算的部署架构，为方案设计相关的验证实验，进行评估与分析，并撰写毕业论文。

毕业设计（论文）进度安排：

序号	毕业设计（论文）各阶段内容	时间安排	备 注
1	调研边缘计算与区块链技术；对边缘计算架构下的隐私问题以及基于区块链技术的解决方案有一定了解。	2019.11-2019.12	
2	了解物联网中常见的隐私安全攻击，面向边缘计算环境，提出基于区块链的隐私保护方案，完成中期审查。	2020.1-2020.2	
3	为提出的隐私保护方案设计验证实验，对实验结果进行评估与分析。	2020.3-2020.4	
4	完成毕业论文撰写。	2020.4-2020.5	

课题信息：

课题性质 ： 设计 √　　论文□

课题来源*： 国家级 √　　省部级□　　校级□　　横向□　　预研□

项目编号 61873166

其他_____

指导教师签名： 龙承念

2019 年 11 月 13 日

学院（系）意见： **通过** 院长（系主任）签名：周越 2019 年 11 月 14 日
学生签名：崔晏哲 2019 年 11 月 13 日

上海交通大学

毕业设计（论文）学术诚信声明

本人郑重声明：所呈交的毕业设计（论文），是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

作者签名：

日期： 2020 年 6 月 25 日

上海交通大学

毕业设计（论文）版权使用授权书

本毕业设计（论文）作者同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权上海交通大学可以将本毕业设计（论文）的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本毕业设计（论文）。

本论文属于

保密□，在___年解密后适用本授权书。

不保密 √。

（请在以上方框内打“ √ ”）

作者签名：　　指导教师签名：

日期：　　年　　月　　日　　日期：　　年　　月　　日

边缘计算架构下基于区块链技术的隐私保护方案

摘要

随着智能终端设备数目大幅增长，人们逐渐意识到海量、呈指数级增长的设备及其交互数据具有潜在的泄露风险，原有集中式云计算方案面临无法承受这样负荷的难题，开始思考如何妥善保存、利用和处理如此规模的数据。边缘计算将计算资源下放到临近用户的边缘侧上，作为新兴的架构模式，逐渐承担起传统云计算扮演的角色。然而，边缘环境上的设备普遍存有薄弱的安全防护机制，容易具有隐私泄露和代码劫持的隐患。区块链技术基于自身结构和密码学原理提供去中心化、不可篡改和可追溯等特性，为实践边缘计算架构下的隐私保护提供了一种可行的解决方案。本文提出一种由区块链、终端层和边缘服务器层组成的可信边缘计算架构，借由区块链的可信机制撑起方案里中枢控制的角色，再以同态加密技术为边缘场景下的数据流通起到隐私保护效果，期望在数据外包的场景下保护用户的隐私。实验结果表明，使用以太坊、微信小程序以及腾讯云开发模拟的三点架构，每轮数据分析约耗时一分钟左右完成，具有足够适应该场景的性能，验证了方案所期待的：即使在物联网有限资源的设备上，也能不牺牲用户的隐私同时拥有良好的使用体验。

关键词：区块链，边缘计算，隐私保护，同态加密

PRIVACY PROTECTION SCHEME BASED ON BLOCKCHAIN TECHNOLOGY UNDER EDGE COMPUTING ARCHITECTURE

ABSTRACT

With the widespread popularity of smart terminal devices, people gradually realize that the mass of device usage data in exponentially growth has potential risks. The former centralized cloud computing solution failed to bear such loads, and people began to think about how to properly save, utilize and process data at this scale. Edge computing decentralizes computing resources to the user's edge side. As an emerging architectural model, edge computing provides a novel solution to relieve the storage burden in cloud center. However, devices on the edge of the environment generally have weak security protection mechanisms, which causes hidden risks such as privacy disclosure and malicious code attacks. Blockchain technology has decentralization, non-tampering and traceability, providing a feasible solution for privacy protection under the edge computing architecture. In this paper, we propose a three-point architecture consisting of a blockchain, a terminal layer and an edge server layer. With the trusted mechanism of blockchain for the central control role in the scheme, and with homomorphic encryption technology for data circulation in the edge environment, we expect to protect user privacy in the scene of data outsourcing. The experiment shows that each round of data analysis takes about one minute to complete, with using Ethereum, WeChat mini-program, and Tecent Cloud Base to simulate the scheme. It has sufficient performance to adapt to the scene, verifying what the program expects: Even on devices with limited resources on the Internet of Things, they can have a good user experience without sacrificing user privacy.

Key words: Blockchain, Edge Computing, Privacy Protection, Homomorphic Encryption

目录

第一章 绪论

第四代移动通信技术的普及让智能手机出现过一段大规模的增长，再随着第五代移动通信技术的到来，用户的终端使用设备数和随之产生的数据量正处于井喷式的增长[1]。这些海量的数据如何妥善利用、处理和保护是一大难题，而在一些低延时场景里诸如无人驾驶领域，如何快速高效的计算这些数据亦是一大难点，这些皆成时下学界研究的热点。

区块链作为一项分布式的点对点系统，其特点是无需第三方背书和校验即可形成一套安全可信任的自治网络[2]，成为当今一套新兴的去中心化数据存储方案。

边缘计算为人们近年来所提出的一种有别于传统集中式计算的服务模式，用于解决传统云计算借由一个或少数若干个集中式服务器所面临的计算资源紧张的问题，其凭借将数据下放到边缘侧的边缘设备上进行处理，从而避免集中式服务器潜在拥有的安全隐患，诸如单点攻击、DoS拒绝服务攻击以及DDoS拒绝服务攻击等；另外，由于数据请求侧与计算侧大幅拉近，边缘计算架构同时能减少通信上的延迟，满足人们日益俱增对于低延时的要求。

本课题的研究重点聚焦在边缘场景下的用户数据隐私保护，以区块链做底层技术支撑，满足行业在数字化变革中对数据聚合、通信以及存储时所期望的安全与隐私保护等方面的要求。

1.1 研究背景及其意义

随着智能终端设备日趋普及，设备联网数激增，形成规模日渐庞大的万物互联的IoT物联网。这些智能设备已突破传统一台笔记本电脑和一台智能手机的范围，涵盖着诸如智能手环、智能手表等穿戴式装置以及智能家居里智慧屏、智能冰箱以及智能音箱等设备，构筑了庞大贴近用户侧的终端生态。

然而，这些在边缘场景里产生的海量数据逐渐浮现出了安全隐患。2020年1月,小米智能家居产品米家智能摄像头就遭网友曝光，当其与Google的智能音箱Nest Hub集成之后，屏幕中意外出现了非预期的静态图像，画面中呈现老人在客厅熟睡的样子，也有小孩躺在婴儿床内熟睡的模样。

图 1-1 米家智能摄像头与智能音箱 Nest Hub 集成后的隐私泄露图像[1]

此事立刻引来各大媒体的注意，Google方面在第一时间为此事做出回应，表示已经关闭了在Nest hub上所有小米智能家居产品的第三方应用，也旋即与小米公司联系，着手调

[1] https://www.xda-developers.com/google-temporarily-kills-xiaomi-mi-home-integration-security-camera-bug/

查潜在的问题并且寻求解决方案。

这起意外事件凸显出了物联网下隐私泄露问题，尤其这种居家场景在当今物联网里是非常常见的，因此这些包含大量用户贴身信息出现隐私泄露时格外令人忧心。

对于这类问题，现有的解决方案是将整个计算架构迁移或部分迁移成边缘计算架构，而非传统中心化的云计算模式。在当今，借由传统的集中式服务器处理数据，已曝露出以下几点问题:

(1) 通信拥堵: 随着联网设备数激增，海量数据不断涌入的情况下，中心化服务器没有那么大的能力处理这样规模的计算。

(2) 隐私泄露: 服务器可能获取到来自终端的明文数据，这些数据（身份状态、健康信息或生活习惯等）通常是敏感的，如遇上中间人攻击或单点攻击，可能存在隐私泄露的风险。

(3) 数据安全: 由于信息存储于集中式的服务器上，数据具有被有不法人士篡改或移除的风险，面临数据不可信的难题，比如2020年2月微盟公司员工删除数据库事件[2]。

因此，已有愈来愈多人意识到边缘计算的重要性，将数据流通的范围限制在边缘侧，也就是说只在局域网的情况下交换、计算和存储数据，不在外网上公开。

在数据保存上，若引入区块链对数据进行加密处理、验证和存储，在架构上由于区块链中的数据由多方共同维护，避免了单点攻击或恶意数据篡改的现象；而区块链的底层依赖于密码学原理，不仅确保数据的可用性和完整性，还能保障这些数据的隐私性。

然而，当我们提到区块链的隐私性时，最常将其关联至账户地址的匿名性上，这是因为区块链的账户地址是由一串随机、看似无意义的字符串所组成，无法与真实的用户做对应。不过，以比特币为例，每个账户的比特币地址是可查询的，一个账户的历史皆是公开可访问的，这就使得人们有机会利用统计学或大数据等手段凭借少数若干已知的真实用户，分析出更多地址对应的真实用户，进而发生隐私的泄露。目前，区块链的隐私性仍有十足的研究空间，其中研究所涉及的零知识证明，简而言之是一种验证者无需得知证明者的秘密信息、也可验证证明者握有秘密信息的能力。

本文聚焦在区块链如何保障边缘层的数据，使其起到隐私保护的作用，创新点在于结合了同态加密技术，让边缘服务器在无从得知或无法解密出终端层的敏感信息的情况下，也能顺利的为系统执行期望的计算；再基于区块链技术实现可信的访问控制，为不可信的边缘环境提供可信的节点支持。

本文将先从边缘计算涉及的知识着手研究，探讨边缘计算常见的架构以及解释其中潜在的风险，接着研究区块链的原理以及同态加密技术，再开发以太坊上的智能合约来发挥区块链公正可信的作用，最后在过程中探究其中数据流通的效果是否达到人们所期待的要求，进而展望未来可以改进的地方。

1.2 国内外研究现状

国内外研究现状将区分成边缘计算、隐私保护方法和区块链三领域去讨论，并都聚焦在隐私保护和数据安全两个层面上。

1.2.1 边缘计算架构下的信息安全和隐私保护

关于边缘计算中基于安全以及隐私保护的研究里，主要侧重于宏观的安全性评估、对加密数据的处理以及数据完整性的问题上。

在宏观的安全性评估中，已有人在终端设备的数据安全上进行研究，针对移动边缘范式展开了安全性分析，并提出一种可移植的安全防护架构，该工作对边缘计算的安全性评

[2] https://new.qq.com/omn/20200228/20200228A0RA4200.html

估提供了理论依据和实质的帮助[3]；也有人认为不应对系统区分内外，而应该采用“零信任”模式，颗粒化地记录每个设备的访问状态或用户特征，以对每个单点做记录和处理的方式更好的适配突发情况[4]。

针对边缘计算的数据安全课题上，研究者多半会先列举常见的危害，比如在边缘侧常遇的安全威胁主要会有拒绝服务攻击（DoS）、中间人攻击（MITM）以及伪造网关攻击，再给出可行的策略去因应这些难题。2013年，有学者提出了一种基于密文策略属性的关键字搜索加密方案，能实现对被加密的密文进行关键词查找[5]；另外，还有的研究重点放在如何不泄露用户隐私的前提下，对用户的数据进行审计、搜索和更新的操作，比如有学者提出基于概率公钥加密技术[6]以及一套CP-ABE方案构建关键词排名搜索算法[7]，实现在资源受限的移动设备上做到外包数据的隐私保护。

除了讨论通信上的隐私，如何确保从终端传输到边缘端或云端数据中心的数据完整性也是一大问题。该问题涉及完整性审计，有学者提出一种分布式数据审计系统，该系统一大特色是利用同态认证器和随机掩码来隐藏隐私数据，保证第三方审计平台无从得知数据内容但又能考察数据的完整性和可用性[8]。

1.2.2 常见的隐私保护研究

边缘计算架构中，用户的数据通常需要存于或转交给半可信的授权节点，同时这些被授权的节点如何确保存于其上的用户隐私不被泄露也是学界研究的焦点之一。常见的隐私保护方法有匿名化处理、随机化处理以及差分隐私等三种，以下针对这些方法进行研究综述。

匿名化处理指的是在对外发布数据时对源数据进行限制发布的手段，实务操作上有抑制和泛化之分。前者通常指的是剔除某些具有身份标识的字段，后者则指对数据进行合并或概括成更为一般的描述。其中，该方法的一个典型是k-anonymity模型，由学者Samarati[9]和Sweeney[10]所提出。该模型的基本思想是尽可能切断准标识符和敏感属性之间的联系，通过降低某个字段的精细度，使每条记录中的某个字段具有和其他k-1条记录完全相同的值，当k值越大隐私保护的程度越高，表中损失的信息也就越多，但也因此保证用户的隐私。

随机化处理是一种让数据失真进而保护隐私的方法，主要通过在发布数据前对数据源模糊化或进行扰动以实现。有学者提出了一种部分隐藏的随机化回答方法[11]，其基本思想是将限制发布和数据扰动相结合，在对数据扰动的同时也把数据的部分信息进行隐藏，巧妙地兼顾真实数据的隐匿性和发布数据的准确性。

差分隐私则是一种针对背景知识的攻击者所提出的隐私保护方法，由Dwork于2006年所提出[12]。差分隐私提供了严格的关于隐私保护的数学定义，同时还拥有一套关于隐私保护程度的量化指标。此外，Dwork等人还提出了一个实现差分隐私的拉普拉斯机制[13]，只要输出属于实数集，就可以借由拉普拉斯噪声对源数据进行扰动从而实现差分隐私；后来，有学者提出一种中位数机制[14]，突破了拉普拉斯机制中输出必须是实数的限制，从而实现任意形式的输出。近几年，差分隐私广泛应用于科技公司提供的产品或服务中，比如苹果公司即在其手机的iOS系统上引入差分隐私，借以提升用户使用智能语音助手和其他数据外包服务的隐私性。

1.2.3 基于区块链技术的隐私保护研究

在区块链上，因为其本身存储着大量的信息，目前在常见的电子支付场景里已容纳着许多用户的交易细节，因此链上的隐私保护也值得注意。

有学者指出一种盲签名的方法来隐藏交易细节[15]，借由第三方的混币服务，实现从外部来看呈现出一种多输入多输出的交易模型，让攻击者无从得知交易发生时单一输入地址

和输出地址直接的关系；还有的学者提出基于零知识证明的加密协议[16]，通过该协议使比特币转为零币，零币在旁人是一种只能看出是否被消费，但却无从得知更多交易细节的加密货币。

另外，区块链的底层原理依赖于密码学，在2009年以前并未有区块链一词，对于其的研究多半从密码学展开。早在1979年，Shamir和Blakley各自提出了秘密共享方案[17][18]：Shamir提出以插值法为背景的方案，而Blakley则突出关于高斯消元法的方案。其主要内容为：在（t，n）秘密分享体制中，将一个秘密值分成n个分片，分发给n个人，欲恢复完整的秘密值，至少需要门限阈值t个分片才有可能得出原来的结果，少于t个分片皆无法获取完整的秘密信息。该方案已广泛应用于门限密码、拜占庭协议、属性密码和安全多方计算等领域，在密码学里具有举足轻重的地位。

而区块链中涉及物联网的隐私讨论里,多半采用两种方案：

（一）服务器可以直接对明文进行运算

该方案显而易见的，有用户隐私泄露的疑虑。

（二）服务器仅能得到来自终端的密文数据

这种方案遇到的问题是，难以对密文进行同态运算。

在大多数的方案里,计算的可行性和用户的隐私性一直难取得平衡。尤其在区块链发展初期,以比特币为代表的区块链网络,用户地址是完全公开的，拥有“假名性”而非真实无可逆推的“匿名性”，使得用户隐私受到一定的挑战。然而这不代表区块链无法保护用户的隐私，事实上正因为其本身拥有的假名性和底层依赖于密码学原理，这些先天因素让人们相信这个新技术有机会保障人们在网络上产生数据的隐私。正基于此，区块链如何确保用户隐私又能对数据进行计算具有充分的研究价值。

1.3 本文研究创新性以及组织结构

本文从区块链如何应用在边缘计算架构中起到隐私保护作用这个研究课题出发,提出一种基于中枢区块链、终端设备以及边缘服务器的三点架构，结合同态加密算法、非对称加密算法以及数字签名技术，构筑一个边缘计算架构下可信且可计算的隐私保护方案。中枢区块链作为信任节点为不可信的参与者背书，解决系统里互相不信任的难题，同时由于中枢区块链是为一种分布式结构，难以实现单一节点把持整个网络各节点的控制权，令单点攻击变得困难甚至不可能；此外，借由同态加密技术，让密文在计算后仍具有意义，满足用户明文数据不愿被任意其他方得知的需求，同时又能让数据被利用起来；如此，以边缘服务器为代表的计算方和以中枢区块链为代表的信任背书方都无法得知终端的明文数据，保护了终端用户的隐私却不影响整个系统的运行，兼顾了数据的可用性和用户的隐私性。

本文共分为五个章节，重点研究区块链如何在边缘环境里参与协作以及如何为此环境构筑一个可信同时能确保用户隐私的架构，主要内容安排如下：

第一章为绪论，阐述研究动机和研究背景，由于原有的计算模式遇到了通信拥堵、隐私泄露和数据安全等难题，引出了诸如边缘计算和区块链等新兴技术以尝试解决这些问题；此外，综述了介绍了边缘计算、隐私保护方法、区块链以及相互间交叉的研究现状，剖析已有学者从不同角度切入隐私保护领域所得的研究成果；最后，在用户隐私领域还分析了区块链在涉及边缘计算时所遇到的挑战，同时点出即使隐私性不足却何以能进一步发展的地方。

第二章为预备知识，主要分为三个部分：边缘计算、区块链基础以及同态加密技术。在该章第一节，首先从边缘计算的概念展开，引出常见的边缘计算架构体系，同时给出边缘计算模型潜在的安全危害；在该章第二节，从区块链诞生的背景开始，由浅入深探讨其所涉及的知识面，包括区块基本结构、哈希加密、Merkle Tree、链状结构、分布式存储以

及共识机制等六个部分；最后在该章第三节，则解释何为同态加密技术，先概述同态加密的用途和特性，接着以Paillier算法为例剖析其原理，研究其是如何在明文空间和密文空间转换的，同时如何在密文空间进行计算。希望经由第二章的陈述，让后续段落关于方案模型和方案实作的研究更易于理解。

第三章为方案模型，将在本章介绍基于区块链的隐私保护方案，从一个很常见的场景出发：“科技公司的分析者欲分析终端设备的统计值以改善下一代产品”，拆解成十一个步骤介绍这个方案。当中有七步为关键的边缘场景，这七个步骤详细陈述了中枢区块链、终端设备以及边缘服务器的关系和协作方式，以一种巧妙的三角关系让用户的数据不被非终端设备得知也能顺利的完成外包所需的计算，满足用户的隐私性和数据的可用性。

第四章为方案实践，在这一章里我们搭建了一个基于以太坊和腾讯云的测试平台以模拟整个方案。过程中，使用以太坊作为区块链的底层平台，在其上部署智能合约，扮演访问中枢的角色；然后以微信小程序作为各终端设备的载体，模拟用户输入和存有的私密值；再以腾讯云开发的云函数模拟边缘服务器层，以之处理同态加密上密文域的计算。同时，对整个实作方案进行性能评估，针对潜在的问题做出解释和对策，供相关研究人员参考。

第五章为总结和展望，本文的结果将在这一章呈现，回顾本文的同时展望未来还需要的努力，同时开放系统源代码至公开可访问的网页地址上，期望在该课题的研究中贡献一己绵薄之力。

第二章 预备知识

本章节主要研究围绕本课题展开的关于边缘计算、区块链技术和同态加密等基础知识。

2.1 边缘计算

本小结就边缘计算以及其衍生的安全隐患进行分析。

2.1.1 边缘计算和其潜在的安全危害

边缘计算中，“边缘”一词是个相对的概念[19]，可以指从数据源到云计算中心之间的任意网络资源和计算资源。有许多组织和机构对边缘计算进行过定义，尽管这些定义不全相同，但在边缘计算的概念上达成了共识：边缘计算是指在网络边缘执行计算的一种新型计算架构，有别于传统集中式云计算的模式。

边缘计算允许终端设备将存储和数据处理的服务移转至网络边缘的节点中，不仅满足了终端设备有限存储空间和计算能力的扩展，还能适当节约数据源到云计算中心通信链路上所消耗的资源。常见的边缘计算架构体系从中心到边缘大致可分为核心基础设施、边缘数据中心、边缘网络以及终端设备，如图 2-1 所示。

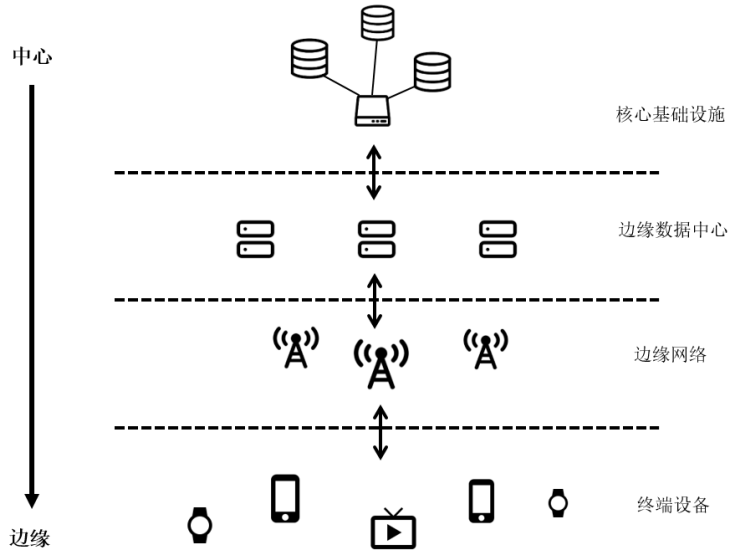

图 2-1 常见的边缘计算架构体系

以下分别阐述图 2-1 四层的概念及潜在的安全危害：

（1）核心基础设施主要为网络上的边缘设备提供集中式云计算服务。集中式云计算服务依照服务的面向群体和使用的难易程度的不同分为基础设施即服务（IaaS）、平台即服务(PaaS)以及软件即服务（SaaS）等三种服务模式。在边缘计算结构下，许可多个云服务厂商同时为客户提供核心基础设施这一层的服务，因此可借由安装多种不同形态的服务器来快速完成这一层的计算模式迁移，并且在相异的物理位置上为这一层的用户同时提供快速移动的代理和即时的响应。然而，这一层的服务最为中心化，除了让数据持有方容易产生操纵垄断的不透明行为，集中式服务还通常面临单点攻击的危害，进而引发用户数据被篡改和用户隐私被泄露的问题。

（2）边缘数据中心提供多服务管理和虚拟化服务，前者主要指第三方服务的开发商在

使用到核心基础设施的基础上提供不同服务给客户，而后者指的是将各式硬件的实体资源抽象成一个个虚拟资源，以更好地满足业务上的需求。在这一层中，由于多服务管理涉及不同平台的协作，一个第三方开发商若不怀好意将大大危害整个系统的安全，通常需要访问控制、身份认知、安全多方计算或数据匿名化等技术来提升平台的安全性，以减少物理攻击、服务垄断、数据被篡改以及用户隐私泄露等情况，或降低这些情况所带来的影响。

（3）边缘网络借由融合多种通信渠道实现传感器和物联网设备的互通互联，这种融合的网络架构容易遭受攻击，因为攻击者可以仅从某个网络单元接入对网络内部进行破坏，这些攻击主要包括拒绝服务攻击、分布式拒绝服务攻击或中间人攻击等。

（4）终端设备泛指与用户贴近、产生用户使用数据以及被视为数据源的所有类型设备，在整个边缘计算架构下其不仅代表着服务的使用者身份，在其上制造的数据还会参与到网络各个环节里，从源头保证网络的安全就显得尤为重要。在最接近用户的这一层里，会出现的安全危害有用户密钥保管不当所产生的隐私泄露，此外攻击者能使用恶意刷机手段进行信息注入，还能利用反编译手段制造恶意代码攻击。

在万物设备皆能入网且海量数据随之产生的物联网时代，为满足存储的负荷和实时响应的需求，以往集中式云计算架构逐渐朝“云+边缘”的模式迈进。在这样新型的计算模式下，网络边缘设备不单只是扮演服务的请求者，同时还承担起部分计算的任务，包括但不限于资料存储、处理、传输和查找等工作。

2.2 区块链基础

2008 年 10 月，一位化名为中本聪（Satoshi Nakamoto）之人发给友人一封电子邮件，邮件里发表了一篇题为《比特币：点对点的电子现金系统》的论文[20]，被公认为区块链的滥觞。该论文中提出一种分布式账本的数据结构，被称为比特币，能实现无需第三方权威机构 CA 的背书，即能实现点对点的线上交易。2009 年 1 月，比特币区块链被称为创世区块的第一个区块诞生。诞生后一周，中本聪转账了 10 枚比特币给密码学家哈尔·芬尼，成为比特币也是区块链历史上第一宗交易记录。

区块链之所以名为区块链，是因它由一个个区块形成链状结构而得名。区块是一组数据打包的总称，这组数据打包着若干笔交易记录，以比特币为例每个区块至多约有两千五至三千笔交易记录。严格来说，一个区块分成区块头和区块体两个部分，交易记录是被包含在区块体里，以比特币为例区块头则包含着版本号、父区块哈希、Merkle 根、时间戳、难度目标以及随机数等 6 个主要元素。

表2-3 以比特币为例区块头的结构

字段	大小	描述
version	4 字节	版本号
pre_block	32 字节	父区块哈希
mrkl_root	32 字节	Merkle 根
time	4 字节	时间戳
bits	4 字节	难度目标
nonce	4 字节	随机数

如何保证被存于区块上的数据不被篡改是一个重要的议题，为让区块链具备不可篡改性和可验证性，引入两个知识点“哈希加密”和“Merkle Tree（默克尔树）”。

2.2.1 哈希加密

哈希加密算法又名散列算法，其效果是将任意长度的字符串明文作为函数 func_hash()的输入，输出的结果为特定长度字符串，这个结果亦被称为哈希值、散列值或明文的消息摘要。满足该算法的函数基本上具有一个特性：单向加密，也就是说仅知明文的摘要甚难得知其对应的明文，仅能用试凑法的方式暴力迭代函数 func_hash()，找出潜在对应的明文。另外，由于任意长度的输入得出特定长度的输出这个特性，可知哈希值的空间远小于明文输入的空间，所以仅有哈希值无法唯一确定对应的明文。

此外，该算法的另一个主要特性是：相似的明文数据作为输入，所得的输出通常差异巨大，也正因为哈希值的跳变剧烈，使该算法被广泛应用于需要确认数据的完整性和不被篡改性中，但凡数据有极为些微的更改，更改前后的哈希值都会显得格外不同。

该算法被广泛用于数字签名以及区块链技术中，用于保证文本的完整性，每个文本的哈希值就相当于该文本的数字指纹。

评价哈希算法的优劣是看其不同的输入映射到同一个输出的概率，发生如此情况又称为碰撞，即

$$func\_hash(k_1) = func\_hash(k_2), \qquad k_1 \neq k_2 \tag{2-2}$$

当这现象出现概率越小说明该算法的性能越好，表示不同的输入越难被碰撞到一起。

广义来说，哈希算法并不是一种算法，而是一种思想，该思想下的实例常见的有 MD5 系列[21]以及 SHA 系列[22]算法。这些算法普遍应用于当今互联网中，跟区块链相关的比特币网络，其区块的哈希值即是对区块头作为输入，迭代两次 SHA_256 哈希加密函数所得到的结果。

2.2.2 Merkle Tree

Merkle Tree 是一套结合哈希加密的树状数据结构，如图 2-2 所示。

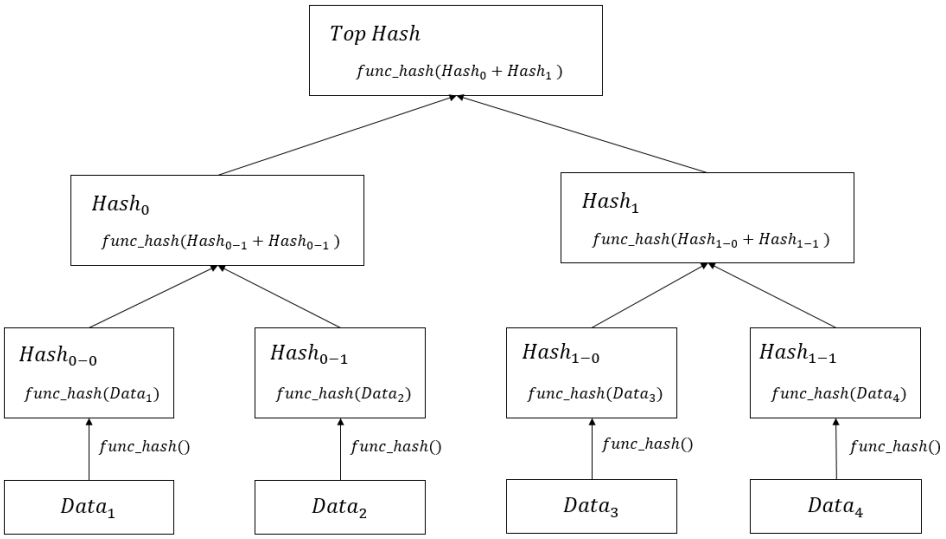

图 2-2 Merkle Tree 示意图

为方便解释，图中仅有四个数据块$Data_i(i = 1,2,\dots,4)$，更多数据块的情形与本例类似。图中，先对源数据分别进行一次哈希加密，得到各自的哈希值，再将相邻的哈希值两两相加，相加后的和再进行一次哈希加密，以此类推……，最后得到根节点的哈希值，此哈希值即为 Merkle 根。

这样树状结构的好处是方便验证。当我们要验证一整个区块的数据是否有被改动时只需

校验根节点的哈希值，因为若根节点下任一子节点或其孙节点有一环数据被更改，根据哈希加密“输入有微小的变化输出会表现得极为不同”的特性，其根节点的哈希值会大大的不同，以此保证了当前区块的完整性。

但显然只保证了当前区块的完整性是不够的，如何确保整个区块链都是不可被篡改的也是一个值得关心的问题。

2.2.3 链状结构与分布式存储

正如前文所述区块链之所以名为区块链是因为每个区块以链状结构所组成。具体来说，每个区块的区块头都包含前一个区块的父哈希值（创世区块除外），该哈希值亦被称为哈希指针。以比特币网络为例，区块的哈希是由当前区块头相加后进行两次 SHA_256 哈希函数所得成。因此，当有一个已经出块的区块 a 中数据出现更动时，a 区块的哈希值就会发生变化，指向它的后一个区块 b 的父哈希值也必随之变化，然而因为区块头包含着父哈希值，所以后一个区块 b 的哈希值也得跟着变化，以此类推,所有在区块 a 后面的区块数据都必须一并被改动才能掩盖数据被篡改的痕迹，这样的机制就增加了数据被篡改的难度。

区块链的数据是存储于 P2P 的分布式网络里，意味着网络上每个节点都会维护自己的一份账本，这也是区块链被称为“分布式账本”的原因。分布式账本的存在有别于以往权威机构的中心化计算模式，使得数据被单一持有方篡改可能性大幅降低。

至此，“链状结构”和“分布式存储”两个特性促成了区块链初步的“不可篡改性”。然而，这无法保证记账的节点不去篡改数据，每笔交易该由谁打包和记录还是个问题；换句话说，整个网络要如何协调记账、怎么让网络里的各个节点相信记账的节点是有公信力的是个问题，大家必须要有 “共识”，这就需要预先设置的“共识机制”来解决这个问题。

2.2.4 共识机制

自古以来人类就如何在一个群体内达成一致性的决策有着广泛的讨论,这类的讨论被统称为共识问题，对于该问题无论社会学科或理工学科都有着长期的研究历史[23]。

在计算机领域里，共识问题主要针对分布式节点如何就一个状态达成统一的讨论；而在区块链中，这个问题主要指网络里各个节点关于记账人选举、区块构造、交易打包、区块验证以及链路更新等一系列指令如何达成一致性决策。借由一套步骤和算法来解决这类问题的，在区块链领域里被称为共识算法或共识机制。

在区块链里，共识机制有若干多种，比如常见的工作量证明（PoW）、权益证明（PoS）以及委托权益证明（DPoS）等，以下分别简明介绍这三种共识机制：

（一）工作量证明（PoW, Proof of Work）

工作量证明最早用于抵御垃圾邮件这个场景里[24]，邮件的发送者必须计算出一道数学难题，借由释出一定算力表示发送邮件的诚意，若欲发送大数量的邮件就必须付出代价高昂的成本，对于滥发广告邮件的人来说显然不符合经济效益，以此减少垃圾邮件的出现频率。

区块链的工作量证明即受上述场景启发，以比特币网络为例，结合前文所述的背景知识区块头、哈希算法以及 Merkle Tree，简单阐述其共识流程：

(1) 节点获取难度目标、交易信息和父哈希值

在比特币网络里，任意有计算能力的节点都可以自由参与网络或退出网络，无需事前注册。在此步骤，节点需要获取当前网络下难度目标 D 和本周期内产生的交易信息，将交易信息以 Merkle Tree 排列得出其根节点哈希值 mrkl_root，并以最长链原则选择最末端区块的哈希值 pre_block,以此作为父哈希值。

(2) 求解满足难度目标的随机数

求解 nonce 以满足

$$H(nonce, merkl\_tree, pre\_block, time) < D \quad (2\text{-}3)$$

其中，H(•)为单向哈希函数，在比特币网络里为 SHA_256(SHA_256(•)),time 为时间戳。该步骤被通俗的称为“挖矿”，仅能凭暴力迭代的方式去试凑，找出满足该公式条件的随机数 nonce。

(3) 区块广播与验证

找到满足工作量证明的节点向网络广播新生成的区块，其他区块收到新生成的区块的信息后开始验证区块以及当中交易的合法性，前者借由计算一次 H(•)查看 nonce 是否满足条件，后者具体为检查当中交易是否存在双花情形。若皆通过，则更新本地的区块链，接着回到第一步，反复该流程；否则，回到第二步，继续挖矿的过程。至于通过验证、找到满足工作量证明的节点将获得系统一定程度上的奖励。

工作量证明算法简单、容易理解，节点间无需交换额外的信息即可达成一致性共识。然而，虽然公平的解决了谁来打包交易的权限，但其也面临 50%算力攻击的问题。若存在恶意节点 E,其拥有整个网络超过一半算力的节点，那么就可能发生双花问题。恶意节点 E 借由先消费代币产生交易 tx，记录在区块 b 上，并在当前区块 b 前进行分叉，计算不含交易 tx 的区块及其之后的区块，让后续的交易打包皆由攻击者 E 产生。那么根据最长链原则，拥有交易 tx 的分支将被注销，如此攻击者 E 实际上消费了代币却没被记账到，等价于多赚了交易 tx 所产生的代币面额。

此外，由于工作量证明容易因为通信延迟或相异节点同时找出满足的随机数而产生分叉，需要等待多个确认方能解决这个问题，造成效率低下；不仅如此，工作量证明在第二步挖矿上浪费过多的计算资源和电力能源，在环保意识日渐抬头的社会里显然是不适宜的。因此后续有人提出了权益证明(PoS)以及改良的委托权益证明(DPoS)

（二）权益证明（PoS, Proof of Stake）

权益证明在工作量证明的基础上，多衡量了节点参与网络工作的时间。主要思想是每个节点获得的代币多了时间的单位，引入“币龄”的概念[25]，把代币的数量乘以持有的时间称为权益。

定义 2.21（权益）权益 R,n 个代币与其分别持有时长$t_i(i = 1,2,\dots,n)$乘积的总和，即

$$R = N \times T \quad (2\text{-}4)$$

其中，N 为一个元素全为 1 的$1 \times n$的矩阵，T 为元素为$t_i(i = 1,2,\dots,n)$的$n \times 1$的矩阵。

当一个节点持有的代币越多且持有时间越长，其解出上述工作量证明流程第二步随机数的难度也就越低，即权益 R 与记账权的获得难度成反比。

权益证明是工作量证明的升级版，优点是能加快找随机数的速度，概率上由于挖矿难度降低，出块的速率比起采用工作量证明的网络加快许多。缺点是仍无法缺少挖矿的环节，有挖矿就代表对能源浪费的情况依然会发生。PoS 引入权益,让网络贡献愈多的节点可以愈容易解出满足式 2-2 的解，使得总体而言在概率上降低了平均单位出块所消耗的能源量，比起工作量证明，效能有一定程度的提升。

除了无法根除资源浪费的问题外，权益证明机制也可能产生“富者更富”的情况，因考量币龄的缘故，使得越先入局的节点优势越大，这也将让整个网络若存在元老节点具有恶意的意图，势必大大危害整个网络的隐患，形成“首富作恶问题”。

（三）委托权益证明（DPoS, Delegate of Proof of Stake）

委托权益证明,也被称为委托股权证明、股份授权证明、代理权益证明或受托人证明,采用类似董事会投票的机制，网络中每个节点都可参与投票选举和有机会被选为见证人（witness）,投票的权重与每个用户持有的代币数量成正相关，获票数前 n 位的的节点将

成为见证人，系统会为这些见证人随机排列，形成一种类似于执行委员会的组织，负责在新的一轮周期内轮流产生和验证新的区块。通常一名见证人生成区块的时间不多，若发生脱机等离线状态，区块生成权限将自动交给下一个顺位的见证人处理。

若有见证人节点存在叛徒的行为，由于新区块生成时需要网络里 2/3 的股权同意，遂难以得逞，且该举动将不被其他选民们所信任，进而无法被选为下一轮的见证人。委托权益证明即是以此手段降低被攻击的可能。

比起工作量证明和权益证明，委托权益证明被认为是效益是最高的。通常只有在选举过程中需要等待确认，待委员会成立后见证人即可按规则高效的执行任务，因此每秒处理事务的次数 TPS 也是在三者中最高的。

通常来说委托权益证明，具有以下几处优点：

(1) 高扩展性：

由于委托权益证明并未要求设备拥有高计算的性能，因此对于算力较低的用户具有更好的普适性。

(2) 交易速度快：

缺少了挖矿争夺记账权的流程，采用 DPoS 的网络比起采用 PoW 或 PoS 的网络出块速率高上许多。

(3) 普惠性：

由于委托权益证明拜托了依赖于算力的竞争，使得更多低端、算力低下的设备也能参与到网络里，投出自己青睐的节点，让网络的参与者更加多元。

虽然 DPoS 似乎解决了浪费计算资源的问题，但也有其缺点：

(1) 偏中心化：

虽然 DPoS 解决了持有代币越久越容易造成垄断的问题，但是被选为见证人的节点在付出计算资源完成数据打包和验证后将会获得一定的奖励，同时若整个网络内参与投票的节点不多，即选民投票意愿低落，同样容易造成少数几个热衷参与投票的节点垄断整个网络，让整个网络变得不可信任。因此与 PoW 和 PoS 网络相比，采用 DPoS 网络更具有中心化的倾向。

(2) 投票积极性低：

在 DPoS 实际应用中，持股人参与投票并没有获得特别的好处，导致参与投票的意愿低下。然而，DPoS 的安全性很大程度上依赖于投票的积极性，若选民越多元，越难以让单一或少数节点垄断整个网络，但要维持高选民的参与度仍是一个问题。

虽然 DPoS 面临偏中心化的质疑，在记账节点较少的情况下这个质疑也越显著，但可以增加至少 50%投票人同意、认为该轮循坏“无关键中心化”[3]的环节来消弭；另外，只要参与选举的节点越多且越多元，就越能保障整个网络不被若干节点控制的问题。

总结这三种常见的共识机制，分优缺点来概括：

从缺点层面来看，工作量证明需要试凑迭代，不断调整 nonce 计算哈希值，故最为浪费计算资源；委托权益证明有少数代表来完成记账，故最具有中心化倾向。

从优点层面来看，工作量证明竞争的是算力，只要算力足够高任何节点无论参与时间早晚，都有机会挖到矿，因此被视为最公平的原则；而委托权益证明由于引入节点投票衍生出议会投票的概念，故最具有民主共治的思想；也因为委托权益证明，选出代理节点后，即可执行一轮 n 个出块，状态确认的次数是三者中最少的，出块速率为三者中最高的。

[3] 在中文语境里“Decentralized”一词常被翻译为“去中心化”，但在这里选出了见证人作为代表，由少数人来执行特定的事，有一定程度中心化倾向，并非完全“去中心化”的，所以应用“无关键中心化”更为恰当。关于 Decentralized 的讨论可参见
https://www.samsonhoi.com/326/blockchain_decentralised_p2p_concept

三种共识机制各有其优缺点，并没有一个绝对完美的方案。值得注意的是，区块链的共识机制远不止所述的三种，比如近几年实用拜占庭容错（PBFT）也越来越常见,其保证了恶意节点[4]若少于总节点数的三分之一整个网络即可顺利运行，其比起前文提及的三种共识算法一致性最强[26]。碍于本文篇幅限制，可参见更多关于共识机制的文献。

2.2.5 区块链基础整理

在 2.2 结，我们介绍了一些区块链的基本知识，从区块链的背景出发，分别阐述了区块的基本结构、哈希加密、Merkle Tree、链状结构以及三种常见的共识机制。区块链作为一种新兴技术，无论是产业界还是学术界都还在探索的阶段，在普遍大众的认知里还没有突破性的改变。值得注意的是，也因区块链正值探索阶段，2.2 节介绍的关于区块链的基本知识仅是从一个普遍的角度出发，并非所有区块链的实例都遵从以上特性或规则。举例来说，许多文献将区块链的区块头定义为表 2-3 的格式，但事实上不是所有区块链都必须遵从这样的表现，表 2-3 只不过是区块链的典型实例比特币所定义的，并非每个区块链皆需如此。

尽管区块链各个实例的表现各异，但普遍上皆会具有以下特性：

(1) 去中心化：

这是区块链有别于其他数据库的主要特点，传统计算架构有一个中心服务器做数据处理、运算和存储，然而在区块链里这些操作皆是分布式的，由一组网络的节点来共同决定和处理的。

(2) 不可篡改性：

借由分布式存储和共识机制来确保存于链上的数据是完整且不可被更改的，这有别于存储于一个中心化数据库里的计算模式，有效避免了数据存储或运行的算法不透明导致被黑箱处理的可能。

(3) 可追溯性：

由于链上的数据可以由后续生成区块来验证，又由于各个节点拥有账本的副本，所以存于链上的数据都可以被追踪或被回溯，是为区块链的可追溯性。

区块链这种具有去中心化、不可篡改性以及可追溯性的新型技术已开始应用于分布式金融（DeFi）[27]、以 IPFS 为代表的分布式文件系统[28]、公益项目募款[29]和产品溯源[30]等，而本文聚焦在物联网里边缘层内的隐私保护领域，凭借区块链的去中心化避免单点攻击以降低数据外泄的风险，依靠不可篡改性保障数据的完整性和可信任性，最后借由可追溯性记录所有访问的历史以评估当前系统状态再做出决策。后文中枢区块链的基本用途皆由这三大特性出发，基于此尽可能满足用户对隐私保护的要求和期待。

2.3 同态加密技术

区块链技术底层依赖于密码学原理，在讨论区块链技术时离不开密码学的贡献。本小结聚焦在一个能保障用户隐私的同态加密技术上，其为本文确保数据不被第三方泄露又能使该数据被计算、处理或利用的重要手段，期望能深入浅出剖析当中数学原理，在后续章节运用时知其所以然。

若一未加密的明文在跨域通道上通信，容易遭受中间人攻击，使数据变得不可信；若一明文 m 经密钥 k 加密后得到密文 c，发送者将密文 c 传送给接收者，接收者用同一密钥 k 对密文 c 解密，这种采取单钥加解密的技术被称为对称加密技术，常见的算法实例如 AES；若一明文 c 使用公钥 pk 加密得到密文 c,发送者将密文 c 传送给接收者，接收者用私钥 sk 对密文 c 解密，这种加解密用到不同密钥的技术被称为非对称加密技术。

宏观来看，仅从密钥曝露的范围大小来论安全性，由于对称加密技术的密钥曝露在接收

[4] 也被称为拜占庭节点。

和发送两端，而非对称加密技术用于解密的私钥仅存于也必须存于接收方单端里，基于越低层度曝露安全性越高的思想，这使得非对称加密技术普遍被认为安全性高于对称加密技术；再加上非对称加密算法对于相同明文加密，由于随机数的作用所得的密文也会不同， 这一特性被称为“语义安全（Semantic security)”，可降低选择明文攻击[5]的发生，更大程度提升了非对称加密技术的安全性。

在非对称加密的基础上,我们引入同态加密的概念,设法让一组数据是被加密的状态,对其运算后等效于对明文空间的运算。

2.3.1 同态加密基础

若现有一用户 Alice,拥有一明文 m，借由公钥 pk 经加密操作 ENCRYPT 得到密文 c,即

$$c = ENCRYPT(pk, m) \tag{2-5}$$

用户 Alice 将 c 传给计算方 Bob,计算方 Bob 对 c 进行一系列计算，得到c'。将这系列计算记为操作F'，有

$$c' = F'(c) \tag{2-6}$$

Bob 将c'回传给 Alice,Alice 再用私钥 sk 对其解密,得到 result

$$result = DECRYPT(c') \tag{2-7}$$

若对任意$m \in S$,S 为明文域的集合,存在操作F'的等价操作F,使得

$$result = F(m) \tag{2-8}$$

我们称式（2-5）至式（2-8）这段过程满足同态加密这种加密形式。

从这段过程可以看到，计算方 Bob 并不知明文为何，他仅仅是对密文进行了计算，然后将计算的结果返回给用户 Alice，Alice 再对其解密得到她真正想要的结果。整个过程 Bob 都无从得知明文，充分保护了 Alice 的隐私。

同态加密这项技术涉及明文空间和密文空间的映射，严谨来说，Rivest,Adleman 和 Dertouzos 等人对同态加密做出如下的定义[31]。

定义 2.31 S 是明文空间的集合，S'是与 S 具有相同基数的另一个集合；$\emptyset: S' \to S$是双射，被是为解密函数,与之对应的是$\emptyset^{-1}: S \to S'$，被是为加密函数。现用如下代数系统表示用户端对明文的操作：

$$U = < S: f_1, \dots, f_k; p_1, \dots, p_l; s_1, \dots, s_m > \tag{2-9}$$

其中，f_i为操作中所需的函数，p_i为操作中所用到的谓词，s_i为区分的常数。对 U 的逆运算，即计算端的代数系统，可以表示如下：

$$U' = < S': f'_1, \dots, f'_k; p'_1, \dots, p'_l; s'_1, \dots, s'_m > \tag{2-10}$$

映射$\emptyset$若满足以下条件：

（1）对于所有$i(1 \le i \le k)$，$a_1, a_2, \dots, a_{g_i} \in S'$,存在$c \in S'$,满足

$$f'_i(a_1, a_2, \dots, a_{g_i}) = c \ \Rightarrow \ f_i\left(\emptyset(a_1), \emptyset(a_2), \dots, \emptyset\left(a_{g_i}\right)\right) = \emptyset(c) \tag{2-11}$$

（2）对于所有$i(1 \le i \le l)$，$a_1, a_2, \dots, a_{h_i} \in S'$，有

[5] chosen-plaintext attacks

$$p'_i(a_1, a_2, \dots, a_{h_i}) \equiv p_i(\emptyset(a_1), \emptyset(a_2), \dots, \emptyset(a_{h_i})) \tag{2-12}$$

（3）对于所有$i(1 \le i \le m)$，有

$$\emptyset(s'_i) = s_i \tag{2-13}$$

则被称为秘密同态。

在现实中并非任何对数据的数学运算都可以拥有这样的映射关系，对于仅满足特定的代数运算同态加密系统被称为单同态[32]。

定义 2.32 对于加密函数 E、解密函数 D、明文空间 P 以及密文空间 C 所组成的密码系统$S(E, D, P, C), x \in P, y \in P, E(x) \in C, E(y) \in C$，若满足

$$D\big(E(x) \otimes E(y)\big) = xy \tag{2-14}$$

其中，$\otimes$为密文空间 P 里的任意操作，则称该系统 S 满足乘法同态。若其还是单同态的，则称该密码系统为乘法同态加密系统。

定义 2.33 对于加密函数 E、解密函数 D、明文空间 P 以及密文空间 C 所组成的密码系统$S(E, D, P, C), x \in P, y \in P, E(x) \in C, E(y) \in C$，若满足

$$D\big(E(x) \oplus E(y)\big) = x + y \tag{2-15}$$

其中，$\oplus$为密文空间 P 里的任意操作，则称该系统 S 满足加法同态。若其还是单同态的，则称该密码系统为加法同态加密系统。

本文后续段落使用 Paillier 非对称加密算法进行实验，以下对 Paillier 算法进行介绍。

2.3.2 Paillier 同态加密算法

Paillier 同态加密算法源自学者 Pascal Paillier 于 1999 年发布的论文[33]，该系统建立在判断$\mathbb{Z}_n^*$上求解关于 n 次剩余的困难问题。

定义 2.34（Carmichael's function $\lambda(n)$）对于任意一个剩余类$a \in \mathbb{Z}_n^*$, $a^y = 1 \mod n$都成立且y是满足该式子中最小的数，则称$\lambda(n) = y$为 Carmichael's function

其中，$\mathbb{Z}_n^*$表示与 n 互质的剩余类集合，比如$\mathbb{Z}_{12}^* = \{[1]_{12}, [5]_{12}, [7]_{12}, [11]_{12}\}$, 当中$[x]_{12}$表示模 12 余数为$x$的集合。

可以证明，已知两个素数 p 和 q, 对于合数$n = pq$，有

$$\lambda(n) = \mathrm{lcm}(p - 1, q - 1) \tag{2-16}$$

其中$\mathrm{lcm}(x, y)$表示取 x 和 y 的最小公倍数。我们简记$\lambda(n)$为λ。

还可以证明，同样已知两个大素数 p 和 q, 对于合数$n = pq$, 有

$$\varphi(n) = (p - 1)(q - 1) \tag{2-17}$$

其中$\varphi(x)$为欧拉函数，即小于等于 x 且与 x 互质的个数。

同样不难证明，$\forall \omega \in \mathbb{Z}_{n^2}^*$，有

$$\omega^{\lambda} = 1 \mod n \tag{2-18}$$

$$\omega^{n\lambda} = 1 \mod n^2 \tag{2-19}$$

定义 2.35（n 次剩余）若存在一个整数$y \in \mathbb{Z}_{n^2}^*$, 使得

$$z = y^n \mod n^2 \tag{2-20}$$

我们称 z 为模n^2的一个 n 次剩余。

定义 2.36（函数 $\mathcal{E}_g$）已知$g \in \mathbb{Z}_{n^2}^*$, 定义整数函数$\mathcal{E}_g$

$$\mathcal{E}_g: \begin{cases} \mathbb{Z}_n \times \mathbb{Z}_n^* \mapsto \mathbb{Z}_{n^2}^* \\ (x,y) \mapsto g^x y^n \mod n^2 \end{cases} \tag{2-21}$$

不难证明，若 g 的次数是 n 的正整数倍，则$\mathcal{E}_g$是一一映射的。当 g 的次数为$n\alpha, \alpha = 1,2,\dots,\lambda$时，以元素集$\mathcal{B}$来表示满足此情况的 g 的集合，$\mathcal{B} \subset \mathbb{Z}_{n^2}^*$。

定义 2.37（关于 g 的ω的 n 次剩余类，$[[\omega]]_g$）若存在$y \in \mathbb{Z}_n^2$, 使得

$$\mathcal{E}_g(x,y) = \omega \tag{2-22}$$

我们称式中唯一的$x \in \mathbb{Z}_n$为关于 g 的ω的 n 次剩余类, 记作$[[\omega]]_g$。
值得注意的是，$[[\omega]]_g = 0$当且仅当ω是模n^2的一个 n 次剩余，因为此时

$$\omega = g^0 y^n \mod n^2 = y^n \mod n^2 \tag{2-23}$$

现在，令

$$\omega_1 = g^{x_1} {y_1}^n \mod n^2 \tag{2-24}$$

$$\omega_2 = g^{x_2} {y_2}^n \mod n^2 \tag{2-25}$$

亦即$x_1 = [[\omega_1]]_g$和$x_2 = [[\omega_2]]_g$。
将ω_1与ω_2相乘，有

$$\omega_1\omega_2 = g^{x_1+x_2}(y_1y_2)^n \mod n^2 \tag{2-26}$$

从此，不难推出对于$\forall \omega_1, \omega_2 \in \mathbb{Z}_{n^2}^*$

$$[[\omega_1\omega_2]]_g = [[\omega_1]]_g + [[\omega_2]]_g \tag{2-27}$$

其中，$g \in \mathcal{B}$，也就是函数$\omega \mapsto [[\omega]]_g$把群$(\mathbb{Z}_{n^2}^*, \times)$同态映射到了群$(\mathbb{Z}_n, +)$。
掌握以上基础，可以构造 Paillier 加密系统如下：

一、密钥生成
选取两个大素数 p 和 q，令 n=pq，再随机选取$g \in \mathcal{B}$。可借由

$$\gcd\left(L\left(g^\lambda \mod n^2\right), n\right) = 1 \tag{2-28}$$

来检查g是否为有效的选取，其中$L(u) = \frac{u-1}{n}$

式（2-28）等价于存在$L\left(g^\lambda \mod n^2\right)$的逆元

$$\mu = L\left(g^\lambda \mod n^2\right)^{-1} \mod n \tag{2-29}$$

然后将(n, g)视为公钥，(p, q)视为私钥，形成 Paillier 加密系统的一对钥匙。

二、加密过程

对于明文$m \in \mathbb{Z}_n$且$m < n$,随意选取$r \in \mathbb{Z}_n^*$,计算

$$c = g^m r^n \mod n^2 \tag{2-30}$$

视 c 为密文。

三、解密过程

对于密文$c \in \mathbb{Z}_{n^2}^*$且$c < n^2$,计算

$$m = \frac{L(c^\lambda \mod n^2)}{L(g^\lambda \mod n^2)} \mod n \tag{2-31}$$

即可解出明文。

2.3.3 Paillier 同态加密算法同态性分析

根据定理 2.32，加密系统满足式（2-15），$D\big(E(x)\oplus E(y)\big) = x + y$，我们称该密码系统为加法同态的。观察式(2-26)，$\omega_1\omega_2 = g^{x_1+x_2}(y_1y_2)^n \mod n^2$，将$E(x) = \omega_1$，$E(y) = \omega_2$，$x = x_1$，$y = x_2$，显然满则$D\big(E(x)\oplus E(y)\big) = x + y$，其中$\oplus$为密文空间$\omega_1$与$\omega_2$的乘法，$+$为明文空间$x_1$与$x_2$的加法，故我们称 Paillier 同态加密算法满足加法同态性。

2.4 本章小结

本章首先阐述了边缘计算的概念，同时对边缘计算架构各层中的安全隐患进行分析；其次，探讨区块链技术的背景、原理和特性，从中本聪 2008 年发布的论文以及关于比特币网络的诞生背景开始描述，阐述了区块链底层所需的哈希加密技术、Merkle Tree、链式结构、分布式存储以及共识机制，在 2-2 节最后整理了区块链具备的去中心化、不可篡改以及可追溯性等特性，总结了区块链目前常见的使用场景；在本章的第三节，介绍了一种保证用户隐私的技术“同态加密”，阐述了它的特性和定义，并且详细的以 Paillier 同态加密算法为例，结合数学公式和推导说明它的由来，最后引出该算法关于密钥生成、加密过程以及解密过程的方法，并分析了其加法同态性的重要特性。

第三章 一种基于区块链技术的隐私保护方案

区块链技术在处理不可信任群体之间的协作具有很大的优势，同态加密算法作为区块链相关的热门技术在隐私保护和数据可计算性上取得了良好的平衡，因此本章设法借助两者的优势，在不可信任的环境里注入可信任的结构区块链，在通信上用非对称加密为数据加密，使用同态加密技术对密文进行计算，再由可信任的节点对数据解密，这样的方案非常适用于边缘环境不可信群体之间协作的问题，解决用户疑虑自身数据所携带的隐私被泄露而不愿让数据被操作或分析的困难。

本章在第二章关于区块链和同态加密的基础上，提出了在数据统计场景里基于区块链、终端层和边缘服务器层的三点架构结合 Paillier 同态的用户隐私保护模型。第一节，研究整个方案的应用场景和欲解决的问题；第二节，深入研究方案的各个步骤，从数据分析者的场景出发，最终传送数据分析者统计数据作结；第三节，对方案进行理论分析。

3.1 方案背景和模型

现如今许多的应用程序常驻系统背景后台，在用户无感知的情况下，收集使用者数据，包括但不限于打开时间戳、前台使用时长、位置信息、网络状态、设备品牌和型号甚至是设备唯一识别号 IMEI 等。多半用户无法选择不提供这些数据，否则就面临无法使用该应用的处境。然而，应用开发商却不见得渴望获得单一个体的数据，对开发商而言，真正实用的应为统计数据而非单位数据，因为前者方能帮助开发商了解应用的整体情况。既然如此，为了避免将所有个体数据皆传送给应用开发商，在边缘环境下处理并计算各单元个体数据进而得出一个统计值的过程就显得格外重要，然而边缘环境被常认为是一个不可信任的环境，因此这个问题涉及边缘环境下各群体如何信任地协作以确保用户隐私，同时让用户相信这个模式是可信的也至关重要。

3.1.1 方案背景和缘由

假设现有一位代表着应用开发商的数据分析者，其具有一个数据分析的需求，可能是求月活跃用户数、日活跃用户数或者各场景的转化率等常见的运营数据统计场景，近几年各个应用或者设备的一周使用时长也成为一种衡量用户依赖应用或设备的指标，这里不妨假设该名数据分析者欲得知该应用开发商服务的用户一周内使用该应用的时长。

为了避免让数据分析者得知单位个体的数据，还需设置一个分析范围，比如将这次数据分析限定在某个社区作统计。同时，该数据分析会对应一套统计算法，以一周使用时长为例所需要的算法即是求在该范围内所有个体的使用时长之平均。

然而，将分析的任务交由谁核可和计算是一大问题。若仅由集中式的单一节点去承担这些任务，就容易面临单点攻击的危害；同时，若仅从某个节点来完成所有的任务，还会面临该节点泄露用户数据的疑虑。因此，该场景里需要一个可信的节点去处理任务的管理和分发，而区块链天生即具备可信任的特质。

区块链拥有去中心化抗单点攻击的能力，同时具备可追溯性和不可篡改性让分析调用的记录能被详实的记录下来，再结合同态加密技术解决各节点隐私数据计算的困难，让数据兼顾保密性和可计算性，保障了用户的隐私。受此启发，结合区块链技术和同态加密技术，提出一种让数据的明文状态锁于用户自身所在的终端层内、计算的过程留于边缘计算层内、而最终解密结果交由区块链去中心化节点处理的架构，呈现出一个三点互有联系却又互相隔离的系统，设法让边缘环境下的用户信任环境的安全，进而信任自身隐私不会遭到泄露。下文给出更细致的架构和方案实践的步骤。

3.1.2 方案模型

将 3.1.1 小结所述的应用场景更进一步抽象，更为一般地，以 A 表示数据分析者，数据分析以 analysis 为记，analysis 所对应的算法写成 analysis_func，某次分析的统计范围定为$\mathrm{range}=\{\mathrm{CB},\mathrm{Sr},\mathrm{D}\}$，将 range 下 n 个终端用户的集合标记为 D，$\mathrm{D}=\{D_1,D_2,\ldots,D_n\}$，并设该 range 下的边缘环境里有 q 个边缘服务器$\mathrm{Sr}=\{Sr_1,Sr_2,\ldots,Sr_q\}$，并有一个中枢区块链负责数据外包的访问控制以及协调终端设备和边缘服务器层的运作，于是有以下关系模型。

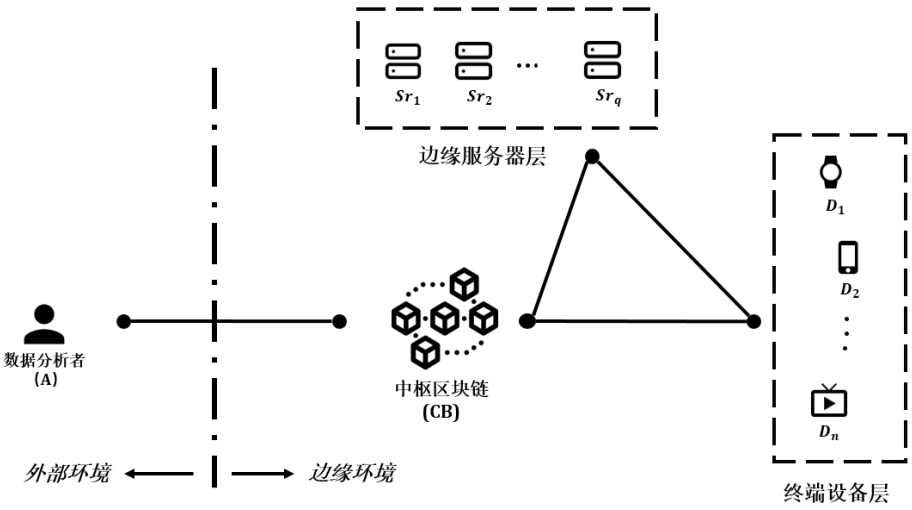

图 3-1 边缘计算下基于区块链技术的隐私保护模型架构图

从图 3-1 可见，将数据分析所在的位置定义为外部环境；相对的，中枢区块链、终端设备层以及边缘服务器层所在的环境称为边缘环境。本文即在研究如何让后者充分协作并且能保护用户的隐私。

在本方案中，所使用到的部分符号和含义如表 3-1 所示。

表3-1 方案部分符号及其含义

符号	说明
range	限定边缘环境的范围，比如一个住宅社区内
analysis	数据分析者发起的某次数据分析
analysis_func	analysis 所对应的计算
A	数据分析者
CB	中枢区块链
Sr	边缘服务器的集合
D	终端设备的集合
n	终端设备的总数
n'	接受此次数据分析的终端设备总数，$\mathrm{n}'\leq\mathrm{n}$
q	边缘服务器的总数
D_i	第 i 个终端设备
Sr_j	第 j 个边缘服务器
S	所需的源数据名称的集合，$\mathrm{S}=\{s_1,s_2,\ldots,s_m\}$
m	所需的源数据名称的总数

P_i	第 i 个终端设备取出自身与 S 对应的明文集合，$P_i = \{p_{i1}, p_{i2}, \dots, p_{im}\}$
C_i	第 i 个终端设备将P_i非对称加密后的密文集合，$C_i = \{c_{i1}, c_{i2}, \dots, c_{im}\}$
he_key	中枢区块链 CB 生成的同态加密密钥$he_key = \{he_pk, he_sk\}$
he_pk	同态加密所需的公钥
he_sk	同态加密所需的私钥

3.2 方案步骤

从 3.1.2 小结所述的边缘计算下基于区块链的三点架构，实现一次数据分析的简要步骤：

（1）**预分析请求**：数据分析者 A 向中枢区块链 CB 发起预分析请求，等待 CB 的响应。

（2）**访问控制**：CB 对 A 的数据分析请求进行判断，若许可则返回权鉴。

（3）**正式分析请求**：A 凭权鉴向 CB 发起正式分析请求。

（4）**中枢区块链验证与数据分发**：CB 验证正式分析请求，通过后记录在链上，并对终端设备D_i下发通知。

（5）**D_i验证与屏蔽名单检验**：每个D_i对数据分析意图进行验证和屏蔽名单检测。

（6）**D_i对源数据加密**：D_i使用来自 CB 的公钥对源数据进行加密。

（7）**D_i发送密文给Sr_j**：D_i发送经同态加密的数据给边缘服务器Sr_j。

（8）**Sr_j验证请求合法性**：Sr_j向 CB 验证这次分析请求是否合法。

（9）**Sr_j展开边缘计算**：Sr_j根据分析函数在密文域展开计算。

（10）**Sr_j返回计算值，CB 对其解密**：Sr_j将密文域的计算结果返回给 CB，再交由 CB 解密。

（11）**CB 发送结果值给 A**：CB 将解密后的值返回给 A。

从第 4 步开始至第 10 步为边缘计算的范围，这七步梳理了中枢区块链、终端设备群以及边缘服务器群之间的关系和协作方式，为保障环境数据安全和用户隐私的关键。

3.2.1 方案详细步骤

以下完整给出整个方案各步骤的流程：

步骤一：**预分析请求**

图 3-2　步骤一的预分析请求示意图

在第一步里，数据分析者发送分析者唯一标识（analystID）、当前设备唯一标识(deviceInfo)、时间戳(stamp)、随机字符串（nonce_str）、数字签名以及分析包裹等一系列数据给中枢区块链(CB)。其中，分析包裹为分析标题、分析描述、分析序号(analysis_no)、分析函数（analysis_func）的统称，前两者用于告知边缘环境当前数据分析的用途，分析函数则起到如何为之展开计算的作用。

另外，数字签名用于使 CB 验证数据的完整性，其签名结果为

Hash(分析者 ID,，分析序号，设备 ID，随机字符串，时间戳，密钥)　(3-1)

其中 Hash()为任意哈希加密算法。

步骤二：**访问控制**

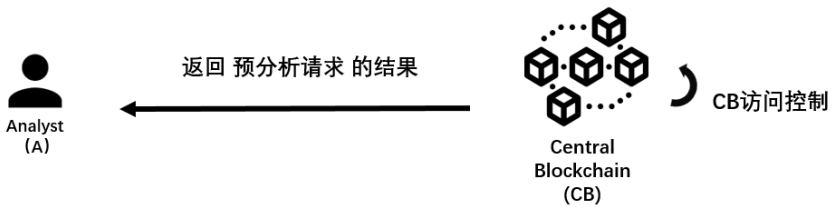

图 3-3　步骤二：CB 对预分析请求访问控制

中枢区块链（CB）收到来自数据分析者的预分析请求时，先校验签名，再根据链上记录判断是否通过该次数据分析的请求。若通过则返回权鉴 preAnalysisToken，该权鉴视为数据分析者在下一步正式分析请求的通行证。

步骤三：**正式分析请求**

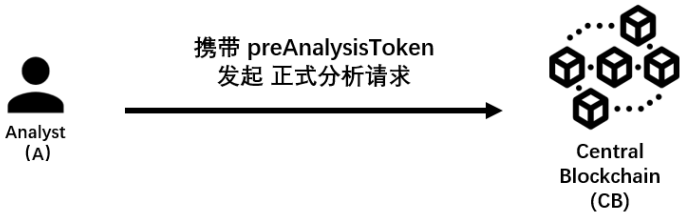

图 3-4　步骤三，正式分析请求

若 A 收到权鉴后，即可向 CB 发起正式的分析请求。在正式分析请求中，数据包必须包含权鉴 preAnalysisToken，并且分析包裹必须与步骤一相同。

步骤四：**中枢区块链验证与数据分发**

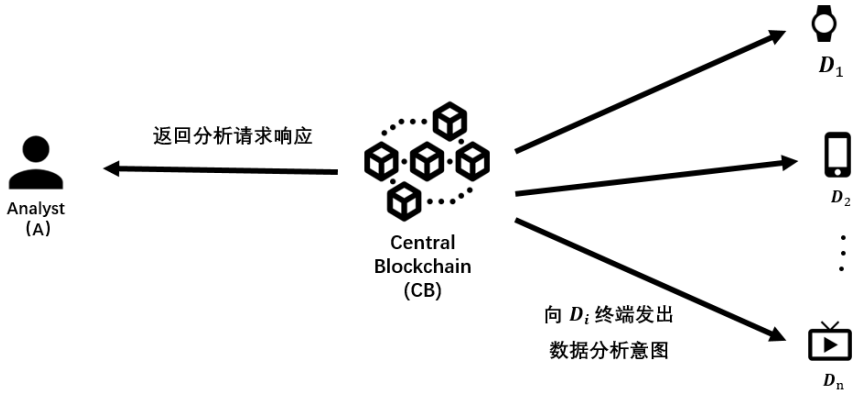

图 3-5　步骤四，CB 对正式分析请求校验并作数据分发

CB 收到正式分析请求后，先对请求进行校验；若校验通过，CB 选举一个领导节点 CBL。CBL 返回分析请求响应给 A，并开始生成 Paillier 同态加密公私钥对：

$$\text{he\_key} = \{\text{he\_pk}, \text{he\_sk}\} = \text{KeyGen}() \tag{3-2}$$

其中，he_pk为同态加密公钥，可对外公开；he_sk为同态加密密钥保留在 CBL 内，不可对外泄露。

当同态加密公私钥对生成完毕，CBL 即可向范围内的终端设备发出数据分析意图。该意图里，能取出数据分析所需要的源数据名称

$$\text{S} = \{s_1, s_2, \ldots, s_m\} \tag{3-3}$$

其中 m 为需要使用到的源数据个数。

意图中，包含但不限于终端设备 ID、分析者 ID、CBL 的 ID、分析序号、分析包裹、对源数据加密的公钥he_pk、经终端设备公钥执行非对称加密的源数据名称 enS、源数据名称的签名 signS、分析者所代表的公司或机构名称以及当前时间戳等信息。

步骤五：**D_i验证与屏蔽名单检验**

终端设备收到数据分析的意图后，用自身设备私钥解密 enS 得S′，并对S′进行数据加密对该意图进行数字签名的校验，以哈希函数对S′加密得到signS′，比对其与 signS 是否一致，即可得知数据的有效性。

同时，对意图里的分析者 ID 进行屏蔽名单检测，若分析者 ID 出现在用户的屏蔽名单里，则拒绝此次数据分析。拒绝的终端设备个数以 r 为记，则参与数据分析的终端设备个数

$$\text{n}' = \text{n} - \text{r} \tag{3-4}$$

步骤六：**D_i对源数据加密**

意图通过步骤五的验证后，终端设备D_i即可取出自身敏感数据

$$P_i = \{p_{i1}, p_{i2}, \ldots, p_{im}\} \tag{3-5}$$

以he_pk对源数据P_i进行加密，得到密文

$$C_i = \{c_{i1}, c_{i2}, \ldots, c_{im}\} \tag{3-6}$$

其中，$c_{ij} = Encrypt(p_{ij}, he\_pk)$，$Encrypt()$为式(2-30)所述的 Paillier 同态加密系统的加密函数。

步骤七：**D_i发送密文给Sr_j**

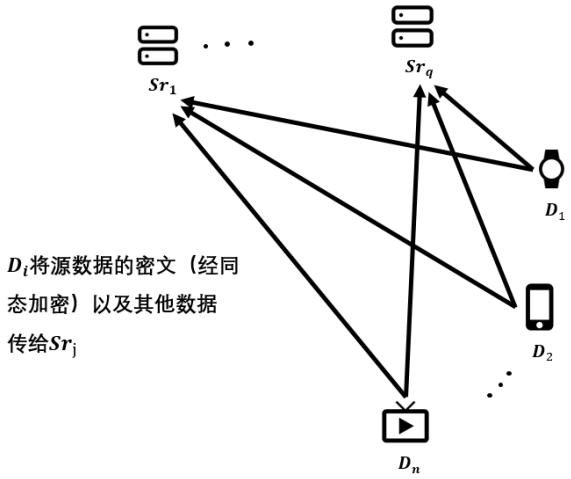

图 3-6　步骤七，D_i发送密文给边缘服务器群

在这一步，D_i将密文数据C_i同分析序列号传给各个边缘服务器$\mathrm{Sr} = \{Sr_1, Sr_2, \ldots, Sr_q\}$

步骤八：**Sr_j验证请求合法性**

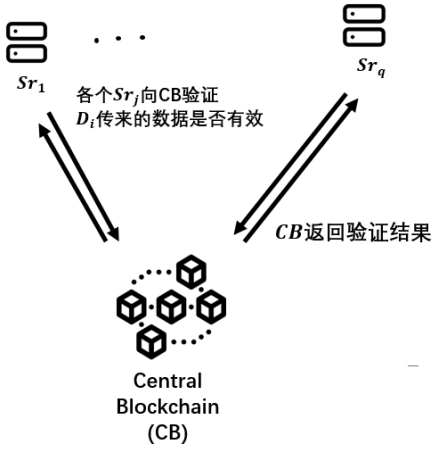

图 3-7　步骤八，Sr_j验证请求合法性

边缘服务器层收到来自终端层提交的密文后，向 CB 询求该轮分析是否存在，若存在方可在下一步骤展开计算。

步骤九：**Sr_j展开边缘计算**

得知数据分析为有效者后，边缘服务器即可在密文域上展开计算。计算结果可以如下表示：

$$res\_c = Evaluate(C) \tag{3-7}$$

其中，$\mathrm{C} = \{C_1, C_2, \ldots, C_n,\}$为来自终端层的源数据密文集合，$Evaluate()$为 Paillier 同态加密系统在密文空间上的计算过程，从式（2-26）可知是在密文空间上的乘操作。

步骤十：**Sr_j返回计算值，CBL 对其解密**

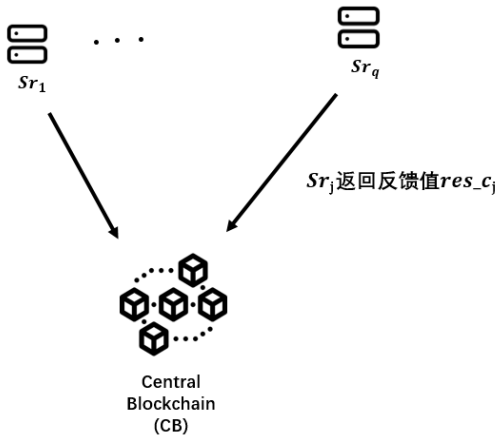

图 3-8　步骤八，Sr_j验证请求合法性

CBL 收到来自边缘服务器层的反馈值$res_c \in \mathbb{C}$,其中 $\mathbb{C}$ 为密文空间，CBL 即可开始对密文解密：

$$res\_p = Decrypt(res\_c, he\_sk) \tag{3-8}$$

其中，Decrypt()为 Paillier 同态加密系统的解密过程，同式(2-30)；he_sk 为式(3-2)所生成的私钥。

步骤十一：**CBL 返回计算结果至 A**

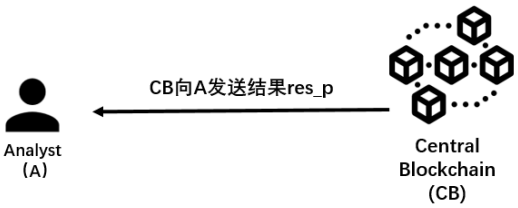

图 3-9　步骤十一，中枢区块链返回计算结果

最后一步，中枢区块链以 CBL 为代表，返回计算结果 res_p 给数据分析者 A，至此即完成一轮数据分析的过程。

3.3 理论分析

根据 3.1 和 3.2 节所提出的方案，若将之聚焦在边缘环境各单元的协作上，一个以边缘环境为主体的方案简要模型，如下图所示。

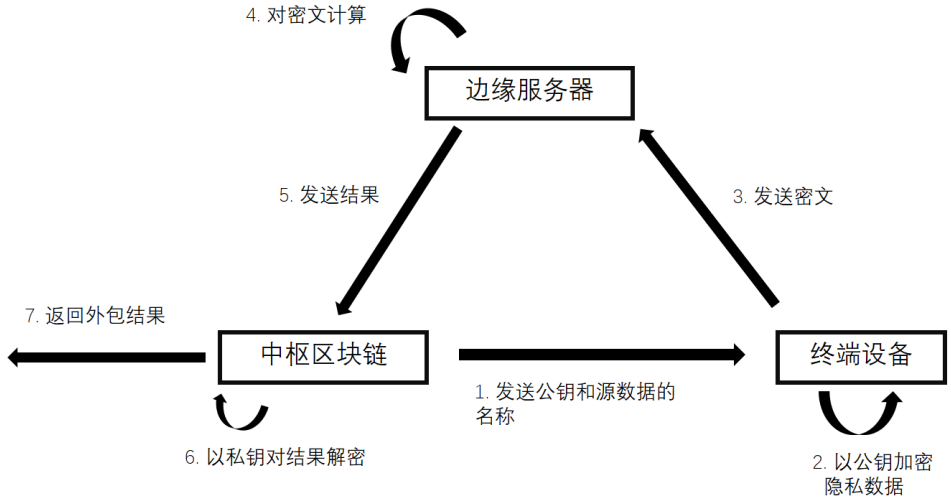

图 3-10　聚焦在边缘环境上的三点结构关系图

在数据安全和隐私保护上，该结构具有避免单点攻击、避免外包对象过度请求以及终端数据的机密性三项特点：

（1）避免单点攻击

避免单点攻击又可分成区块链网络对边缘环境内部的攻击和外部环境对边缘环境的攻击。关于前者，由于分发指令和验证请求的单元为中枢区块链，由 2.2.5 节知区块链具有去中心化的特征，有效阻隔了单点攻击发生的可能。即使区块链上任意节点作恶，会使节点在共识机制上不被认可，进而避免区块链网络上作恶的可能；关于后者，由于终端设备层和边缘服务器层的请求验证皆向中枢区块链进行交互以实现，因此外部攻击者势必要取得中枢区块链的主导权方能达成目的，然而区块链网络去中心化的结构就使得这样的攻击变为极其困难，只有在共识机制上证明的攻击者数量 f 大于模型设定的 F,才能完成这样的攻击。

（2）避免数据外包对象过度请求

数据分析者每次请求数据分析时，所有访问记录都会如实被记账在区块链上，若请求过多，企图用多轮查询还原个体数据，那么在方案步骤二里 CB 即可能返回空的 token 给数据分析者，拒绝本次数据分析请求。由 2.2.5 节可知，区块链上数据具有不可篡改的特性，因此在访问控制环节上，数据的完整性能得到保证，也就能信任这些作为访问控制的源数据。

（3）终端数据的机密性

终端数据的机密性是由同态加密技术所确保，本文在后续实验部分所用到的同态加密系统为 Paillier 同态加密方案，该算法来自式（2-20）求解 n 次剩余的数学难题，可证明该破解该方案密文是无法用多项式时间求解的；而从式（2-31）知，解密时所需的λ来自式（2-16）：$\lambda(n) = \mathrm{lcm}(p-1, q-1)$，也就必须先取得一组相异的大素数p和q。另外，边缘服务器层也无从窃取解密所需的私钥 sk，因为无法得知中枢区块链网络中何节点拥有私钥，即使被探知私钥具体所在的节点位置，也没有提供私钥对外曝露的接口，这是因为智能合约被部署完成后的代码是无法篡改的，只要接口写定成不让边缘服务器层访问，边缘服务器层去窃取私钥就变得极其困难。由此，可保证边缘服务器层无法还原被加密的数据，只能在数据的密文域执行预期的计算。

3.4 本章小结

本章在第一节阐述了方案诞生的背景以及期望解决的问题，同时给出了一个方案的大框架和当中参数符号的解释；第二节，在第一节方案框架的基础上细化了实现一轮数据分析

所用到的步骤，阐释了由数据分析者、中枢区块链、终端设备层以及边缘服务器层之间的关系和运作方式，其中关键的第四至第十步聚焦在后三者所组成的边缘环境上；在本章第三节，列举了阻挡单点攻击、避免数据外包对象过度请求以及终端数据的机密性三项层面，分别剖析了其对终端用户隐私的作用和影响。

第四章 基于以太坊和腾讯云的测试平台搭建与性能测试

本章针对第三章给出的理论模型实践一个基于以太坊 Ropsten 区块链、微信小程序以及腾讯云开发的测试系统，分别对应着图 3-1 三点架构中的中枢区块链、终端设备层以及边缘服务器层，期望在本节给出理论方案可行的证明。

4.1 系统框架

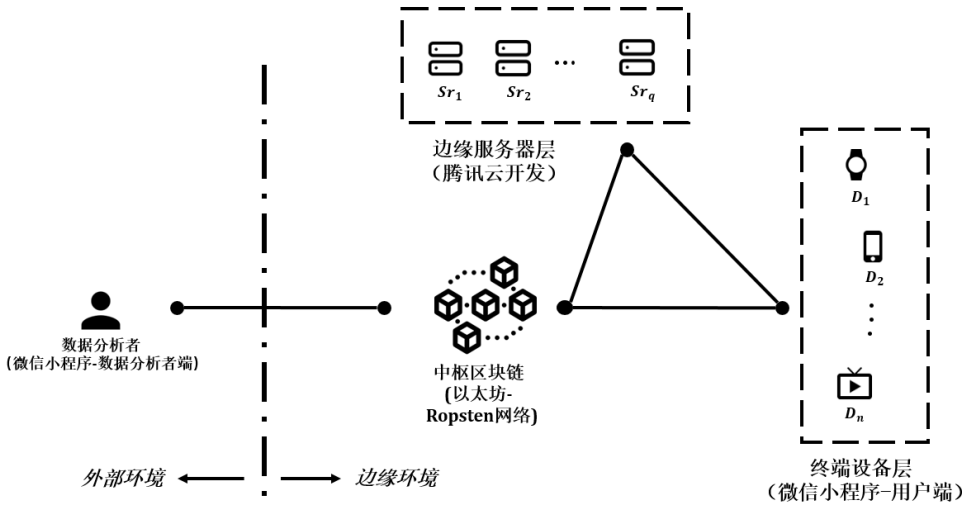

图 4-1 一种基于区块链技术保护隐私的系统方案

图 4-1 展示了一种基于区块链技术保护隐私的系统方案，该系统方案与图 3-1 相似，细化了各层所使用到的实例。以下就当中被采用的实例做说明：

（1）以太坊（Etherum）：

一个开源的拥有智能合约部署功能的区块链平台。于 2013 至 2014 年间由 Vitalik Buterin 提出[34]，该平台的一大特点是其提供了一套编程语言 Solidity/Serpent 供开发者编写，编写成的程序脚本称为智能合约，其被成功部署后拥有防篡改的特性，是首个供开发者将智能合约应用到区块链的平台。

本文的中枢区块链模拟即以以太坊作为基底实现。以太坊除了主网外，还拥有若干个测试网络，本文所采用的区块链网络为以太坊-Ropsten 测试网络。

（2）微信小程序：

一套由腾讯公司微信团队于 2017 年 1 月提出的编程语言框架，运行于微信应用程序内，借由 webview 和手机原生系统混合渲染的方式兼顾了终端程序开发的快速性和用户使用体验上的可用性，是一个近年来备受开发者追捧、完成的代码能快速触及大众的终端产品形态。其编程语言由 wxml/wxss/javascript 所编写，wxml 类似于 html 定义界面的布局，wxss 类似于 css 定义界面的样式，而 javascript 则负责逻辑层的实现。2020 年 1 月，微信小程序硬件框架 WMPF 上线[6]，可实现将小程序代码运行在非手机的硬件设备上，成为一种面向海量物联网设备的代码形态。

本文在终端设备层和数据分析者的用户界面皆采用微信小程序开发。

[6] https://developers.weixin.qq.com/doc/oplatform/Miniprogram_Frame/

（3）腾讯云开发：

腾讯云开发是由腾讯公司云开发团队打造的 Serverless 服务。Serverless 服务没有一个统一的定义，不过业界有共识指的是后端即服务（BaaS）和函数即服务（FaaS）的泛称。Serverless 主要整合了后端所需的各项资源，使开发者无需租用服务器，也无需关心后端运维负荷，即可享用后端存储和计算的能力。

本文使用腾讯云开发模拟边缘服务器层，使用其云函数服务在同态加密的密文域上展开计算，同时使用其作为终端层和中枢区块链之间的代理，减轻终端普遍资源不足的压力。

4.2 系统环境和开发

本系统所使用到的平台和开发工具如下所示：

表4-1 系统底层平台和开发工具

工具和平台	说明
CentOS 7.2	运行云函数的操作系统
Node.js 8.9	运行云函数的底层编程语言
ethereumjs-tx 2.1.2	运行于云函数的私钥签名库，为交易做签名用
web3 0.20.6	运行于云函数的模块，负责与区块链上节点建立通信
web3-eth-abi 1.2.7	运行于云函数的模块，负责 abi 与字符串之间的转码
solcjs	Nodejs 模块，负责将智能合约 sol 文件转成 abi 文件
remix-ide	网页，用于在线编译和部署智能合约
infura.io	以太坊节点托管平台
MetaMask chrome 扩展	以太坊账户管理工具
big-integer	大数处理库
paillier-js	Paillier 同态加密库
微信小程序 sdk 2.11.0	小程序运行环境基础库
微信开发者工具	小程序集成开发工具
Visual Studio Code	代码编辑器，主要用于编写智能合约

安装或配置完成后，即可着手开发。开发依照图 4-1 的框架，分成中枢区块链智能合约部署和各层与区块链代理桥梁的部分，终端层界面布局、样式以及逻辑的部分，以及边缘服务器层执行密文域计算的云函数部分。其中，终端层又分成数据分析者和一般用户两种操作界面。以下就这三部分开发过程展开解释：

（1）中枢区块链智能合约部署和各层与区块链代理桥梁

编写了一个 CentralBlockChain.sol 文件的智能合约，由 Solidity 语言编写而成。当中有若干函数，其中核心的 canAnalyze()函数负责决定是否通过当前预分析请求，分别对应着 3.2 节方案的第二步；isAnalysisLegal()函数负责检查当前 preAnalysisToken 是否合法，分别对应 3.2 节方案第四步的中枢区块链验证。这两个核心函数对外入参接口和返回参数如图 4-2 所示。

```
function canAnalyze(
    string memory id,
    string memory analysis_no,
    string memory analysis_func,
    string memory analysis_title,
    string memory analysis_desc
) virtual public returns(string memory, string memory);

function isAnalysisLegal(
    string memory id,
    string memory analysis_no,
    string memory preAnalysisToken
) virtual public view returns(bool, string memory);
```

图 4-2 CentralBlockchain.sol 中两个核心的函数

各层与区块链代理，是由云函数实现的。由于小程序终端代码包单一分包有 2MB 的容量限制，而与区块链交互的 web3.js 模块本身就接近 2MB,故编写一个名为 cb 的云函数，作为连接区块链和终端设备的桥梁。

（2）终端层界面布局、样式以及逻辑

该部分由名为 enter,identity,user 以及 analyst 等页面所组成,分别对应着入口页、身份选择页、终端用户页以及数据分析者页。入口页负责获取用户唯一识别，调用云函数获取用户的唯一 id；identity 页提供两个按钮，供用户选择“终端用户”亦或是“数据分析者”；user 页负责监听当前有无数据分析请求，若出现了数据分析请求，展示当前分析请求包裹，提供一个输入框供用户输入一个 1-99 的数字模拟一个秘密值，用户点击提交按钮后再进行加密和发送；analyst 页面则为数据分析者操作界面，主要用于发起数据分析，在智能合约里每个涉及链上数据创建和修改的[7]操作被视为一种交易，由于链上数据同步和打包出现分叉的缘故，不是每笔交易都会被百分之百确认，因此在 analyst 界面还涉及不断监听交易请求是否被成功载入。这四个页面完整的呈现如图 4-3 所示。

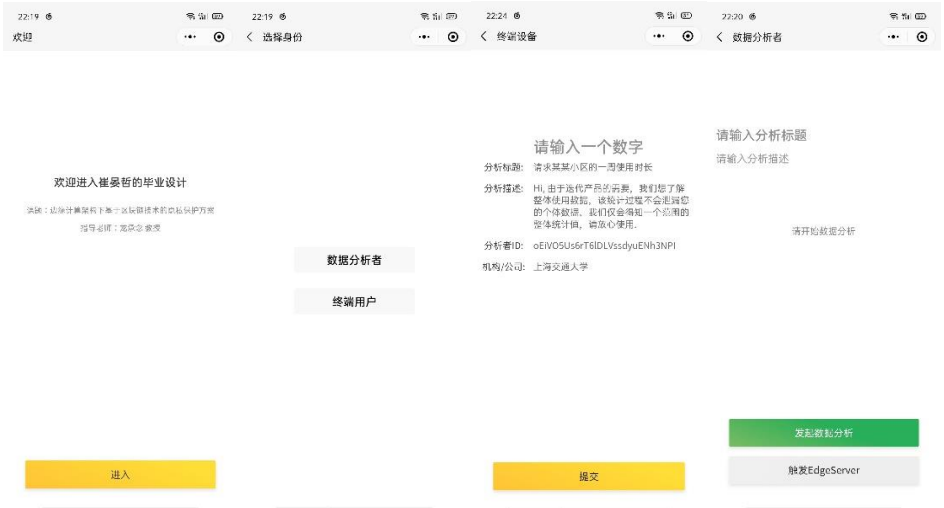

图 4-3 从左至右分别为进入页、身份选择页、终端用户页以及数据分析页

[7] 这里的“修改”是指复制一个新的数据结构再对原数据进行更改，原来的数据仍是存在、可追溯的。

（3）边缘服务器层执行密文域计算的云函数

边缘服务器层的代码被写进 edgeServer 云函数里，主要任务在于收集完毕从终端层传来的密文后，执行式 3-7 确定的密文域计算函数，实例为 paillier-js 的 publicKey.addition()函数。图 4-4 即为该云函数核心的计算部分。

```
let sum = bigInt(d2[0].enNum)
for (let i = 1; i < actualN; i++) {
  let v = d2[i]
  let enNum = v.enNum
  let tmpInt = bigInt(enNum)
  sum = publicKey.addition(sum, tmpInt)
}
```

图 4-4 edgeServer 云函数核心的计算部分

值得注意的是，publicKey.addition()虽然名为 addition，由式 2-26 知，两个密文域上的大素数ω_1与ω_2实为相乘的关系。这里的 addition 仅告知开发者这样的操作等价于明文域上的累加。

4.3 单元测试

本节单元测试主要测试 Paillier 同态加密系库的性能，分为四个部分，分别是不同公钥长度对于运行生成密钥函数的耗时影响，不同公钥长度在运行生成密钥函数时对 CPU 占用率的影响，不同公钥长度对于运行加密函数的耗时影响，以及不同明文数字对于加密耗时的影响。

4.3.1 不同公钥长度对于运行生成密钥函数的耗时影响

设置生成不同的公钥长度，所需的耗时应随公钥长度增长而增多。以下执行多轮公钥生成密钥函数后所得的结果。

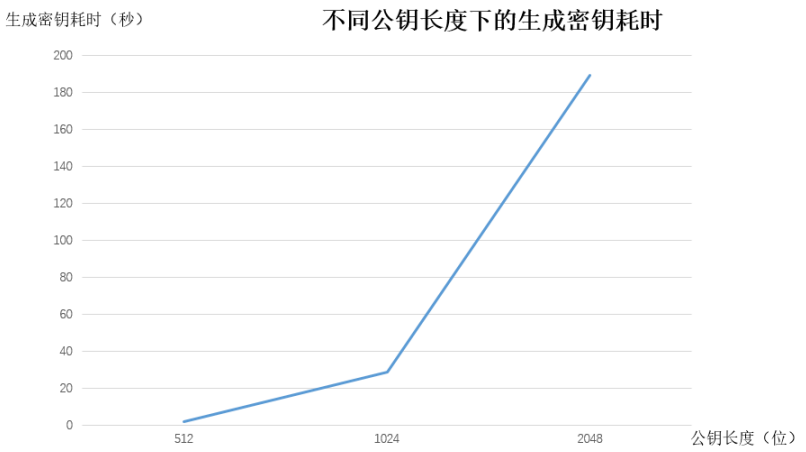

图 4-5 不同公钥长度对于运行生成密钥函数的所需耗时

从图中可知，随着公钥长度的增长，生成密钥所需耗时呈现指数级的增长。值得注意的是，运行的时间主要消耗在随机寻找大素数上。由图 4-6 所示的代码片段可知，有一个 do-while 回圈负责随机寻找两个大素数，这即为耗时随指数增长的主因。

```
const generateRandomKeys = function (bitLength = 1024, simplevariant = false) {
    let p, q, n, phi, n2, g, lambda, mu;
    // if p and q are bitLength/2 long ->  2**(bitLength - 2) <= n < 2**(bitLenght)
    do {
        p = bigInt.prime(bitLength / 2);
        q = bigInt.prime(bitLength / 2);
        n = p.multiply(q);
    } while (q.compare(p) == 0 || n.bitLength() != bitLength);
```

图 4-6 运行生成密钥函数耗时的主因来自于随机寻找大素数上

4.3.2 不同公钥长度在运行生成密钥函数时对 CPU 占用率的影响

在设置公钥长度为 512 位时，运行生成密钥函数KenGen(·)时的 CPU 占用率如图 4-7 所示。

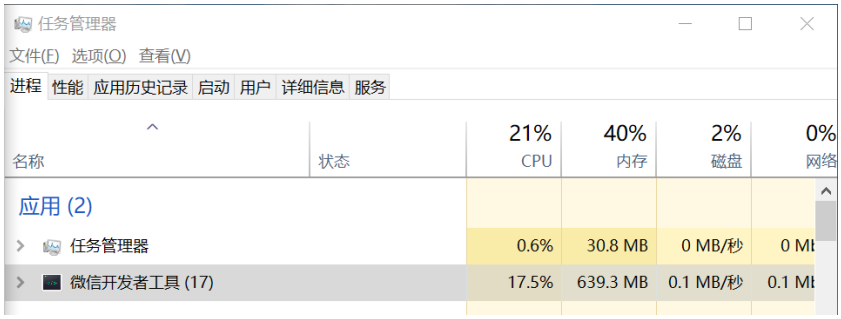

图 4-7 在设置公钥长度为 512 位时运行生成密钥函数的 CPU 占用率

接着，再多轮测试 1024 位以及 2048 位公钥长度的情况，有结果如图 4-8 所示。

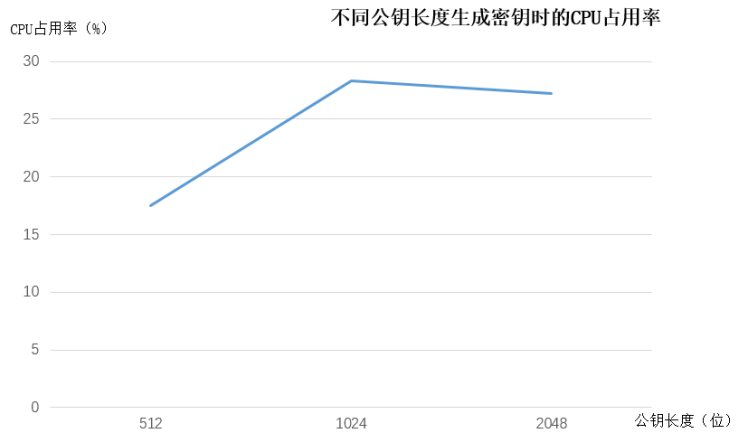

图 4-8 不同公钥长度在运行生成密钥函数时对 CPU 占用率的影响

从图 4-8 发现 1024 位公钥长度下 CPU 占用率最大值为 28.3%，而 2048 位公钥长度下 CPU 占用率最大值则为 27.2%，由此可见 CPU 占用率约在 28%上下出现阈值。该折线图表明，随着公钥长度的增长，CPU 占用率并不会无止境的增长，而是有一明显阈值，当 CPU 占用率触及该阈值后即不再增长。结合图 4-5，当设置 1024 位及其以上的公钥长度后，CPU 皆会以 28%上下的占用率资源为之付出计算，在这公钥长度以上的长度变化仅体现在运行耗时上。

4.3.3 不同公钥长度对于运行加密函数的耗时影响

前两小结主要在对生成密钥函数 KeyGen()的性能进行分析，本小节和 4.3.4 小节主要

针对加密函数 Encrypt()上的性能着手研究。

在不同公钥长度的前提下，执行加密函数所需的耗时增长如图 4-9 所示。

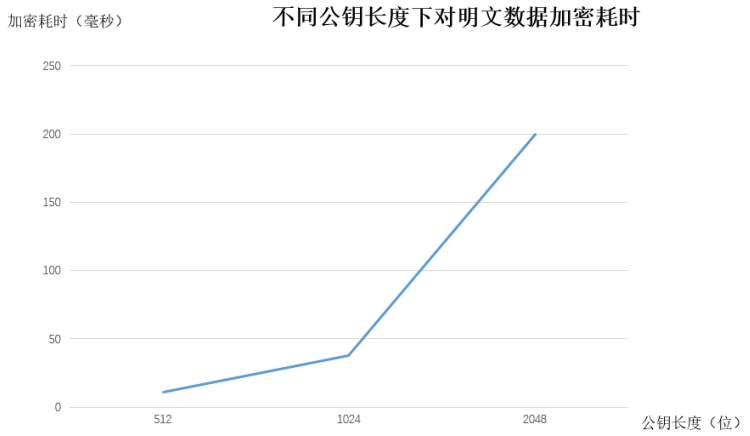

图 4-9 不同公钥长度情况下关于加密耗时的关系图

从图 4-9 可见，随着公钥长度的增长，加密所需耗时呈现指数级的增长，该关系同图 4-5 类似。

4.3.4 不同明文数字对于运行加密函数的耗时影响

同时，还应该关心不同的明文数字是否会对加密耗时产生影响。于是，随机选取了三个数字 7、31、84 以及 1051，以相同公钥长度的公钥对其进行加密考察其所需耗时，反复测试后结果如图 4-10 所示。

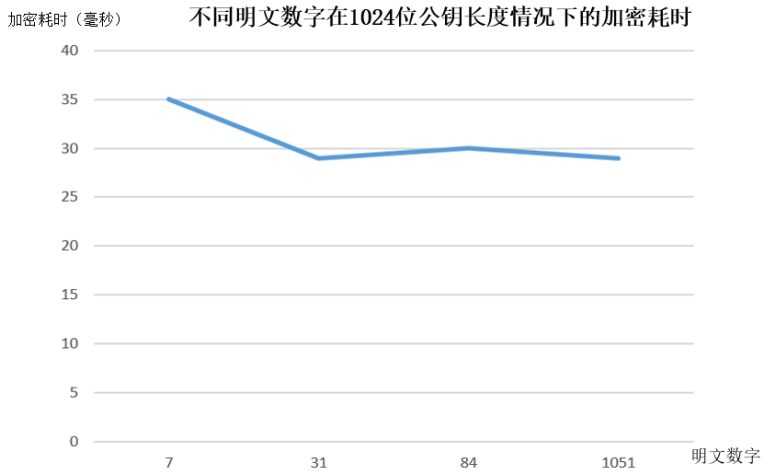

图 4-10 不同明文数字在 1024 位公钥长度情况下的加密耗时

从图 4-10 可见，相异明文数字关于加密耗时的影响不大，因此可以忽略明文数据对于加密耗时的影响。

4.4 系统测试

从图 4-5 以及云函数目前有 20 秒运行时长的限制，故本系统选定 512 位的 Paillier 同态加密公钥做测试。一个完整的系统测试流程如表 4-2 所示。

表4-2 系统测试主要流程

次序	流程	主要实现层	备注
1	发起交易	数据分析层 centralBlockchain	operation = 1 对应方案步骤 1
2	确认交易	centralBlockchain	operation = 2 对应方案步骤 1
3	取 Token	centralBlockchain	opertaion = 3 对应方案步骤 2
4	A 发起正式分析请求 同时监听是否得到响应； CB 生成公私钥对。	数据分析层 centralBlockchain	operation = 4 对应方案步骤 3 和 4
5	终端监听是否有分析	终端层	对应方案步骤 5
6	终端加密数据并发送	终端层	对应方案步骤 6 和 7
7	边缘服务器层展开计算，计算函数设置为取平均	edgeServer	对应方案步骤 8 和 9
8	edgeServer 返回计算结果给 CB， CB 解密、发送结果给 A	centralBlockchain	operation = 5 对应方案步骤 10
9	A 监听 Analysis 集合， 得到结果。	数据分析层	对应方案步骤 11

承表 4-2，经多轮测试，一个各环节的耗时结果如表 4-3 所示。

表4-3 系统测试主要流程

次序	流程	耗时（毫秒）	备注
1	发起交易	9379	
2	确认交易	32370	
3	取 Token	1480	
4	A 发起正式分析请求 同时监听是否得到响应； CB 生成公私钥对。	4090	
5	终端监听是否有分析	547	在已存在数据分析的情况下
6	终端加密数据并发送	581	
7	边缘服务器层展开计算，计算函数设置为取平均	7	在$n' = 10$的情况下
8	edgeServer 返回计算结果给 CB， CB 解密、发送结果给 A	259	
9	A 得到数据分析结果	/	

整体系统测试，根据第三章给定的理论框架，依据实验可行性做了增删改，从实践的角度拆分成以上九个单元进行测试。从表 4-3 发现次序 1 和 2 耗时最为明显，其中次序 2

确认交易在等待交易被确认，涉及以太坊网络共识机制挖矿和块同步的过程，故耗时最为突出。当数据分析成功被接受后，可借由 view 属性只读的智能合约接口，获取后续通行的权鉴， 反之则返回空的权鉴，这一步测试对应次序 3 也需要一秒多的时长。当数据分析者获得 Token 后，发起正式分析请求，这一步对应次序 4，在该步中 cb 云函数仅作桥接，收到 Token 后向区块链确认是否合法，若为合法 cb 云函数即开始生成同态加密所需的公私钥对。次序 4 同样耗时较长，除了需要同链上的节点进行通信外，还需要生成公私钥对，故耗时也较为明显。其他环节的测试，皆在一秒内完成，累加起来不超过三秒。

整体而言，写链上数据的函数耗时最为突出，约需要 30 秒左右；接着是生成密钥的过程，约需 3 秒；读链上数据的函数耗时再次之，约需 1.5 秒；其余过程则皆在 1 秒内完成，累计约 3 秒。整体系统流程，完成一论数据分析约需 45 秒上下，若加上终端用户十秒左右的输入时长，整体耗时约在 60 秒上下完成。数据分析场景属于低频场景，在次序 1 发起交易时，智能合约会去读过去五分钟内有无成功的交易，若有则拒绝了当前数据分析请求，同时数据分析者每天取一次结果即已足够多，故一分钟的等待时长尚在合理范围内。

4.5 场景分析

本小结就频繁数据分析请求和使用任意大素数解密两项潜在场景进行分析：

（1）频繁数据分析请求

在数据分析者尝试使用多轮查询时发生，借由微小的查询变化尝试得出特定个体数据的时候发生，也可以看作是一种单点攻击。

解决方案是直接在智能合约上设定多轮请求间的最小时间差，若在时间差内再次请求数据分析，则会被系统拒绝。由于链上数据的可信任性，判定的源数据就是可信的，同时已部署的智能合约无法修改，这让判定的结果就是绝对公正的，避免了单一节点垄断的可能。一个合理的最小时间差 MIN_DIFF，应该满足

$$\Delta n \geq 2，T > \text{MIN\_DIFF} \tag{4-1}$$

其中，Δn为在接受数据分析的范围内秘密值发生变化的终端个数,T 为数据分析的周期。式 4-1 的意义在于周期内必须有个数大于等于二的秘密值发生变化。若$\Delta n = 1$，即泄露了某个个体的秘密值；而$\Delta n = 0$，则浪费了系统资源，也容易造成攻击者不断监听的可能，只要Δn一变为 1，某单位个体的隐私也发生了泄露。在测试系统里，系统被设置了 5 分钟的多轮请求最小时间差。

（2）使用任意大素数解密

场景是若有人尝试任意大素数解密，这样系统又会反应何种结果是值得关心的。由于考量到公私钥存储若皆存于区块链会有被公开的危害，公链上的数据即使经过加密任何能也都能访问它，所以在系统中公私钥的存储仍选择了传统的集中式存储。

为模拟该场景，将数据库上的私钥大素数，做任意修改：

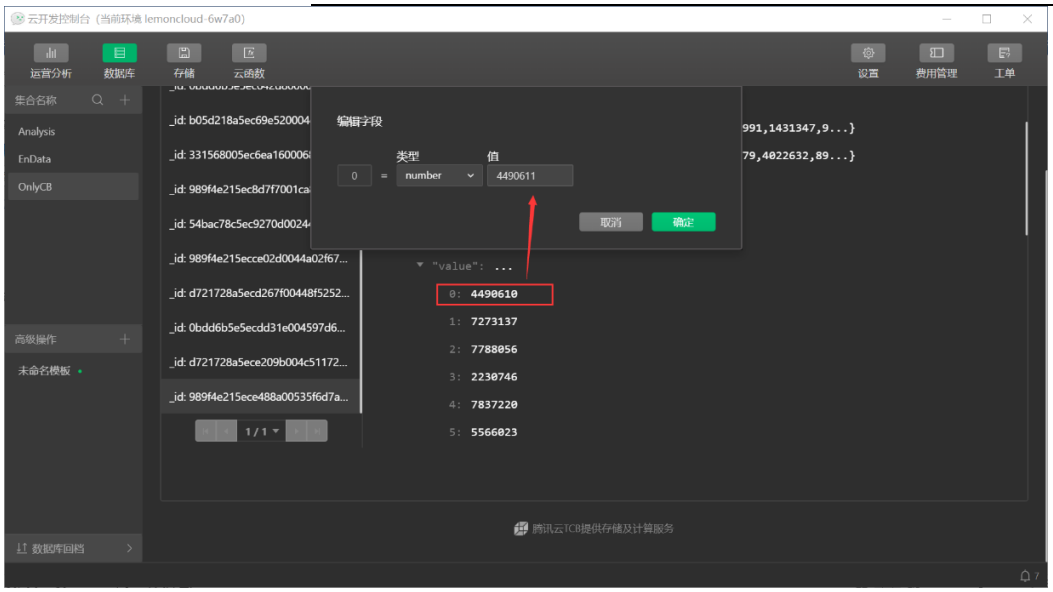

图 4-11 篡改私钥操作

如图 4-11 所示，不妨将私钥的实例结构中，Paillier 解密过程式 2-31 中λ的实例 lamda 数组做微小更动，测试解密时将产生何种结果。测试结果如图 4-12 所示。

图 4-12 篡改私钥后结果为一个无意义的大数

图 4-12 展示了，只要任意微小的变化，即使再接近私钥的真值，任何人皆是无法破解的，验证了多项式时间无法破解该系统的私钥这个命题，也由此测试了解到系统良好的稳壮性，方能确信其有能力保障用户的隐私。

4.6 本章小结

本章第一节依照第三章的理论框架对各环节锁定了实例，并分别展开了以太坊、微信小程序和腾讯云开发的功能和使用场景；在本章第二节就各个环节所需要的工具逐一列出，并将开发过程分述成“中枢区块链智能合约部署和各层与区块链代理桥梁”、“终端层界面布局、样式以及逻辑开发”以及“边缘服务器层执行密文域计算的云函数”等三大部分，对每个部分开发的重点进行了解释；第三节对系统中所用到的 Paillier 同态加密的性能做了分析,分别考察公钥长度对生成密钥耗时的影响，公钥长度对 CPU 占用率的影响，公钥长度对加密耗时的影响，以及不同明文数据对加密耗时的影响；第四节，则从实践层面把系统测试流程分为九步，分别分析各环节所需的耗时以及潜在的原因；第五节，对两个潜在的场景“频繁数据分析请求”和“使用任意大素数解密”进行分析，由数学公式辅助和实际检验得出系统良好的健壮性。

第五章 总结和展望

5.1 总结

这十年移动终端应用兴起，到近几年不仅限于智能手机的各式智能终端设备不断出现，使学界关于海量数据的处理以及终端用户的隐私一直成为研究热点。本文的工作在于研究边缘计算架构下的隐私保护方案，提出了一个由中枢区块链、终端设备层和边缘服务器层组成的边缘环境架构，以区块链技术和同态加密技术作支撑，保证终端下的用户隐私；同时使用以太坊、web3.js、微信小程序以及腾讯云开发对方案进行模拟，验证了架构的可行性；并再最终对系统各单元进行评估，对各个可能潜在的场景进行分析，完善系统实践中所遇到的细节。

本文整体考虑到由于成本的限制造成终端设备存储空间不足或计算资源有限的情况，故以小程序单个代码包仅能 2MB 大小作为既定的限制，在此限制下去探索能快速覆盖大量用户的小程序或其他终端产品形态也能考量用户的隐私。实验结果表明，运用区块链技术和同态加密技术能很好的兼顾用户体验又能保证用户隐私。尽管牺牲了数据分析者的部分响应时长，但在数据分析的场景下处于合理的范围内。因此，验证了即使在有限的终端资源中，同样能运用区块链技术和同态加密技术等新兴技术来保证终端的隐私，使用户毋须以快速方便的易用性牺牲了自身涉及敏感数据的隐私，这正是本文的贡献和研究意义所在。

5.1.1 研究成果

本文研究的重点是解决边缘计算架构下终端资源不足和边缘环境信任两难所衍生的隐私问题，具体的研究成果归纳如下：

(1) 提出了一种基于区块链、终端设备层以及边缘服务器层的三点架构隐私保护方案。凭借区块链数据不可篡改性和可追溯性，确保评判数据分析方是否滥用分析的过程是公正的，同时以同态加密技术让用户隐私数据具备可用性和机密性。

(2) 完成一套由以太坊、微信小程序以及腾讯云开发所组成的三点架构系统。过程中，实现了兼容小程序有限环境限制的 Paillier 同态加密技术，验证了终端资源不足的情况下，即使只有 2MB 代码包的限制，也能为用户隐私保驾护航，使用户的原始数据仅锁于终端层内，不被另外其他层或第三方所得知；另外，还实现了基于以太坊平台的链上存储机制和访问控制，减轻了单点攻击的危害，同时避免了数据被篡改的可能性，在数据安全层面考量用户的隐私。

(3) 分析系统性能并开源原始码。本文对系统的性能从各单元的测试到总体的测试皆做了分析，总结来说一轮数据分析在一分钟左右完成，性能大致满足需求。同时，一个在探讨隐私保护的课题，还必须兼顾用户心理层面的考量，如何让广泛大众认知到系统的隐私保护程度进而相信隐私保护机制正在作用是不可回避的问题，本文的做法是开源原始码：https://github.com/yenche123/cb-privacy，供世人任意查阅，减轻用户不信任系统的疑虑。

然而，本系统并非完美的，仍存有以下几点未竟之业：

(1) 未有共识层的设计。在方案中提到选举中枢区块链领导节点的过程，以它来完善后续跟各层的交互，本文仅在应用层和智能合约层做设计，未研究到涉及区块链网络中节点作恶的情况，为美中不足之处。

(2) 终端直接与区块链网络面临困难。在本文系统实践中，由于终端直接安装 web.js 和 ethereumjs-tx.js 会遭遇代码包超额的问题，故实际上终端层与中枢区块链的

交互仍靠云函数做代理，而云函数在本质上为一个集中式计算的模式，不可避免的仍面临单点攻击的危害，同时也存有数据垄断的疑虑，这同为本文缺憾之处。

总的来说，本文所作的贡献在于将区块链技术和同态加密技术直接下放到面向十亿用户的手机内，将这两者方兴未艾且互有联系的技术完成了应用落地，在边缘计算领域为用户隐私付出一己力量。

5.1.2 对人文环境与社会的影响

近几年，科技与隐私之间的关系与相互的影响是个时下热门话题。即使有人认为用户免费使用了许多互联网服务，牺牲一点隐私是合理的，但若问他如果隐私不重要，请他将自己的邮箱地址与密码共享出来，供众人随意浏览，多数人也是不愿意的。

然而，尽管隐私的关注度受到了提升，但层出不穷的数据泄露事件不断让用户感到失望；同时，即使科技公司宣称在隐私层面做了总总考量，但用户看不到内部实现机制，代码运行在机器里使得用户无从感知，都加深了用户在科技面前不存在隐私这样悲观的想法。

一个社会的进步不应只是物质上的日新月异，在制度上和精神上也要同步并进才是人类文明真正的进步。在重建大众对科技关于隐私层面的信任上，本文采取的是以通俗易懂的方案建立架构以及开放原始码两种方式。前者是指避开普罗大众无法领会的艰涩方案，而是采用直观易于理解的方式，建立三点架构这样巧妙的关系；后者是指让系统实现的机理或底层运作的方式毫不遮掩地公开于世，让每个人都有权查阅，每个人都有办法知情这些智能终端究竟是如何存储和处理我们的隐私信息，才方能使我们自己放心坦然的使用之。

5.2 展望

边缘计算与区块链技术皆是近年研究热点，将这两者新型技术应用在隐私保护领域还有广泛的探索空间。本文给出了一个基于区块链技术在边缘计算架构下的隐私保护可行方案，然而仍有诸多可以深入研究之处。如前文不足之处所述，可以在共识层上深入研究，让系统在区块链层更加健全；另外，还可以改善 web3.js 代码，使之更加轻量能兼容于终端设备，进而让终端直接于区块链建立通信；同时，还可以引入更多同态加密方案或其他隐私技术，提升边缘服务器层计算更多期望函数的能力。

最后，期望在未来的工作里能将这些隐私保护技术普及至更多的终端设备和应用程序里，挖掘出这些技术背后的蕴含价值，重塑人们对科技美好的向往。

参考文献

[1] AL-FUQAHA M, GUIZANI M, MOHAMMADI M, ALEDHARI M, AYYASH M. "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," in IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347-2376, Fourthquarter 2015.

[2] 方俊杰,雷凯.面向边缘人工智能计算的区块链技术综述[J].应用科学学报,2020,38(01):1-21.

[3] ROMAN R, LOPEZ J, MAMBO M. Mobile edge computing, fog et al.: a survey and analysis of security threats and challenges[J]. Future Generation Computer Systems, 2018, PP(78): 680-698.

[4] WARD R, BAYER B. “BeyondCorp: A new approach to enterprise security,” login, vol. 39, no. 6, pp. 6–11, 2014.

[5] WATERS B. Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization[C]//The 14th International Conference on Practice and Theory in Public Key Cryptography(PKC’11). 2011: 53-70.

[6] PASUOULETI S K, RAMALINGAM S, BUYYA R. An efficient and secure privacy-preserving approach for outsourced data of resource constrained mobile devices in cloud computing[J]. Journal of Network and Computer Applications, 2016, 64(C): 12-22.

[7] WANG C J, LI W T, LI Y, et al. A ciphertext-policy attribute-based encryption scheme supporting keyword search function[C]//The 5th International Symposium on Cyberspace Safety and Security (CSS’13). 2013: 377-386.

[8] WANG C, WANG Q, REN K, et al. Privacy-preserving public auditing for data storage security in cloud computing[C]//The 29th IEEE AnnualInternational Conference on Computer Communications (INFOCOM’10). 2010: 1-9.

[9] SAMARATI P. Protecting Respondents’ Identitiy in Microdata Release[J]. IEEE, Transactions on Knowledge and Data Engineering, 2001, 13(6):1010-1027

[10] SWEENEY L. K-Anonymity: A Model for Protecting Privacy[J]. International Journal of Uncertainty Fuzziness and Knowledge-Based Systems, 2002, 10(5):557-570

[11] 张鹏,童云海,唐世渭,杨冬青,马秀莉.一种有效的隐私保护关联规则挖掘方法[J].软件学报,2006(08):1764-1774.

[12] DWORK C. Differential privacy[C]. International Colloquium on Automata Languages and Programming, 2006:1-12.

[13] DWORK C, MCHSERRY F, NISSIM K, et el. Calibrating noise to sensitivity in private data analysis[J]. Theory of Cryptography Conference, 2006:265-284.

[14] MCHERRY F, TALWAR K. Mechanism Design via Differential Privacy[C]. Foundations of Computer Scence, 2007:94-103.

[15] VALENTA L, ROWAN B. Blindcoin:blinded accountable mixes for bitcoin[C]//Proceedings of International Conference on Financial Cryptography and Data Security.Berlin, Germany:Springer, 2015:112-126.

[16] MIERSI，GARMAN C, GREENM, et al．Zerocoins: anonymous distributed E-cash from bitcoin[C]//Proceedings of IEEE Symposium on Security and Privacy.Washington D. C. , USA:IEEE Press, 2013:394-411.

[17] SHAMIR A. 1979. How to share a secret. Commun. ACM 22, 11 (Nov. 1979), 612–613. DOI:https://doi.org/10.1145/359168.359176.
[18] BLAKLEY G. Safeguarding cryptographic keys[C]//Proceedings of the 1979 AFIPS National Computer Conference (p./pp. 313--317), Monval, NJ, USA: AFIPS Press.
[19] 张佳乐,赵彦超,陈兵,胡峰,朱琨.边缘计算数据安全与隐私保护研究综述[J].通信学报,2018,39(03):1-21.
[20] SATOSHI N. Bitcoin: A peer-to-peer electronic cash system. Consulted, 1:2012, 2008. http://nakamotoinstitute.org/bitcoin/.
[21] JOSEPH D. TOUCH. Performance analysis of MD5//ACM SIGCOMM Computer Communication Review October 1995: 77-85.
[22] 刘飞. Hash 函数研究与设计[D]. 南京航空航天大学,2012.
[23] 韩国栋,麦志英,赵玉香.数字货币共识算法综述与展望[J].青海金融,2020(01):9-12.
[24] DWORK C, NAOR M. Pricing via processing or combatting junk mail[C]. In:Advances in Cryptology—CRYPTO'92. Springer Berlin Heidelberg, 1993:139-147.
[25] KING S, NADAL S. PPCoin: peer-to-peer crypto-currency with proof-of-stake [EB/OL]. 2012. https://www.chainwhy.com/upload/default/20180619/126a057fef926dc286accb372da46955.pdf
[26] 任佩. 区块链技术中共识机制的安全分析[C]. 公安部第三研究所、江苏省公安厅、无锡市公安局.2019 中国网络安全等级保护和关键信息基础设施保护大会论文集.公安部第三研究所、江苏省公安厅、无锡市公安局:《信息网络安全》北京编辑部,2019:24-27.
[27] GUO Y, LIANG C. Blockchain application and outlook in the banking industry[J]. Financial Innovation, 2016, 2(1):24.
[28] BENET J. IPFS - Content Addressed, Versioned, P2P File System[J/OL], 2014. https://ipfs.io/ipfs/QmR7GSQM93Cx5eAg6a6yRzNde1FQv7uL6X1o4k7zrJa3LX/ipfs.draft3.pdf
[29] 李琪,李勍,朱建明,关晓瑶,王慧,郄晨梓. 基于区块链技术的慈善应用模式与平台[J]. 计算机应用,2017,37(S2):287-292.
[30] 王子岳. 面向供应链溯源的区块链系统研究与优化[D]. 北京交通大学,2019.
[31] RONALD L, MICHAEL L, DERTOUZOS M. On Data Banks and Privacy Homomorphisms[M].[S.1]:In Foundations of Secure Computation,1978.pp.169-177.
[32] 陈志伟. 同态密码理论的研究与应用[D].西安电子科技大学,2014.
[33] PAILLIER P. Public-key cryptosystems based on composite degree residuosity classes［C］// International Conference on Theory and Application of Cryptographic Techniques, Berlin: Springer-Verlag, 1999:223-238
[34] BUTERIN V. A next-generation smart contract and decentralized application platform. White Paper, 2014

谢辞

能完成这次课题，首先我得感谢我的指导教授龙承念老师。特别感谢老师在万忙中给予我指导和交流，令我这半年多的时光里收获良多，真的非常感谢！

其次，我得感谢杨雨菡师姐给予我的指导和协助。非常感谢她总是在我姗姗来迟提交进度后马上给予我协助，还告知我应注意的地方，真的特别感谢！

最后，我得感谢我的家人和金凌敏女士，陪我度过这段艰辛的时光，真的非常感激他们的陪伴，让我走到了这里。谢谢你们！

PRIVACY PROTECTION SCHEME BASED ON BLOCKCHAIN TECHNOLOGY UNDER EDGE COMPUTING ARCHITECTURE

With the popularity of the fourth generation of mobile communication technology and the rapid development of the fifth generation of mobile communication technology, the number of the mobile terminals and the amount of data on the mobile edge show a blowout growth. How to process and protect these massive data is a crucial problem. Moreover, in the time-sensitive scenarios (e.g. unmanned driving), how to achieve efficiently calculation has become a hot topic.

Blockchain as a distributed peer-to-peer ledger technology, provides a secure and credible autonomous networks without the endorsement and verification of a third-party, which becomes a emerging solution of decentralized data storage.

Edge computing is a service model that has been proposed in recent years, which is different from traditional centralized computing. It is used to solve the problem of tight computing resources faced by traditional cloud computing. To avoid potential security risks of centralized servers, such as single point attacks, denial-of-service attacks (DoS), and distrbuted denial-of-service attacks (DDoS), data is sunk to the edge and processed by the edge server. In addition, the data requester and the computing provider are greatly drawn closer, the communication delay is reduced and the low latency demand of service is guaranteed.

In this paper, we focus on the protection of users' privacy in the edge scene. Using blockchain as the underlying technical support, we try to meet the industry's requirements for data fusion, communication and storage in terms of security and privacy protection during digital transformation.

The main contribution in our research is: to study domestic and foreign literature, to build a theoretical framework, and to deploy a testing system based Ethereum, WeChat Mini-Program, and Tecent Cloud Base.

The study of domestic and foreign literature is chapters 1 and 2. In this part, we research on edge computing architecture, blockchain foundation, and homomorphic encryption technology. With this background knowledge, it is found that the blockchain as a trust machine is suitable to support the cooperation of all parties in an untrusted environment. The edge environment is a perilous environment where the nodes do not trust each other. Thereforce, blockchain is a good choice as a trust center node in such a distrustful environment.

Building a theoretical framework, one of the core focuses of this paper, is what we take after reading domestic and foreign literature. We look forward to solving data outsourcing without revealing user privacy. The data outsourcing refers to the transfer of one or a group of data from one scene to another. In the process, the data must be processed properly for users' privacy. We propose a three-point structure consisting of a central blockchain, a terminal device layer, and an edge server layer.

The central blockchain serves as a trusty center for the edge environment. Due to immutability of data and decentralization of structure on the blockchain, the blockchain has enough advantages to convince the parties of the ability of its fairness when the parties do not trust each other. The main

work of the central blockchain is to permit requests from external data analysts, elect leadership nodes, generate asymmetric key pairs for homomorphic encryption, and verify requests for all parties.

The terminal device layer is the storage location of users' private data. The work at this layer is to verify whether the data analysis is legal, and the user has the right to choose whether to pass the data request initiated by the data analyst. In other words, users have the opportunity to execute the blacklist verification. If the verification is passed, the terminal device starts to encrypt the source data used, and then sends it to the edge server layer for processing.

As the name implies, the edge server layer is composed of edge servers. It is responsible for computing user ciphertext. When the edge server receives the ciphertext from the terminal device layer, as the terminal device layer does, it first requests the centeral blockchain for whether the data analysis is legal. After that, the data is calculated according to the functions expected by the data analyst. The entire process of calculation is completed in the homomorphic encrypted ciphertext domain. The edge server has no ability to restore the ciphertext to the plaintext state of the source data, and thus we ensure that the privacy of the user is protected at this layer. After the calculation, the edge server layer transmits the result in the ciphertext space to the central blockchain.

At the end, the leader node of the central blockchain decrypts the result in the ciphertext space with the private key that it can obtain, and obtains the result in the plaintext space. The result in the plaintext space is sent to the data analyst to complete a round of data analysis request.

In the theoretical framework, a total of eleven steps are required to realize a data analysis in the three-point architecture. They are: pre-analysis request, central blockchain (CB) access control, formal analysis request, CB verification and data distribution, terminal layer verification and blacklist detection, terminal layer encryption of source data, terminal layer sending ciphertext to the edge server layer and the latter verifying the validity of the request, the edge calculation on the edge server layer, the transmission of the result from the edge layer to CB, the decryption of the ciphertext result in CB, and the transmission of statistical results from CB to the data analyst.

The deployment of a testing system is the second core part of this article. We used Ethereum, WeChat Mini-Program and Tencent Cloud Base to simulate the theoretical framework. Ethereum is an open blockchain platform, which is the first blockchain to implement smart contracts so that develpers can deploy unchangeable code. We deploy the smart contract of the central blockchain layer on Ropsten, one of Ethereum’s testing network to achieve what the central blockchain needs to execute in the theoretical framework, such as access control after pre-analysis request and data distribution after formal analysis request. Furthermore, we design the user interface and develop the logic of users’ manipulation on WeChat Mini-Program. It is worth noting that there is a 2MB limit for the size of a single code package in a mini-program. This limit can be regarded as a manifestation of insufficient terminal resources. Besides, it can be assumed that the user in the terminal device layer can randomly input a number to represent the user’s private data. In addition, we use Tecent Cloud Base, a serverless service free of back-end operation and maintenance, to simulate the situation of the edge server layer. Some cloud functions the cloud base offers are deployed to realize the collection of ciphertext data from end users, calculation of ciphertext data, and transmission of calculation results to the central blockchain layer in the theoretical framework.

The experimental results show that a round of data analysis takes about one minute and can be enough to meet the needs of the scene related to data outsourcing. Furthermore, the various scenes of the system are analyzed. The results show that it has good anti-jamming ability in frequent

requests about data analyses and multiple decryption using arbitrary large prime numbers, and thus it is concluded that users' privacy is guaranteed in testing.

The focus of this research is to solve the privacy issues arising from the lack of terminal resources and the dilemma of trust in the edge environment under the edge computing architecture. The specific research results are summarized as follows:

(1) Propose a privacy protection scheme with a three-point architecture based on central blockchain, terminal edge layer and edge server layer. With the immutability and traceability of blockchain data, it is ensured that the process of judging whether the data analysis requester abuses any analysis is fair. Furthermore, homomorphic encryption technology is used to make user privacy data available and confidential.

(2) Complete a system developed by Ethereum, WeChat Mini-Program and Tencent Cloud Base. In the process, we implemented the Paillier homomorphic encryption technology compatible with the limited environment of mini-programs. Additionally, it has been verified that the system can protect users' privacy and only lock the users' original data in the terminal layer even if the terminal resources are tight and insufficient, such as the limitation of only 2MB code package. In other words, the sensitive information is not known by other layers or third parties. In addition, we have also implemented an on-chain storage mechanism and access control based on the Ethereum platform, reducing the harm of single-point attacks, avoiding the possibility of data tampering, and considering users' privacy in terms of information security.

(3) Analyze system performance and open source code. In this paper, we analyze the performance of the system from each unit test to the overall test. In summary, a round of data analysis is completed in about one minute, and the performance generally meets the needs. Moreover, when discussing the topic of privacy protection in terms of technology level, we must also consider the psychological level of users. It is an unavoidable problem: how to make the public understand the degree of privacy protection of the system and then believe that the privacy protection mechanism is working. The approach in this paper is to open the source code: https://github.com/yenche123/cb-privacy, which can be arbitrarily consulted around the world, to alleviate users' doubts of the system.

It is expected that this paper can do some meagre efforts in the field of user privacy based on blockchain technology under the edge computing architecture, and we look forward to inspring relevant researchers in the future.