区块链数据层安全与隐私保护设计指南

CSA GCR cloud security GREATER CHINA REGION alliance®

序言

在数字化变革的今天，随着新冠疫情仍在全球蔓延，大国间战略博弈塑造了现今百年未有之国际变局，包含数据安全的数字安全成为了保障国家安全的重要组成部分。各国间的战略博弈带来了全新挑战，威胁形式层出不穷，铸剑亮剑刻不容缓，各国通过立法、技术升级等各种措施积极应对挑战是国际共识。

随着新兴技术如区块链、人工智能等飞速发展，区块链逐渐成为“价值互联网”的重要基础设施，各国都开始积极拥抱区块链技术，开辟国际产业竞争的新赛道。区块链自身有着分布式、点对点传输、透明、可追踪、不可篡改、数据安全等特点，在某种程度上解决了数据的完整性、真实性及唯一性等问题，使得用区块链技术解决数据安全与隐私的问题成为人们竞相关注的焦点。

区块链数据安全与隐私保护能解决哪些问题？本文档从区块链数据概念与范畴、数据格式入手，分析区块链数据安全与隐私保护的需求，从接入层、处理层及展示层的设计指南全面呈现数据安全与隐私保护总体框架，并提供测试指南，最后以区块链数据安全应用场景收尾。文章内容深入浅出，是研究区块链数据安全与隐私保护领域很好的参考材料。

李雨航 Yale Li

CSA 大中华区主席兼研究院院长

致谢

云安全联盟大中华区（简称：CSA GCR）区块链安全工作组在 2020 年 2 月份成立。由黄连金担任工作组组长，9 位领军人分别担任 9 个项目小组组长，分别有：知道创宇创始人&CEO 赵伟领衔数字钱包安全小组，北大信息科学技术学院区块链研究中心主任陈钟领衔共识算法安全小组，赛博英杰创始人&董事长谭晓生领衔交易所安全小组，安比实验室创始人郭宇领衔智能合约安全小组，世界银行首席信息安全架构师张志军领衔 Dapp 安全小组，中国移动集团信安中心高工于乐博士领衔去中心化数字身份安全小组，北理工教授祝烈煌领衔网络层安全小组，武汉大学教授陈晶领衔数据层安全小组，零时科技 CEO 邓永凯领衔 AML 技术与安全小组。区块链安全工作组现有 100 多位安全专家们，分别来自中国电子学会、耶鲁大学、北京大学、北京理工大学、世界银行、中国金融认证中心、华为、腾讯、知道创宇、慢雾科技、启明星辰、天融信、联想、OPPO、零时科技、普华永道、安永、阿斯利康等六十多家单位。

本白皮书主要由区块链数据安全小组专家撰写，感谢以下专家的贡献：

原创作者：陈晶　何琨　杜瑞颖　刘鳌　粟栗　刘斐　王东晖　赵勇　贺志生　刘虎　张书东　徐丽华　董喆

审核专家：黄连金　陈钟　祝烈煌　王安宇　高巍　姚凯　郭鹏程

贡献单位：华为技术有限公司　OPPO 广东移动通信有限公司　北京奇虎科技有限公司

在此感谢以上专家。如译文有不妥当之处，敬请读者联系 CSA GCR 秘书处给与雅正！

联系邮箱：research@c-csa.cn；国际云安全联盟 CSA 公众号：

目录

1 区块链数据概念和范畴

1.1 区块链发展趋势

随着互联网技术的发展，计算机、大数据、人工智能、物联网和信息安全等技术对人们的工作和生活产生了深远的影响。互联网的快速普及促进了社会进步和国家经济发展。区块链技术于 2008 年由中本聪在比特币中首次提出之后，这十多年间处于快速发展的阶段。全球调查显示，有 39%的企业高管表示已将区块链投入生产当中。区块链的应用正在跨行业蔓延，远远超出了其最初的金融科技应用范围，技术、媒体、电信、生命科学、健康医疗和政务等更多领域正在扩展和丰富各自的区块链计划。

区块链作为数字时代的前沿技术，近年来逐渐成为中央和地方政府关注的重点，被视作具有国家战略意义的新兴产业，在 2019 年的全国两会上也被频繁提及。早在 2016 年，中国政府就开始了对于区块链技术的探索。区块链与 5G、物联网、人工智能都成为了未来科技发展的重中之重。2016 年 12 月，在国务院印发的《十三五国家信息化规划》中，就提出要强化战略性前沿技术超前布局，首次提及区块链技术。2017-2018 年，国务院发布的指导意见中，有 6 份文件提及要明确发展及利用区块链技术。2019 年，中央各部委出台区块链相关政策更是紧锣密鼓，在 2019 年 10 月 24 日，中共中央政治局第十八次集体学习时，习近平在主持学习时强调，把区块链作为核心技术自主创新的重要突破口，明确主攻方向，加大投入力度，着力攻克一批关键核心技术，加快推动区块链技术和产业创新发展。2020 年 2 月 5 日中央一号文件更是将区块链放在了人工智能、第五代移动通信网络、智慧气象等前列，体现了区块链产业在国家战略中的地位，也意味着中央对区块链产业的重视。此外，我国在此领域的专利申请比重由 2014 年的 33.33%上涨到 2018 年的 82.1%。以全球视野来看，我国在区块链技术领域占据了领先优势。

区块链作为分布式数据存储、点对点传输、共识机制、加密算法等技术的集成应用，近年来已成为联合国、国际货币基金组织等国际组织以及许多国家政府研究讨论的热点，产业界也纷纷加大投入力度。目前，区块链的应用已延伸到物联网、智能制造、供应链管理、数字资产交易等多个领域，将为云计算、大数据、移动互联网等新一代信息技术的发展带来新的机遇，有能力引发新一轮的技术创新和产业变革。，可以用来解决现有业务的一些痛点，实现业务模式的创新。在一定程度

目前，区块链逐渐成为“价值互联网”的重要基础措施，各国都开始积极拥抱区块链技术，开辟国际产业竞争的新赛道，抢占新一轮产业创新的制高点，以强化国际竞争力。区块链有着分布式、点对点传输、透明、可追踪、不可篡改、数据安全等特点上解决了价值传输过程中完整性、真实性、唯一性的问题。区块链技术将继续加快在产业场景中的广泛应用，与实体经济产业深度融合，形成一批“产业区块链”项目，将会成为区块链技术的应用趋势。同时，企业应用将成为区块链的主战场，联盟链/私有链将成为主流方向。例如：阿里将区块链应用到了电子票据、医疗信息存证、公积金存缴证明等生活场景上；腾讯将区块链的应用更侧重于金融方面，区块链电子发票、区块链游戏等；百度作为互联网三大巨头，也有百度区块链引擎 BBE、超级链等区块链产品。相比大企业，实际上中小企业更需要应用区块链来进行产业升级转型，以此来降低资金、时间、人工等成本并更快的获得市场认可取得更大的收益。

目前，网络安全问题越发突出，而随着区块链的快速发展，也会让区块链的数据安全问题成为人们关注的焦点。在区块链迅速发展的背景下，中国顺应全球化需求，紧跟国际步伐，积极推动国内区块链的相关领域研究、标准化制定以及产业化发展。与对比特币等数字货币监管日益趋严的背景相比，针对区块链技术的支持与鼓励已在全球范围内达成共识，国内区块链技术也将在未来数年内得到国家与地方的政策支持。在政策与市场的双层驱动下，区块链相关产业有望脱虚入实，区块链技术加速探索可能的应用场

景。

1.2 区块链数据概念

区块链数据是指对区块链中的事件进行记录并可以识别的抽象符号。主要包括区块、事务、实体、合约、账户、配置等数据对象。不同类别的数据对象功能不相同。这些区块链数据可以通过数标识符、中文名称、英文名称、数据类型、数据长度、数据说明、数据备注等属性来描述，且具有一定的格式规范。

1.3 区块链数据范畴

区块链数据主要包含以下数据对象：

1) **区块数据：**区块链网络的底层链式数据，用于把一段时间内发生的事务处理结果持久化为成块链式数据结构。通常区块由区块头和区块体构成。区块头包含相关控制信息，区块体包含具体事务数据。

2) **事务数据：**区块链系统上承载的具体业务动作的数据。事务包含非交易类型事务和交易类型事务。

3) **实体数据：**描述事务静态属性的数据。包含发起方地址、接收方地址、交易额、交易费用、存储数据及实体数据备注。

4) **合约数据：**描述事务动态处理逻辑的数据。是一套以计算机代码形式定义的承诺，以及合约参与方可执行承诺的协议。

5) **账户数据：**描述区块链事务的发起者和相关方的数据。区块中记录的事务信息均被关联到相关账户上，每个区块链服务客户拥有多个账户使用区块链服务。

6) **配置数据：**区块链系统正常运行过程中所需要的配置信息。包含共识协议版本号，软件版本号和网络通信底层对等节点配置信息等。

2 区块链数据安全与隐私需求

2.1 区块链数据保护安全需求

数据安全是区块链的基本安全目标。区块链作为一种分布式存储系统，需要存储包括交易、用户信息、智能合约代码和执行中间状态等数据。这些数据至关重要，是区块链安全保护的首要实体，主要包含以下五点基本需求：

1) **保密性：**规定了不同用户对不同数据的访问控制权限，仅有权限的用户才可以对数据进行相应的操作，信息不能被未授权的用户知晓和使用。具体要求区块链设置相应的认证规则、访问控制和审计机制。认证规则规定了每个节点加入区块链的方式和有效的身份识别方式。访问控制规定了访问控制的技术方法和每个用户的访问权限。审计监管是指区块链能够提供有效的安全事件监测、追踪、分析等一整套监管方案。

2) **完整性：**指区块链中的任何书籍不能被未经过授权的用户实施伪造、修改、删除等非法操作。具体指用户发布的交易信息不可伪造，经过全网共识的区块不可篡改伪造，智能合约的状态变量、中间结果和最终输出不可伪造。区块链系统中一切行为不可抵赖。完整性在交易底层需要数字签名，哈希函数等密码组件支持，在共识层面上，数据的完整性依赖于共识安全。

3) **可用性：**指数据可以在任何时间被有权限的用户访问和使用。区块链中的可用性包含以下四方面：首先，可用性要求区块链具备在遭受攻击时仍然具备能够提供可靠服务的能力。其次，可用性要求区块链受到攻击导致部分功能受损的情况下，具备短时间内修复和重构的能力。另外，可用性要求区块链可以提供无差别服务。最后，可用性需要保证用户的访问请求可以在有限时间内得到区块链网络响应，即需要区块链具有高吞吐量，低响应时延，可以提供稳定服务。

4) **不可抵赖性：**指区块链协议通信双方在信息交互过程中，确信参与者本身，以

及参与者所提供的信息的真实同一性，即所有参与者都不可能否认或抵赖本人的真实身份，以及提供信息、执行操作的真实痕迹。

5) **可控性：**指对流通在区块链系统中的信息传播及具体内容能够实现有效控制的特性，即区块链系统中的任何数据信息要在一定传输范围和存放空间内可控。

2.2 区块链数据隐私保护需求

在信息系统中，隐私是指数据拥有方不想透露的初始数据或者数据隐含的特征。而区块链技术为了维护分布式节点之间的数据同步实并达成全网共识，需要公开全网的交易信息，这样会给用户带来严重的隐私问题，因此，为了保护用户隐私以及降低隐私泄露的风险，必须对一些敏感数据进行相应的处理。在区块链中，隐私保护主要是针对用户身份信息和交易信息两部分内容，区块链的其余信息（如配置数据，合约数据等）对于隐私保护的需求相对而言并不高。因此，区块链的隐私保护主要可以划分为身份隐私保护和交易隐私保护。

1) **身份隐私保护：**身份隐私是指区块链地址关联于用户身份的隐私关系。区块链地址是用户在区块链系统汇总使用的昵称。无论用户身份信息是怎样的，用户都不需要第三方参与来创建和使用地址。而当用户使用区块链地址参与区块链服务时，个人的敏感信息存在被泄露的可能。因此，区块链身份隐私保护要求用户的身份信息、物理地址、IP 地址与区块链上的用户公钥、地址等公开信息之间是不关联的。任何未授权节点仅仅依靠区块链上公开的数据无法获取有关用户身份的任何信息，也不能通过网络监听，流量分析等网络技术对用户的身份和交易进行追踪。

2) **交易隐私保护：**交易隐私是指保存在区块链中的交易记录以及其背后隐藏的信息。在金融领域中，各种业务往来会产生大量的交易记录，这些交易记录是十分隐私且重要的数据，通过用户消费的交易记录很有可能推断出用户的收入水平、生活状态等隐

私信息，这会导致用户个人的敏感信息泄露。因此，交易隐私保护要求交易本身的数据信息对非授权节点匿名。在比特币中特指交易金额，交易的发送方公钥，接受地址等其他交易信息。任何未授权节点无法通过有效的技术手段获取交易相关的知识，在一些需要高隐私保护强度的区块链，还要求割裂交易与交易之间的关联性，即非授权节点无法有效推断出两个交易是否具有前后连续性，是否属于同一个用户等关联关系。

2.3 区块链数据安全与隐私保护的联系

区块链数据安全，是指在区块链中承载信息的数据的安全，保护区块链中数据安全就是需要保护包含区块数据、事务数据、交易数据、账户数据等采取不同的数据保障措施保证数据的安全。而目前，区块链发展迅速，区块链中的数据量逐步变大，数据的交互需求增多，数据安全与数据的使用呈现相伴相生的，只要有数据使用，就存在数据安全问题，而数据的使用是由业务所驱动。

区块链隐私保护是更关注某些特定的数据元素，如身份隐私，是指用户身份信息和区块链地址间的关联关系，用户在使用区块链地址参与区块链业务时可能会泄露敏感信息导致被用于推测区块链地址对应的真实身份；又如交易隐私，是指区块链中存储的交易记录等信息，通常交易记录也能反映出一些敏感知识，泄露用户的隐私。

3 区块链数据格式

区块链发展至今，按照区块链的开放程度主要分为公有链，联盟链，私有链三类，也是目前大众普遍接受的一个分类。在不同类型的区块链系统中，区块链的数据结构在细节上会存在差异，但在整体功能上基本是一致的。

一般情况下，区块链体系架构由自下而上的数据层、网络层、共识层、激励层、合约层和应用层六层组成。其中，网络层包括分布式组网机制、数据传播机制和数据验证机制等；共识层主要封装网络节点的各类共识机制算法，是区块链系统的核心技术；合约层主要封装各类脚本、算法和智能合约，是区块链可编程特性的基础；应用层则封装了区块链的各种应用场景和实例，如搭建在以太坊上的各类区块链应用就是部署在应用层；而数据层主要描述区块链的物理存在形式，包含了区块链上从创世区块起始的区块数据、链式结构以及区块上的随机数、时间戳、公钥、私钥数据等，是整个区块链技术中最底层的数据结构。

本章将从区块链数据服务的实体和面向的对象的角度，将区块链数据分为区块链业务数据、区块链支撑数据、区块链身份数据三大类，并进一步阐述区块链体系架构中数据层与其他各层的关联关系。区块链业务数据、区块链支撑数据和区块链身份数据之间的关系视图如图 3.1 所示。其中，业务数据和身份数据主要为区块链链上数据，而区块链支撑数据主要为区块链链下数据，部分数据由区块链链上数据提供完整性保障。

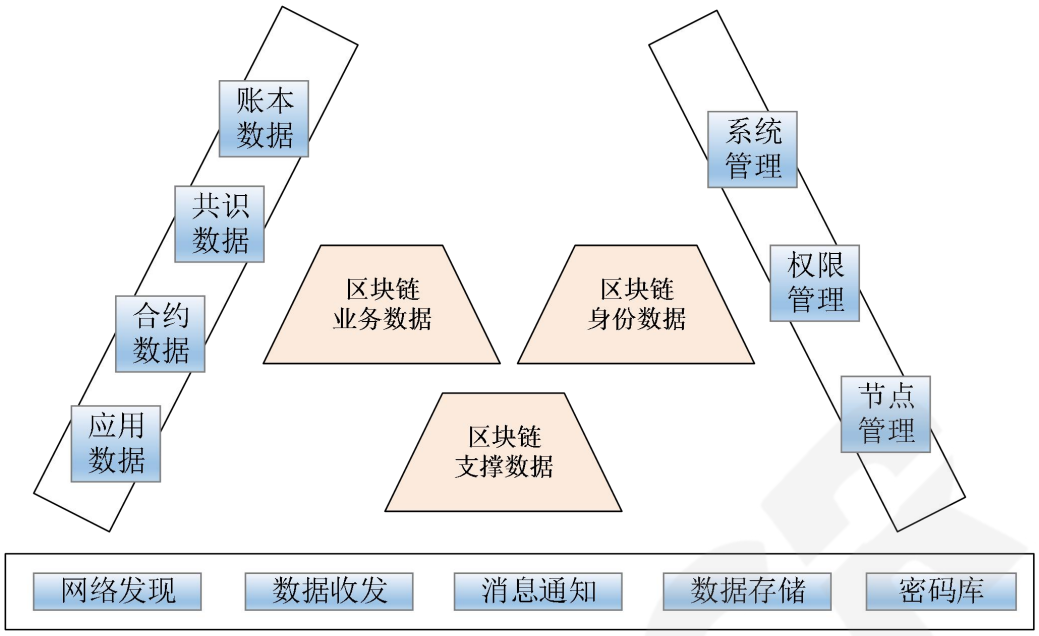

图 3.1 区块链业务数据、支撑数据与身份数据

区块链业务数据按具体的业务服务目标主要分为账本数据、共识数据、合约数据、应用数据。

(1) 区块链账本数据负责区块链系统的信息存储，包括收集交易数据，生成数据区块，对本地数据进行合法性校验，以及将校验通过的区块加到链上，将上一个区块的 Hash 签名嵌入到下一个区块中组成块链式数据结构，使数据完整性和真实性得到保障，这正是区块链系统防篡改、可追溯特性的来源。账本数据有两种数据记录方式：基于资产和基于账户。基于资产的模型中，首先以资产为核心进行建模，然后记录资产的所有权，即所有权是资产的一个字段。基于账户的模型中，建立账户作为资产和交易的对象，资产是账户下的一个字段。

(2) 区块链共识数据服务于协调保障全网各个节点数据记录的一致性，区块链系统中的数据由所有节点独立存储，在共识机制的协调下，共识层同步各节点的账本，从而实现节点选举、数据一致性验证和数据同步控制等功能。

(3) 区块链合约数据服务于将区块链系统的业务逻辑以代码的形式实现、编译并部

署，完成既定规则的条件触发和自动执行，最大限度的减少人工干预，主要内容包括智能合约代码、智能合约账户地址和数字资产描述等。

(4) 区块链应用数据服务于最终呈现给用户的部分，主要作用是调用智能合约层的接口，适配区块链的各类应用场景，为用户提供各种服务和应用。

区块链支撑数据主要服务于实现区块链系统中信息的记录、验证和传播的基础组件部分。

（1）**网络发现**：区块链网络系统由众多分散节点通过网络连接构成，网络发现支撑数据使得每个节点通过网络发现协议发现邻居节点，并与邻居节点建立链路。

（2）**数据收发**：节点通过网络通信协议连接到邻居节点后，数据收发支撑数据协助节点完成数据交换、区块广播，消息共识以及数据同步等。

（3）**消息通知**：消息通知相关数据为区块链中不同组件之间以及不同节点之间提供消息通知服务。交易成功之后，客户通常需要跟踪交易执行期间的记录和获取交易执行的结果，消息通知模块中消息的生成、分发、存储和其他功能都与消息通知数据密切相关。

（4）**数据存储**：根据数据类型和系统结构设计，区块链系统中的数据使用不同的数据存储模式，存储模式包括关系型数据库（比如 MySQL）和非关系型数据库（比如 LevelDB）。通常，需要保存的数据包括公共数据（例如：交易数据、事务数据、状态数据等）和本地的私有数据等。

（5）**密码库**：区块链的多个环节都涉及到密码学算法。密码库数据为上层组件提供基本的密码学算法支持，包括各种常用的编码算法、哈希算法、签名算法、隐私保护算法等等。与此同时，密码库数据还涉及诸如密钥的维护和存储等方面的功能。

区块链身份数据主要服务于系统管理、权限管理和节点管理三大类功能。**权限管理**是区块链技术的关键部分，尤其对于数据访问有更多要求的许可链而言，通过权限管理，

可以确保数据和函数调用只能由相应的操作员操作。通过将权限列表提交给账本层，可以实现分散权限控制，如实现可编辑的区块链功能。**节点管理**的核心是对节点标识的识别，通常由以下技术实现：(1)CA 认证，集中式颁发 CA 证书给系统中的各个应用程序，身份和权限由这些证书进行认证和确认；(2)PKI 认证，身份由基于 PKI 的公私密钥对确认；(3)第三方身份验证，身份由第三方提供的认证信息确认。由于各种区块链具有不同的应用场景，因此节点管理各有差异，但是区块链身份数据的存储、管理、维护和更新是区块链系统管理、权限管理和节点管理最根本和最重要的组成部分。

3.1 区块链业务数据

一般的区块链数据结构可以分为区块头数据和区块块数据两部分内容。图 3.2 描述了区块链块数据与区块头结构之间的关系。

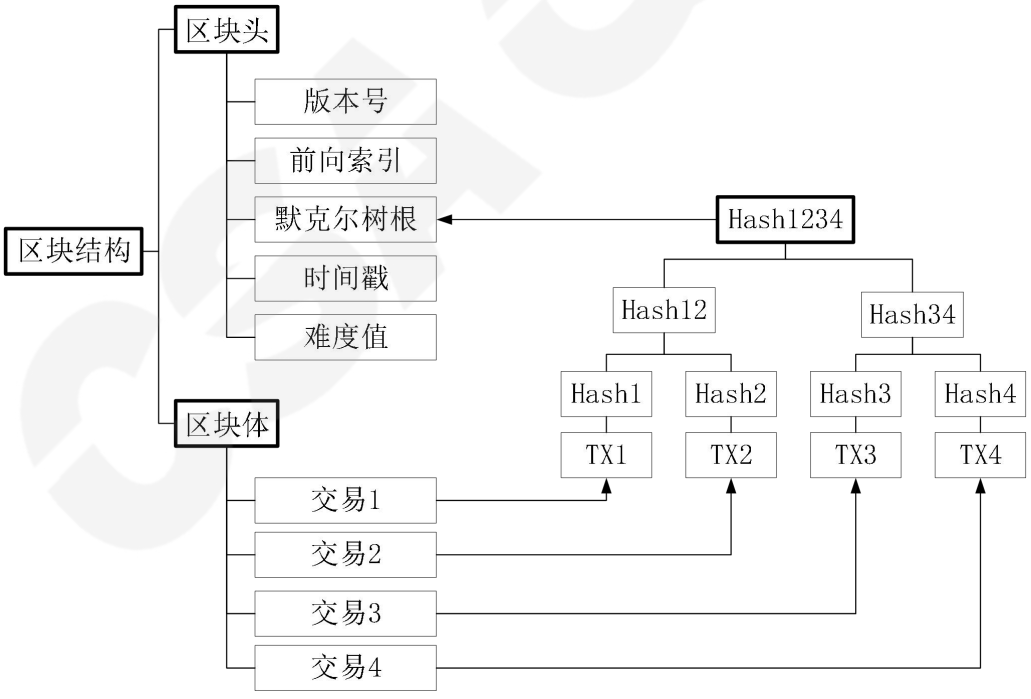

图 3.2 区块链块数据与区块头数据关系

3.1.1 区块链头数据

在一般的区块链系统中，区块头数据按照不同功能可以划分为以下三组数据，如表3.1 所示。第一组是用于连接前面区块、索引父区块哈希值的数据。第二组主要是描述区块信息以及显示区块状态信息的重要数据，如版本、区块序号、时间戳、区块生成时间、额外附加数据和难度等数据。第三组是与默克尔树相关的数据，用于快速归纳校验区块中的所有交易信息。区块链的大部分功能都由区块头实现。

功能数据分组	描述
第一组	连接前面区块、索引父区块哈希值的数据
第二组	描述区块信息以及显示区块状态信息的重要数据
第三组	与默克尔树相关的数据，用于归纳校验区块中的所有交易信息

表 3.1 区块头功能数据分组

以比特币区块链系统和以太坊区块链系统为例，表 3.2 和表 3.3 分别描述了比特币区块头和以太坊区块头的数据结构。

功能数据分组	字段	描述
第一组	父区块哈希值	引用区块链中父区块的哈希值
第二组	版本	版本号，用于跟踪软件/协议的更新
	时间戳	该区块产生的近似时间
	难度目标	该区块工作量证明算法的难度目标
	Nonce	用于工作量证明算法的计数器
第三组	Merkle 根	该区块中交易的默克尔树根的哈希值

表 3.2 比特币区块头数据结构

功能数据分组	字段	描述
第一组	ParentHash	父区块的哈希值
	UncleHash	叔区块列表哈希值
	Coinbase	矿工接受奖励的地址
	Difficulty	区块难度目标
	Number	区块号,从 0 号开始算起

第二组	GasLimit	理论 Gas 上限
	GasUsed	区块内所有交易执行产生的 gas 总和
	Nonce	随机数，工作量证明算法的计数器
	Time	区块创建的时间
	Bloom	交易收据日志组成的布隆过滤器
	Extra	区块的附加数据
第三组	Root	状态树的根哈希值
	TxHash	交易树的根哈希值
	ReceiptHash	收据树的根哈希值

表 3.3 以太坊区块头数据结构

3.1.2 区块链块数据

区块链块数据，也被称为区块链体数据，其中主要的部分是该区块打包的所有交易记录。这些所有的交易将通过 Merkle 树的 Hash 过程产生一个唯一 Merkle 根值记入区块头。

对于一般的区块链系统而言，一个区块链块数据中的交易可以分为五类数据，详情参照表 3.4。

字段	描述
版本/类型	交易的类属或参照的版本规则
输入信息	交易输入的数量，哈希值，序号，交易单价、数量等
输出信息	交易的接收方地址，交易费用总量等
附加数据	与区块链交易相关的额外数据
其他数据	不同类型区块链支持功能的数据

表 3.4 区块链块数据内容

以比特币系统为例，每笔交易的详细信息如表 3.5 所示。

参照	字段	描述
版本/类型	版本	比特币交易参照规则
输入信息	输入计数器	包含交易的输入数量
	输出位置索引	被花费的 UTXO 的索引号，第一个为 0

	交易哈希值	指向被花费的 UTXO 所在的交易的哈希
	序列号	固定值 0xFFFFFFFF
	解锁脚本相关	解锁脚本大小，需要满足的条件等
输出信息	总量	用“聪”表示的比特值
	锁定脚本相关	锁定脚本大小，需要满足的条件等
其他数据	锁定时间	一个区块号或 UNIX 时间戳

表 3.5 比特币块数据内容

在以太坊系统中，块数据交易详情如表 3.6 所示。

参照	字段	描述
版本/类型	Type	交易类型，创建合约还是调用合约
输入信息	Nonce	交易次数
	Value	需要交易的以太坊数量
	Gasprice	交易的 gas 价格
	GasLimit	执行交易可使用的最大 gas 量
输出信息	R_Address	交易接收方地址
	Gas	交易付出的 gas 总量
附加数据	Data	交易的附加数据
其他数据	VRS	交易签名
	Init	合约创建时的初始化代码
附加数据	Data	与交易相关的附加数据

表 3.6 以太坊块数据内容

3.1.3 特殊区块链数据结构

上述描述的比特币区块链系统和以太坊区块系统，这类公有链的区块结构一般分为区块头数据和区块块数据，而在联盟链的 Fabric 区块链系统中，区块的数据结构分为三部分，区块头（Header），区块数据（Data）和与当前区块相关的元数据（MetaData）。由于在联盟链中，各项交易的需求和面向的对象与私有链不同，因此 Fabric 区块链在数据结构上会存在差异。Fabric 区块数据结构如表 3.7。

区块结构	字段	描述
Header	Number	区块编号
	PreviousHash	上一个区块头的哈希值
	DataHash	上一个区块数据的哈希值
Data	Type	交易类型
	Version	交易版本
	Timestamp	时间戳
	Channel	交易的通道
	TxId	交易编号
	Chaincode Info	链码信息
	Read Set	读取集
	Write Set	写入集
	Endorser_nodeInfo	背书节点信息
MetaData	Creator	消息创建者身份
	Signature	签名
	Configuration	配置
	TxFlag	交易标记

表 3.7 Fabric 区块结构内容

Fabric 区块头包含三个字段，当前区块号（number）、前一个区块头哈希（previous_hash）和当前区块的数据哈希（data_hash），其中 data_hash 字段并非当前区块的哈希值，而只是当前区块数据体的哈希值。

Fabric 区块块数据 Data 包含展示交易信息的数据结构 Envelope 数据，此数据类型主要用于存储区块中的交易信息，交易信息主要包括两个字段，交易发送者的签名（signature）以及数据载荷（payload）。

Fabric 区块元数据部分包含与当前区块相关的元数据内容，用于描述 Data 的相关信息，包含排序节点的 MspId（Fabric 区块链以发放证书的方式进行节点身份管理，Msp 为成员管理服务提供商），最新的配置块信息，消息创建者的证书、公钥以及签名值。

3.2 区块链支撑数据

3.2.1 业务支撑数据

支撑区块链协议执行的数据内容，如交易缓存池、状态前缀树、转账前缀树、回执前缀树等。业务支撑数据遭受安全威胁时，将破坏区块链协议的正常执行，降低区块链运行效率。

3.2.2 网络支撑数据

支撑区块链网络发现与节点连接的数据内容，如目录服务器地址，可连通节点地址和候选节点地址。例如比特币区块链中的 tri 表数据包和 new 表数据包，以太坊区块链中的网络状态前缀树。网络支撑数据遭受安全威胁时，将破坏区块链节点的连通能力，导致可能遭受隔离攻击和延时攻击等攻击。

3.2.3 存储支撑数据

支撑区块链临时信息存储和永久信息存储的组件数据内容，如以太坊保存着三种树，分别为状态树、交易树和收据树。

状态数据存储：整个以太坊系统中只有一棵状态树，记录整个以太坊系统的所有账户状态。每个区块保存着一棵交易树，记录该区块的交易情况，一棵收据树用来记录该区块的交易收据。状态树采用 Merkel-Patrica(MPT)树，而交易树和状态树采用 Merkel 树。

底层数据存储：以太坊中共有三个 LevelDB 数据库，分别是 BlockDB、StateDB 和 ExtrasDB。BlockDB 保存了块的主体内容，包括块头和交易；StateDB 保存了账户的状态数据；ExtrasDB 保存了收据信息和其他辅助信息。

存储支撑数据遭受安全威胁时，将破坏所存储信息的真实性和完整性，降低各类交易验证和身份验证速度。

3.2.4 应用支撑数据

支撑区块链各类具体应用服务的组件，如可用的 API 接口，合约调用指向的地址等。应用支撑数据遭受安全威胁时，将影响区块链应用服务的效果，甚至导致数字资产流失等后果。

3.3 区块链身份数据

在现实世界或者网络世界要证明自己身份需要提供政府出具的身份证或者是个人的社交账号，而在区块链中，要证明自己的身份需要依靠加密算法、数字签名和数字证书所形成的一个体系来证明身份。区块链的身份数据就是指在区块链中这类带有身份标识属性的数据。区块链身份的证明让区块链满足了公信力的需求，信任不是由政府或者第三方组织等掌控，而是建立在区块链上。区块链对身份数据的管理不是放在中央数据库中，而是利用用户的设备来存储和加密数据，并使用加密来连接多个虚拟网络中的数据块。这确保每个信息块存储所有记录的完整并确保副本准确，这些记录被更改或误用会被网络发现。

区块链身份数据往往是跟用户的信息数据和交易转账等直接关联，区块链身份数据的一个体现是交易额存储的地址，就比如在比特币系统中的比特币地址；以太坊中的合约账户等提供了区块链交易额存储的地址。

比特币地址是由公钥经过 SHA-256 哈希运算后，再通过 RIPEMD-160 哈希运算得到的，定长为 160bit 的地址。用户通过锁定脚本的方式将比特币锁定到某个比特币地址，持有该地址私钥的用户则相当于拥有拿到这些比特币的钥匙。比特币地址就相当于用户

的钱包。通过图 3.3 来举例描述比特币地址的身份属性。用户 A 需要向用户 B 转账 50BTC，其做法是将 50BTC 放入区块链并用锁定脚本将这笔钱锁定到用户 B 的比特币地址。用户 B 通过提供签名和公钥来获取和使用该比特币地址里面的比特币。

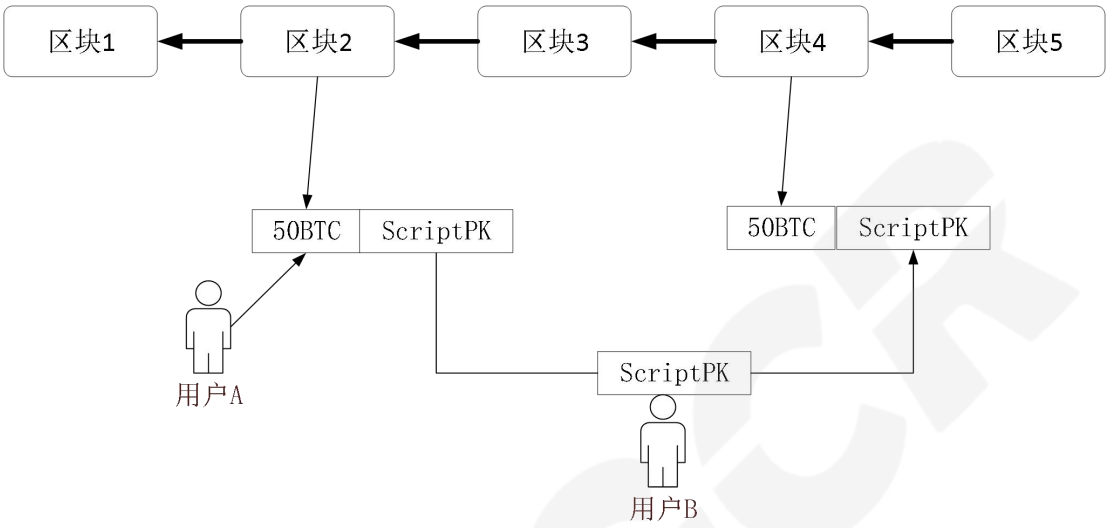

图 3.3 比特币地址示例

同样，在以太坊智能合约里，先要构建数字身份认证服务，如以生物学信息或密码学标识的特有特征信息去构建的数字身份概念。有了数字身份才可以构建合约主体。智能合约里需要去除掉现实世界中如银行、房管所一类的中心化的登记机构，设法让合约里的权益和资产数字化。有了数字身份和资产数字化后，就可以构建两者的映射关系，也就可以使用数字身份去绑定数字化的资产，然后才可以使用代码的形式来执行。

以太坊合约账户是在在以太坊区块链中，部署智能合约时生成由代码控制的账户。合约账户可以通过设置多重签名来体现合约账户的身份属性，如图 3.4。合约账户要求一笔转账需要转账人用户 A 与用户 C 的签名，当用户 A 需要向用户 B 转账 20 个 ETH 时，合约会通知用户 C 签名，批准签名后，用户 B 则可以得到这 20 个 ETH。

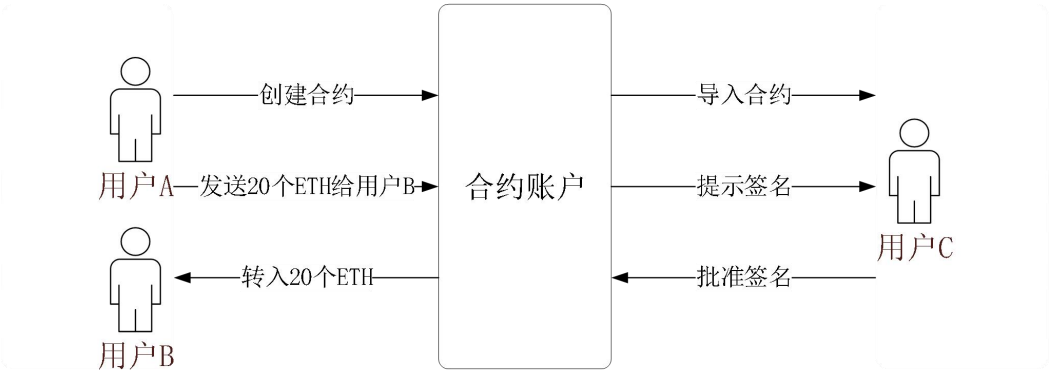

图 3.4 以太坊合约账户示例

区块账户数据主要包括由 PKI 体系为账户生成的账户公私钥信息，账户拥有的资产说明以及数字证书等。区块链一系列交易的进行离不开账户，各个账户通过公私钥机制以及数字签名方式完成交易并通告全网。保证账户数据信息安全是区块链应用需要格外关注的。

4 数据层安全与隐私保护

4.1 数据安全与隐私保护总体框架

区块链隐私保护架构可以根据隐私保护的范畴分为区块链内容隐私保护、区块链身份隐私保护和区块链行为隐私保护。

1) **区块链内容隐私保护：**个人、机构等实体不愿意被外部世界知晓的信息，包括敏感数据以及数据所表征的特性。

2) **区块链身份隐私保护：**可识别的个人数据，以及可以直接定位到具体自然人的信息都属于隐私保护的范畴。

3) **区块链行为隐私保护：**不会直接泄露用户隐私，但可以据此间接推断出个人行为轨迹、趋势或行为特征的信息内容也在区块链隐私保护架构之下。

4.2 接入层设计指南

接入层需要确保终端在接入区块链平台时经过严格的标识和认证，防止伪造和假冒，实现接入安全需要具有可靠的密钥管理机制，从而对实现并支持用户、节点接入过程中安全传输的能力，并能够阻断异常的接入；需考虑：

1) **身份认证：**需考虑应用用户的身份认证，防止身份伪造，确保用户仅访问其授权的资源；

2) **访问控制：**需考虑用户与系统资源的访问策略，严格限制用户访问的系统权限；

3) **WEB 应用攻击防护：**需考虑 WEB 应用可能面临的 SQL 注入、跨站脚本、信息泄露、恶意代码等攻击行为；

4) **APP 安全：**需综合考虑 APP 面临的移动安全问题，包括不安全传输、信息泄露、反编译等攻击行为。

4.3 处理层设计指南

4.3.1 数据加密

要保证在公开信道上传输或在公共可访问区域内存储的消息不泄露给非目标用户，通常应采用数据加密技术。简单来说，数据加密技术是将要保护的消息，又称为明文，变换为非目标用户无法理解的形式，又称为密文。这一过程称为加密，加密过程中需要使用加密密钥。密文可由拥有解密密钥的目标用户还原为明文，非目标用户因为没有解密密钥，所以无法获得明文，从而保障了消息的机密性。

根据加密密钥与解密密钥是否相同，数据加密算法可分为对称加密和非对称加密两类。对称加密中的两种密钥相同，而非对称加密密钥中的两种密钥不同，并且很难通过加密密钥计算出解密密钥。一般来说，对称加密的速度较快，适合加密大量数据。但是对称加密面临着密钥共享的困难，即消息的收发双方如何安全地共享加密/解密密钥。这一困难在互联网环境中尤为突出，用户无法在互联网的开放环境中直接秘密地传输加密/解密密钥。相对的，非对称加密速度较慢，但是不需要担心密钥共享的问题，因此通常用于在互联网环境中传输对称加密的密钥。

1) 对称加密

a) 对称加密算法流程：

进行对称加密时，首先要生成密钥。加密时，加密方使用密钥对明文加密，得到密文；解密时，解密方使用密钥对密文解密，得到明文。

b) 分组模式：

一般来说，对称加密使用的加密算法只能加密一定长度（称为分组大小）的明文（称为分组）。实际情况中，明文长度往往远大于分组大小。需要使用分组模式加密多个分组。

c) CTR 模式：

加密过程:CTR 模式为每个分组生成相应的掩码，每个分组再与相应的掩码进行异或操作，实现将明文加密成密文。生成掩码时，首先生成随机初始化向量 ctr，对(ctr+i)加密的结果作为第 i 个分组的掩码，以此实现对所有分组加密。

解密过程:解密方首先计算每个分组相应的掩码，再将掩码与密文异或即可得到明文。

d) CBC 模式：

加密过程：CBC 模式加密将明文从前向后加密。加密第 i(i > 1)个明文分组时，首先将第 i 个明文分组与第 i-1 个明文分组的加密密文异或，再使用加密算法将异或结果加密，作为密文。加密第 1 个明文分组时，首先将其与一个随机生成的初始化向量异或，再使用加密算法将异或结果加密，作为密文。

解密过程：CBC 模式将分组从后向前依次解密，解密第 i(i > 1)个密文分组时，首先使用密钥将其解密，再将解密结果与第 i-1 个密文分组异或，即可得到明文。解密第 1 个密文分组时，首先使用密钥将其解密，再将解密结果与初始化向量异或，即可得到第 1 个明文分组。

e) OFB 模式：

加密过程：OFB 模式与 CTR 模式类似。首先为每个分组生成相应的掩码，每个分组再与相应的掩码进行异或操作，实现将明文加密成密文。生成掩码时，首先生成随机初始化向量，以递归方式对其加密，递归加密 i 次作为第 i 个分组的掩码。以此实现对所有分组加密。

解密过程：如加密过程，依次求出各分组对应的掩码，将掩码与对应密文异或，即可得到对应明文。

2) 非对称加密

非对称加密中，加密密钥与解密密钥不同。加密密钥称为公钥，是公开的。解密密

钥称为私钥，一般来说只有目标用户持有。这一特性使得通信双方不用秘密共享密钥。消息发送方可以使用公开可见的公钥加密明文。加密得到的密文只有持有私钥的接收方可以解密。

进行非对称加密时，首先生成加密、解密所需密钥，即公钥 pk 和私钥 sk；加密时，加密方使用公钥 pk 将明文加密成密文；解密时，解密方使用私钥 sk 将密文解密成明文。

4.3.2 匿名签名

在隐私敏感安全应用中，用户既希望通过签名操作验证身份的合法性，使用签名操作的可追溯和不可抵赖功能，同时又希望在签名及验证过程中不泄露身份信息，让自己的身份既真实又不可公开。匿名签名机制是实现网络环境下隐私保护的主要技术途径。群签名和环签名可以给数字签名增加匿名性。在群签名和环签名中，成员可以以匿名的方式代表整个群体对消息签名，这个签名是可公开验证的。

1) 群签名

在一个群签名方案中，一个群体中的任意一个成员可以以匿名的方式代表整个群体对消息进行签名。与其他数字签名一样，群签名是可以公开验证的，且可以只用单个群公钥来验证。当签名存在争议时，群管理者可以确定签名者的身份。

一个好的群签名方案应满足以下的安全性要求：

a) 匿名性:给定一个群签名后，对除了唯一的群管理员以外的任何人来说，确定签名者的身份是不可行的，至少在计算上是困难的。

b) 不关联性:在不打开签名的情况下，确定两个不同的签名是否为同一个群成员所签的是不可行的，至少在计算上是困难的。

c) 不可伪造性:只有群成员才能产生有效的群签名。其他任何人包括群管理员也不能伪造一个合法的签名。

d) 可跟踪性:群管理员在发生纠纷的情况下可以打开一个签名来确定出签名者的

身份，而且任何人都不能阻止一个合法签名的打开。

e) 防陷害性：不论是群成员，还是管理员，都不能代表其他群成员对消息签名。

f) 抵抗联合攻击:即使一些群成员串通在一起也不能产生一个合法的不能被跟踪的群签名。

群签名通常包含以下过程：

在系统初始化时，生成群公钥和群私钥；当一个用户加入系统时，用户和群管理员使用规定的交互式协议，产生群员的私钥和成员证书，并使群管理员得到群成员的私有密钥；当群成员签名时，群成员使用自己的私钥对消息进行签名；验证签名时，验证者使用群公钥验证签名是否由群成员所生成；在需要时，群管理员可以根据签名和群私钥确定签名者的合法身份。

2) 环签名

环签名和群签名相似，但相比于群签名，环签名中只有环成员，没有管理者。

好的环签名方案应满足以下安全性要求：

a) 正确性：如果按照正确的签名步骤对消息进行签名，并且在传播过程中签名没被篡改，那么环签名满足签名验证等式。

b) 无条件匿名性：攻击者即使非法获取了所有可能签名者的私钥，他能确定出真正的签名者的概率不超过 1/N，这里 N 为所有可能签名者的个数。

c) 不可伪造性：外部攻击在不知道任何成员私钥的情况下，即使能从一个产生环签名的随机预言者那里得到任何消息 m 的签名，他成功伪造一个合法签名的概率也是可以忽略的。

一般环签名主要有以下过程：

在系统初始化时，根据安全参数生成公钥和私钥。为每一个成员 u_i，产生一个公钥 y_i 和私钥 x_i；成员进行签名时，成员根据自己的私钥 x_i、环成员的公钥 $L=\{y_1,y_2,...,y_n\}$ 对消息产生签名 R，其中 R 中的某个参数根据一定的规则呈环状；验证时，验证者根据

环成员的公钥对签名进行验证。

4.3.3 多重签名

在数字签名应用中，有时需要多个用户对同一个文件进行签名和认证，这就需要多重签名。简单来说，多重签名是多个用户对同一个消息进行数字签名。

多重数字签名可分为有序多重数字签名和广播多重数字签名。它们都有三个过程：系统初始化、产生签名和验证签名；它们应用场景中都有三种角色：消息发送者、消息签名者和签名验证者。在广播多重数字签名方案中还包含有签名收集者。

1) 有序多重数字签名

在有序多重数字签名中，消息发送者U_s预先给消息签名者规定好一个签名顺序，记为$\{U_1,U_2,...,U_n\}$，U_i代表第 i 个消息签名者。将此顺序发送给每一位签名者和消息验证者U_v。签名者按照此顺序进行签名。签名结束后，消息验证者U_v对签名进行验证。

2) 广播多重数字签名

广播多重数字签名与有序多重数字签名不同，消息发送者U_s将 m 同时发送给所有消息签名者进行签名，消息签名者将签名的结果发送到签名收集者U_c，U_c收到签名信息后进行计算整理，再发送给消息验证者U_v进行验证。

4.3.4 安全多方计算

安全多方计算协议要解决的问题可以描述如下：一组参与者希望共同计算某个约定的函数，每个参与者提供函数的一个输入，这个输入要对其他人保密。如果存在安全可信第三方，则安全多方协议所要解决的问题可以轻易地得到解决：只需各参与者将各自的输入交给安全可信第三方，由安全可信第三方计算出函数值，再将函数值公布给各参与者。但现实中很难找到这样的安全可信第三方，安全多方计算协议的研究应运而生。

目前安全多方计算已得到许多学者的研究，其在密码学上的地位也日益重要，它是电子选举、电子拍卖等密码学协议的基础。

1) 数学描述及算法流程：

有 *n* 个参与者 P1, P2,..., Pn，要以一种安全（安全意味着输出结果的正确性以及输入信息、输出信息的保密性）的方式共同计算一个函数。每个参与者 Pi($0 < i <= n$)有一个自己的保密输入信息 *Xi*，*n* 个参与者要共同计算一个函数 F:(*X1*, *X2*,…, *Xn*) -> (*Y1*, *Y2*,…, *Yn*)，计算结束时，参与者 Pi 只能了解 *Yi*，不能了解其他方的任何信息。大多数情况下 *Y1* = *Y2* =...= *Yn* = *Y*，此时 F:(*X1*, *X2*,…, *Xn*) -> *Y*。

2) 安全多方计算主要特点：

安全多方计算理论主要研究参与者间协同计算及隐私信息保护问题，其特点包括输入隐私性、计算正确性及去中心化等特性。

输入隐私性：安全多方计算研究的是各参与方在协作计算时如何对各方隐私数据进行保护，重点关注各参与方之间的隐私安全性问题，即在安全多方计算过程中必须保证各方私密输入独立，计算时不泄露任何本地数据。

计算正确性：多方计算参与各方就某一约定计算任务，通过约定安全多方计算协议进行协同计算，计算结束后，各方得到正确的数据反馈。

去中心化：传统的分布式计算由中心节点协调各用户的计算进程，收集各用户的输入信息，而安全多方计算中，各参与方地位平等，不存在任何有特权的参与方或第三方，提供一种去中心化的计算模式。

4.3.5 可验证计算

外包计算使得具有有限计算资源的用户可以将大规模的计算任务外包给拥有强大计算能力的服务提供者，例如云计算场景中用户将负载较大的计算任务外包给云服务的提供者。然而外包计算场景中用户面临着一定的风险：对于外包给服务提供者的计算任

务，用户无法确认其是否得到正确的执行。

可验证计算协议可检测出远程服务器返回的程序执行结果是否正确，且不需要将远程服务器所执行的程序再重新执行一遍。使用可验证计算协议是解决外包计算的结果正确性验证问题的有效途径。

在可验证计算模型中，客户端将计算外包给服务器端；服务器端运行这些繁重的计算，并返回计算结果和证明；验证者可以根据计算结果和证明验证服务器端返回的结果的正确性。根据验证者的身份，可以将可验证计算方案分为私有可验证计算和公开可验证计算。私有可验证性约束了验证者的身份必须是将任务外包出去的客户端本身。相对于私有可验证性，具有公开可验证性的可验证计算方案应用更为广泛，它不仅支持客户端本身验证结果，同时支持任何拥有相关公开参数的第三方验证结果。

1) 可验证计算的基本条件

a) 正确性：服务器按照方案诚实运算得到的结果，返回给客户端一定能够通过验证结果。

b) 安全性或可靠性：安全或者可靠的可验证计算方案保证客户端不会接受服务器返回的错误。

c) 高效性：方案中规定的客户端对函数和输入预处理所花费的时间，连同随后进行的可靠性验证所花费的时间，要远小于直接计算函数所需要的时间，否则也就失去了委托计算的意义。

除了以上 3 个基本要求外，数据的隐私性也是可验证计算中考虑的主要因素之一。可验证计算所涉及的隐私性主要有两种，一种是客户端的数据相对于服务器的隐私保护，另一种是最终计算结果相对于非授权用户的隐私保护。

2) 常用可验证计算

a) 可私自验证计算

可私自验证计算是指客户端委托服务器计算函数 F 关于 x 点的值，客户端会对函数 F 和 x 进行预处理，得到可以公开的信息和自己保存的私密信息。客户端将公开信息传给服务器，而不泄漏私密信息。服务器使用公开信息进行计算，将结果返回给客户端。对于返回的计算结果，客户端利用前面保存的私密信息，对服务器返回的结果进行验证。

b) 可公开验证计算

可私自验证计算协议执行过程中产生的信息不对外公开。上述可私自验证方案中，客户端一般每委托服务器计算一个新的函数，就得预处理一次，因此只有客户端委托服务器计算同一函数关于多个不同输入的值时，算法才更有意义。可公开验证计算可以使客户端不一定对函数进行预处理，可以利用其他客户端的预处理结果进行委托计算。

可公开验证计算有两个性质：公开代理和公开验证。

公开代理是指一个客户端对函数 F 作预处理之后将信息公布，其他客户端可以直接利用这些信息，只需要对函数的输入作处理便可将计算任务委托给服务器；公开验证是指任何人都可以验证服务器返回的结果。

可公开验证计算过程相比于可私自验证计算的过程，客户端预处理的结果是一组公开值，其他客户端可以使用这些公开值委托服务器计算并验证。

4.3.6 零知识证明

零知识证明指的是证明者能够在不向验证者提供任何有用的信息的情况下，使验证者相信某个论断是正确的。零知识证明实质上是一种涉及两方或更多方的协议，即两方或更多方完成一项任务所需采取的一系列步骤。证明者向验证者证明并使其相信自己知道或拥有某一消息，但证明过程不能向验证者泄漏任何关于被证明消息的信息。

零知识证明并不是数学意义上的证明，因为它存在小概率的误差，欺骗者有可能通过虚假陈述骗过证明者。换句话来说，零知识证明是概率证明而不是确定性证明。但是也存在有技术能将误差降低到可以忽略的值。

1) 零知识证明的性质

a) 正确性:证明者无法欺骗验证者。换言之，若证明者不知道一个定理的证明方法，则证明者使验证者相信他会证明定理的概率很低。

b) 完备性:验证者无法欺骗证明者。若证明者知道一个定理的证明方法，则证明者使验证者以绝对优势的概率相信他能证明。

c) 零知识性:验证者无法获取任何额外的知识。

2) 零知识证明的过程

零知识证明本质上是一种涉及两方的协议，其中的一方称为证明者，一般用 P 表示，另一方称为验证者，一般用 V 表示。在协议的执行过程中，证明者 P 向验证者 V 声称其已经掌握了某种信息，证明者 P 根据公共数据 *u* 和隐私数据 *w*，生成验证结果 *y* 和证明 *Ty*，并将 *y* 和 *Ty* 发送给验证者 V。验证者 V 根据 *y*、*Ty* 和 *u* 进行验证，或者相信证明者 P 的声称，或者拒绝证明者 P 的声称。在这个过程中，验证者 V 没有获得证明者 P 声称的所掌握信息的具体内容。

4.3.7 不经意访问

1) 不经意随机访问的产生背景

随着大数据与云计算技术的发展，越来越多的数据可以在云端进行存储、计算以及共享。但是这也带来了一系列安全问题。加密只能保证数据内容的机密性，却不能保证其他的隐私信息不被泄露。

在云存储场景中，用户需要查询某一个数据块的内容。为了保证数据隐私性，通常会对数据库中的数据进行加密。但是在执行查询请求的过程中，数据块的索引并不能加密，这就泄露了用户访问数据库元素的位置。这使得攻击者可以通过用户的访问模式来

推断存储数据的重要性。同时，攻击者还可以通过匹配前后两个连续的访问模式来推断数据查询之间的关联关系，甚至是加密数据的内容。

在实际场景中，这些信息潜在性地暴露了用户的行为特征、兴趣爱好、社交范围等，另一方面，在安全计算领域，当用户将隐私数据和任务以加密方式存储在内存中，如果暴露了处理器对内存的访问模式，就可能泄露数据和任务本身的信息。

2) 不经意随机访问的概念

不经意随机访问机是一种重要的保护访问模式的手段，它通过混淆每一次访问过程，使其与随机访问不可区分，从而保护真实访问中的访问操作、访问位置等信息。不经意随机访问机在安全云存储系统以及安全计算领域有着非常重要的作用。利用不经意随机访问机可以降低攻击者通过访问模式推测隐私信息的可能性，减小系统受到的攻击面，从而提供更安全更完整的服务。

3) 不经意随机访问特性

a) 低效性：相比于正常的访问，不经意随机访问机需要执行额外的操作来保护访问模式的隐私性，这种访问模式往往带来昂贵的开销，包括带宽以及本地存储等，这严重的限制了不经意随机访问机的实用性；

b) 安全性：不经意随机访问机提供了保护访问模式的可证明安全性，相比于传统的加密手段，该技术可以很大程度上减少攻击者利用访问模式推断隐私信息的可能性；

c) 用途广：不经意随机访问机可以广泛地应用于安全存储以及安全计算领域，对于存在数据访问的应用，都可以利用 ORAM 提供访问模式的保护。

4) 不经意随机访问几种模型

a) 简单模型：服务器以数组的方式连续存储客户端的数据块。为了隐藏客户端访

问了哪一个数据块，客户端每一次访问需要遍历所有数据块。对于不需要的数据块，客户端读取之后再写回相同的数据块；对于目标数据块，客户端读取后在本地更新，再写回更新后的数据块。同时，即使客户端找到目标数据块，依然继续访问；

b) 平方根模型：将服务器划分成两部分：排列数组(permuted array)与缓冲区(shelter)。排列数组中包含N个真实数据块(real block)和$\sqrt{N}$个无效数据块(dummy block)。客户端每次访问之前需要将排列数组中的所有的数据块混洗。在每一次访问中，客户端先查找目标数据块是否在缓冲区中：如果在缓冲区中，就从排列数组中读取一个无效数据块；如果没有在缓冲区中，就从排列数组中读取目标数据块。为了混淆数据块的访问位置，每一个访问周期($\sqrt{N}$次访问)需要重新混洗排列数组；

c) 层次模型：将服务器的存储划分为层，第i层包含2^i个数据块集合(bucket)，对于每一层而言，当一个访问周期(2^i次访问)结束后，需要将当前层的数据块和下一层的数据块合并并且混洗后放入下一层。每一层实际上都是一个 hash 表，并且包含一个 hash 函数。每一层的访问周期结束后，需要更换 hash 函数。当客户端访问一个数据块时，从最顶层到最底层依次查找，通过计算 hash 函数来判断目标数据块是否在当前层：如果在，则根据 hash 函数计算所得的偏移量获得对应数据块；如果不在，则继续查找下一层。与简单模型类似，为了保护访问模式的隐私，即使找到了目标数据块也会继续查找下一层，之后，每一层会随机访问一个无效数据块，直到所有层都被访问。当客户端更新完数据块后，将其写入服务器最顶层。由于顶层的访问周期短，其中的数据块会频繁地混洗到下一层，因此不用担心顶层数据块溢出的问题；

d) 分区模型：将数据存储在$\sqrt{N}$个服务器(partitions)中，每一个服务器利用平方根模型或者层次模型构建，同时，客户端存储数据块索引到数据块在服务器位置之间的映射表以及每一个服务器的缓存(cache slots)。对于客户端每一次访问，先根据映射表查找数据块所在的服务器，然后使用对应不经意随机访问机模型的访问方式获取数据块。

当获得数据块之后，将其重新分配一个新的服务器，并写入对应服务器的缓存中。缓存满之前执行将数据写回对应的服务器；

e) 树状模型：在层次模型上进行改进，将每一个数据块集合分配到树的节点上，客户端本地存储每一个数据对应树的叶子节点的映射关系(position map)。每一次访问先查询数据块所在的叶子节点，将从根节点至这个叶子节点上所有的数据块集合取回本地。更新完之后，将目标数据块写入根节点。每一次访问结束之后，在每一层随机选择 v 个数据块集合执行驱逐操作，将一个真实数据块写入它对应叶子节点那条路径上的子节点数据块集合中。同时，再选出一个无效数据块写入它另一个子节点的数据块集合中。

4.4 展示层设计指南

4.4.1 数据发布

与传统针对隐私保护进行的数据发布手段相比，区块链数据发布面临的风险是区块链的发布是动态的，且针对同一用户的数据来源众多，总量巨大。需要解决的问题是在数据发布时，保证用户数据可用的情况下，高效、可靠地去掉可能泄露用户隐私的内容。传统针对数据的匿名发布技术，包括 k-匿名、l-diversity 匿名、t-closeness 匿名、个性化匿名、m-invariance 匿名、基于“角色构成”的匿名方法等，可以实现对发布数据时的匿名保护。在区块链应用环境下，需要对这些数据进行改进和发展。

匿名技术：数据持有方在公开发布数据时，这些数据通常会包含一定的用户信息，服务方在数据发布之前需要对数据进行处理使用户隐私免遭泄露。一般的，用户更希望攻击者无法从数据中识别出自身，更无法窃取自身的隐私信息。

数据发布匿名：在确保所发布的信息数据公开可用的前提下，隐藏公开数据记录与特定个人之间的对应联系，从而保护个人隐私。实践表明，仅删除数据表中有关用户身

份的属性作为匿名实现方案是无法达到预期效果的。现有的方案是静态匿名技术、个性化匿名、带权重的匿名等。后两类给予每条数据记录以不同程度的匿名保护，减少了非必要的信息损失

4.4.2 安全审计技术

当用户将数据存储在区块链分布式服务器中时，就丧失了对数据的控制权。为了防止数据在用户不知情的情况下被修改，可以采用类似于云存储中的审计技术。安全审计指的是数据拥有者或第三方机构对存储结构中的数据完整性进行审计。通过对数据进行审计，确保数据不会被服务提供商篡改、丢弃，并且在审计的过程中用户的隐私不会被泄露。区块链环境下可用的安全审计技术主要有：

可证明的数据持有模型（PDP）：该模型可以对服务器上的数据进行完整性验证，该模型中挑战应答协议传输的数据量非常少，因此所耗费的网络带宽较小。

可恢复证明模型（POR）：利用纠错码技术和消息认证机制来保证远程数据文件的完整性和可恢复性。该模型面临的挑战在于需要构建一个高效和安全的系统来应对用户的请求。

5 区块链数据层测试指南

5.1 性能测试

5.1.1 数据处理吞吐量测试

测试对象：吞吐量指一次性能测试过程中在区块链上传输的数据量总和，而单位时间内链上传输的数据量称为吞吐率，这个指标是衡量区块链网络性能的重要指标，也可表示为是区块链系统每秒能处理的数据量。

测试方法：通过模拟多客户端并行向区块链中发送交易，每笔交易中带有不等的数据量，统计发送的总交易数 *T*，发送的交易中成功处理的交易数量 *S*，完成所有交易花费的时间 *t*。交易处理的吞吐量 tps=*S/t*，成功率 $S_{rate}=S/T$.

测试目的：吞吐量指标反映的是区块链服务器承受的压力，能够说明系统的负载能力。通过测试区块链的数据处理吞吐量，测试区块链系统单位时间内的承压能力，可以为抵御分布式拒绝服务攻击等攻击提供依据。

5.1.2 数据处理响应时间测试

测试对象：区块链系统对交易请求作出响应所需要的时间，一般分为两部分：数据传输时间和系统处理时间。

测试方法：响应时间一般取数据处理的平均响应时间，基本思想是记录延时的起止时间，多次对相同数据的处理测试取平均值。可以通过模拟多用户并发请求，计算用户得到响应的时间。统计发送的总交易数 *T*，完成花费的所有交易的时间 *t*.那么区块链系统处理数据的平均响应时间为 $t_{avg}=t/T$.

测试目的：评测区块链系统在不同工作量条件下的性能行为，衡量系统的可用性。

5.2 算法安全性测试

5.2.1 密码算法安全性测试

测试对象：区块链应用在落地时，都使用了密码运算、密钥操作等密码服务。针对区块链系统中所涉及的密码算法，最基本要求就是保证算法是安全的。算法的安全主要体现在用已知的统计检测原理来检测随机算法输出序列是否随机。可以把密码算法看成是一个黑盒子，只关心密码算法输出序列的随机特性。

测试标准：（以最新版本为准）

GM/T 0028-2014 密码模块安全技术要求

GM/T 0019-2012 通用密码服务接口规范

GM/T 0018-2012 密码设备应用接口规范

GM/Z 4001-2014 密码术语

GM/T 0006-2012 密码应用标识规范

GM/T 0030-2014 服务器密码机技术规范

GM/T 0054-2018 信息系统密码应用基本要求

测试方法：

1) 伪随机数发生器测试：产生一定数量序列长度为 n 比特的检测样本，选择一定数量的检测项目中序列对每一个 n 比特序列串进行检测，并记录通过的百分比。通过率若小于某个临界值，则被测算法通过检测

2) 分组频数检测：检测密文分组的“0”，“1”平衡性，若输出的密文分组是随机

的，则应该有较好的“0”，“1”平衡性。

3) 明密文独立性检测：检测密文是否有不依赖与明文统计特性的性质。

5.2.2 哈希算法安全性测试

测试对象：除了加解密操作，区块链系统中最常见的密码学操作是哈希处理。多数哈希算法是通过多次迭代混乱和扩散两种基本技术的组合来达到安全性目的。混乱性和扩散性的检测是哈希算法检测的主要内容。

测试方法：

1) 局部碰撞测试： 假设 y_0，y_1，...，y_N 为数据中的哈希值 Y，对于任意固定的 s 个比特位，两两比较，记录相等的概率 p，计算 T 渐近服从标准正态分布，在给定的显著水平 α 下，若$|T|>\mu_{\alpha}$则认为发生局部碰撞，说明算法不安全。

2) 均匀性测试：测试 Y 的单比特的分布，统计整体的 0,1 频次，计算“1”率；测试 Y 的 m 个分位各自的 0,1 分布；测试 Y 的半字节分布，统计 Y 的半字节分布排列。

5.3 协议一致性测试

测试对象：协议一致性测试是协议测试中最重要的测试，目的是测试通信协议软件的实现是否与通信协议规范一致。区块链协议一致性指协议的实现与相应的区块链功能标准相一致，具体来说是区块链系统自身的行为和它与其他实际系统的通信行为都符合协议标准。

测试步骤：

1) 明确测试需求：根据区块链功能文档阅读需求，深入理解需求。

2) 制定测试计划：根据需求估算测试所需资源，安排功能点划分。

3) 编写测试用例：测试人员根据需求设计编写测试用例。

4) 执行测试用例：测试人员根据测试用例的详细步骤，执行测试用例，提交缺陷，跟踪缺陷至缺陷关闭。

5) 生成测试报告：对每个用例记录测试的结果，分类统计 bug 级别，评估区块链已实现的功能和未实现的功能，通过不断测试追踪，直到被测系统达到测试需求。

测试方法：协议一致性测试通过检查被测实现 IUT(Implementation Under Test)输入输出行为是否与相应的协议规范相一致，按照测试器是否向 IUT 发送数据可以把一致性测试分为两类：主动测试和被动测试。

5.3.1 主动测试

由测试器向 IUT 施加输入事件序列，接收 IUT 向测试器反馈的输出事件序列，并判断输出事件序列是否符合预期结果，然后检查状态转换后到达的新状态是否符合协议规范。

Upper Tester 是集成在被测件中的一个应用模块，与外部测试仪相互配合，执行对被测区块链系统的测试。整个测试环境包含三个成员，分别是上测试器（Upper Tester，简称 UT）、下测试器（Lower Tester，简称 LT）和测试系统（Test system，简称 TS）。其中 LT 和 TS 合并做为外部测试仪，而 UT 需要被集成到被测件（Device Under Test，简称 DUT）中，操作被测模块（Implementation Under Test，简称 IUT）来执行测试。

图 5.1 展示了具体的测试环境方案。测试过程中，由 TS 构建测试命令，然后 LS 将命令封装成 UDP 数据包，发送给 UT。UT 根据收到的命令，操作 IUT 执行对应的测试行为。DUT 发出的报文，由 LT 负责监听。下面的时序图简单展示了与数据发送相关的测试项的执行过程。TS 下达了发送数据“1，2，3”的测试命令，LT 将这个命令打包成一

条 UDP 数据包，发送给 UT，UT 根据数据包中的命令，操作 IUT 发送这些数据。LT 负责监听 DUT 实际发出的数据，LT 收到 DUT 发出的数据之后，将接收到的结果提交给 TS，TS 通过比对命令数据和收到的数据是否一致，来判定这个测试用例是否通过。

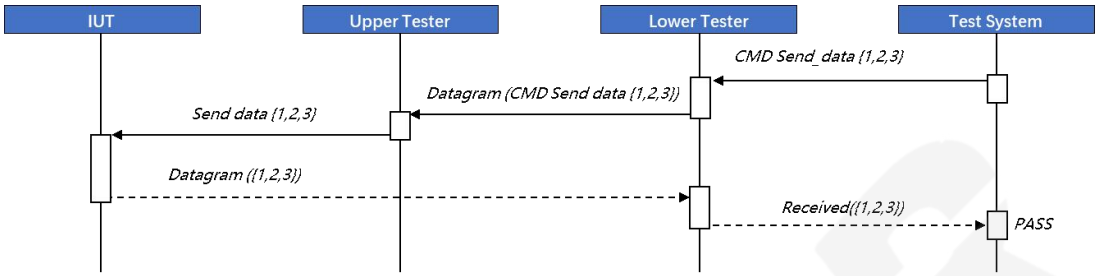

图 5.1 测试用例时序图

5.3.2 被动测试

被动测试中，测试器监控并采集处于运行状态的被测区块链系统的输入输出数据，然后通过消息比较来发现协议实现的行为是否有误，判断被测实现的状态转换是否正确。被动测试是一种监控诊断方法，由于被动测试不对 IUT 施加信息，只是被动地检测 IUT 与周围网络环境的交互，实现了在线测试。

5.4 协议安全性测试

测试对象：区块链安全漏洞指的是在区块链网络硬件、软件或协议中，其系统安全策略在设计或应用时潜藏的安全隐患，这些隐患可能导致网络系统在受到恶意攻击时发生安全问题，导致系统瘫痪或重要数据丢失。任何一种新兴技术的应用都可能产生网络安全漏洞，因此对区块链协议进行安全性测试是有必要的。

网络协议是指存在于区块链系统两个或多个通信实体间的网络行为规范，规定了两者或多者之间的相关责任和义务，对网络系统交互做出保障，一份网络协议的制定需要对协议规范、协议编址和寻址以及协议数据包进行重点把握。

协议安全性测试是指验证协议的安全等级和识别潜在安全性缺陷的过程。安全测试的主要目的是查找协议自身设计中存在的安全隐患，并检查协议对非法侵入的防范能力，根据安全指标不同测试策略也不同。

测试步骤：

1) 测试准备；

2) 测试执行；

3) 测试结果采集；

4) 测试结果分析；

5) 测试报告生成。

测试内容：

1) 区块链骨干协议安全性测试：针对区块链协议容错能力、异常处理能力、信息保障能力和通信活跃性进行测试。

2) SSL 协议安全性测试：针对区块链系统中包含 SSL 实现的模块的主要功能和性能进行测试，测试集应能覆盖 SSL 协议的认证、保密、完整性保护等功能，对 SSL 上的各种典型应用如安全超文本传输协议（SHTTP）等也应该覆盖，主要测试 SSL 协议的身份鉴别功能、数据加密功能、数据完整性保护功能。

3) IPSec 协议安全性测试：针对区块链系统中包含 IPSec 实现的模块的主要功能和性能进行测试，测试集应能覆盖 IPSec 协议的认证、保密、完整性保护等功能，对 IPSec 上的各种典型应用如 VPN 等也应该覆盖，主要测试 IPSec 协议的身份鉴别功能、数据加密功能、数据完整性保护功能。

4) Kerberos 协议安全性测试：针对区块链系统中包含 Kerberos 实现的模块的主要功能和性能进行测试，测试集应能覆盖 Kerberos 协议的身份认证等功能，主要测试 Kerberos 协议的票据分发功能、数据加密功能、票据验证功能。

5) TCP/IP 协议安全性测试：针对区块链系统中包含 TCP/IP 实现的模块的安全脆弱性进行测试，测试集应能覆盖 TCP/IP 协议簇中各协议的安全脆弱问题，主要测试 TCP/IP 协议的抗 IP 地址欺骗能力、抗 TCP 序列号攻击能力、抗极小数据段攻击能力和抗源路由攻击能力。

针对某些特定攻击的测试：例如对 DDOS 攻击、ARP 类攻击的测试。

测试方法：针对区块链系统实现的协议安全性测试分为以下三种测试方法。

5.4.1 随机扰动测试（fuzzing testing）

随机扰动测试方法主要思想是产生大量的随机输入参数传递给被测系统，并观察被测系统能否正确处理这些随机输入而不出现错误或可利用的漏洞。Fuzzing 测试对系统的相关协议进行无规律的尝试，以制造半有效数据，之后将这些数据输入到需要进行测试的协议中，根据该协议对其处理的过程，分析系统中是否存在安全漏洞。

一个典型的 Fuzzing 的工作流程如下：首先，确定需要进行测试的网络协议；其次，挑选进行测试的交互接口；第三，编写 Fuzzing 相关测试数据；第四，将得到的有效数据传输到交互接口中；第五，对测试协议的状态进行实时监测，收集有异常现象的信息；最后对安全漏洞进行评估，完成整个网络安全协议的安全性测试过程。

5.4.2 协议语法变异测试

语法变异测试是使用最广泛的协议安全测试方法，这类方法基于协议本身的语法特点，通过设计变异算子对相关语法元素进行变异而产生安全测试例。将变异分析用于寻找错误的一般做法是利用变异分析对区块链软件的外界接口参数进行变异，产生随机的或事先设计的恶意参数发送给被测区块链软件，观测它做出的反映。在区块链协议安全测试中，一般的方法是通过基于语法测试的变异，向协议提供异常的 PDUs（Protocol Data Unit，协议数据单元），并观察其反应和行为。

5.4.3 基于有限状态机模型的错误注入测试

这种方法提出“客户端/服务器行为模型化(CSB Modeling)”，即首先建立一个有限状态机来描述区块链中客户端和服务器之间的正确通讯，再使用图遍历算法来产生模板(test pattern，即一个转换序列，表示了某个正确的消息交换序列)。得到模板之后，再利用 SPIKE 技术模型化客户端和服务器之间交换的消息的语法，从而在错误注入时能灵活的产生测试数据。在客户端和服务端测试模式中，模糊器和被测对象分别为测试过程的两个端点。此时，模糊器可充当客户端的角色，用来测试服务端程序的安全性。同时，模糊器也可以充当服务端的角色，用来测试客户端程序的安全性。模糊器中的监控模块用来对被测对象的行为进行收集、分析以判断是否存在异常情况。

5.5 匿名性测试

测试对象：匿名性是互联网应用的重要情境特点，也是影响互联网上行为的重要因素。匿名性一般被定义为在一定的情境下，行为者对于自身是否被他人所知觉情况的感知，是行为者自己区分于他人的一种关系特征。区块链技术是典型的互联网应用，本节匿名性测试主要关注于区块链使用者，即对区块链用户的匿名保护。与前文对区块链隐私保护框架的描述类似，所考量的匿名属性包括数据内容匿名、用户身份匿名、和行为方式匿名。

测试内容：以下给出描述区块链系统不可观测性的定义。

1) 匿名性（anonymity）：指区块链通信主体在一组匿名集合中不可识别的状态。如果攻击者能够从获取到的信息中关联到匿名集合中的某一个发送者，如该发送者发送消息的 IP 地址，则认为该用户是可识别的。

2) 不可检测性（undetectability）：从攻击者的角度，攻击者不能够区分感兴趣的区块链通信客体是否存在。匿名性是研究通信主体的通信关系，保护的是通信主体的身

份信息，而不可检测性研究对象为感兴趣的通信客体，保护的是通信行为和通信客体。

3) 完美不可检测性（perfect undetectability）：当感兴趣的区块链通信客体存在与否是完全不可区分的，则认为该通信客体具有完美不可检测性.

4) 不可观测性(unobservability)：不可观测性是指感兴趣的区块链通信客体在任何其他相同类型的通信客体集合中不可区分的状态，不可观测性包括两个方面的含义：通信主体的匿名性和通信客体的不可检测性。

本节考虑对区块链系统匿名属性的度量和评估，测试内容是基 于某一特定消息出现的概率，即攻击者监视匿名通信系统一段时间 后，攻击者给每一个通信主体在通信过程中出现的消息的外显行为特征赋予一个概率值，用以表征该消息可能出现程度。

测试方法：目前对匿名通信系统的匿名性属性度量和评估缺乏统一的工具和测试方法，因此考虑从敌手的威胁模型出发，将区块链系统的输入输出状态映射到一个交互式图灵机，采用网络流量分析技术等手段研究协议在通信行为上是否具有统计意义上的不可区分性，即计算区块链协议通信过程中数据包的外显行为特征的相对熵值，用以量化评估协议的匿名性程度。分析手段主要包括输入输出关联分析、IP 地址关联分析和业务流分析。

5.5.1 输入输出关联分析

为了更具体的评估区块链系统匿名能力，从攻击方式、可见性、敌手拥有的计算资源三个方面定义敌手的攻击能力，每次测试采取控制变量法执行测试用例。

攻击方式

1) 被动攻击。敌手能够在区块链网络边界部署流分析系统，以监视、分析网络流以及通信主体的通信行为。

2) 介入攻击。敌手能够操纵区块链通信网络流，如加时延、丢包、修改包内容、

注入攻击等。

3) 主动攻击。敌手能够模拟区块链系统的客户端发包，以主动探测网络状态、节点行为特征等。

可见性

1) 局部可见。敌手仅仅能够监视区块链网络边界（节点加入和离开网络）或者部分网络节点。

2) 全局可见。敌手能够监视整个区块链通信网络节点以及流量。

资源

1) 有限。敌手具有有限的计算和存储资源。

2) 充分。敌手具有充分的计算和存储资源，支持从不同网络位置同时收集到大规模的网络流量进行分析处理。

在输入输出关联分析中，对于给定的区块链输入输出测试用例，测试模块将尝试恢复输入和输出的关联关系，构造输入输出交易关联拓扑图。其中恢复成功的概率与被测区块链数据内容匿名能力成反比。

5.5.2 IP 地址关联分析

在 IP 地址关联分析中，对于给定的区块链数据包测试用例，测试模块将在发送者匿名集中找出通信过程中该消息是否出现、来自于哪个发送者，在此发送者定义为所有可能的用户 IP 地址集合。其中 IP 地址关联成功的概率与被测区块链用户身份匿名能力成反比。

5.5.3 业务流分析

在业务流分析中，对于给定的区块链数据流量测试用例，测试模块将通过推断为数

据流添加业务标签，并将区块链数据流分类为若干类业务。其中业务流推断成功的概率与被测区块链行为方式匿名能力成反比。

6 区块链数据安全应用场景（举例）

6.1 变色龙哈希函数构建可编辑区块链

6.1.1 可编辑区块链的现实需求

可编辑区块链是区块链领域新兴的热点课题,致力于在保障区块链安全可信等良好性质的前提下实现链上数据的可控编辑操作。从区块链应用实践来看,目前区块链技术在信息监管、隐私保护、数据更新、可扩展性等四个方面都存在切实的数据编辑需求,迫切需要研究和应用可编辑区块链技术。首先，从信息内容安全的角度，由于一些区块链缺乏必要的上链信息审核与评估机制，为了防止不良信息带来持续的、永久性的负面影响，设计数据修改的渠道是有必要的；其次，从隐私保护的角度，用户个人具有要求责任方隐藏或者删除与自己有关的隐私数据记录的权利；最后，合适的可编辑手段可以用来修改区块链上由于主观故意或者客观疏忽而导致的错误数据，更新陈旧数据，精简历史数据，节省全节点存储空间，能够有效实现区块链数据更新提升区块链可扩展性。

由此可见,可编辑区块链具有明确而迫切的现实需求。同时，对区块链上数据的安全可控编辑操作也是实现区块链数据安全的重要范畴。

6.1.2 可编辑区块链的实现框架

可编辑区块链通常从编辑类型、编辑对象、编辑模态、编辑架构和控制策略五个方面来进行设计。如图 6.1 所示。

按照对区块链数据的操作类型，可以分为修改、删除、插入、过滤和隐藏共五类编辑操作，其中修改、删除、插入和隐藏是面向链上数据的操作，数据过滤是面向上链前数据的筛选和净化过程，最大限度阻止不良信息的写入。

按照按照数据编辑对象，可以分为区块级、交易级和数据项级编辑操作。区块级编辑技术粒度最大，只可以替换完整的区块而无法精准定位和修改区块中的特定数据；交易级和数据项级编辑技术粒度相对较小，前者重点针对区块中的交易数据(例如交易金额和接收方地址等)，而后者则侧重于文本数据(例如 OP_RETURN 类交易附言或其他文本数据)。通常来说，交易级编辑技术将会改变区块链内部的交易逻辑流和价值分配体系，是强上下文相关的编辑操作，因而难度更高。

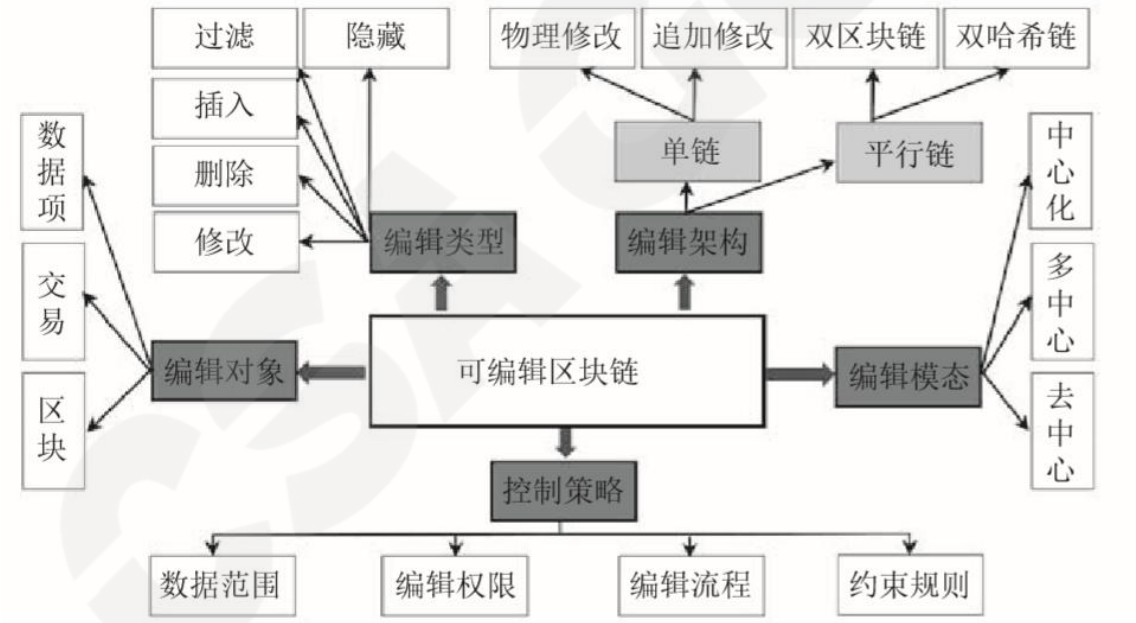

图6.1 可编辑区块链的实现框架

按照数据编辑模态，可以分为中心化、多中心化和去中心化的数据编辑，表示数据编辑权限(包括请求发起权、数据验证权和数据修改权)是否属于特定的中心化机构或者实体。可编辑区块链中，面向记账权竞争的共识过程与面向编辑权竞争的共识过程可能是同一阶段进行的，也可能是相对独立的，因此数据编辑模态结构并不一定与私有链、

联盟链和公有链一一对应。公有链中也可能发生由中心化实体主导的区块链数据修改操作，例如由比特币社区主导的硬分叉行为。

按照数据编辑架构,可以分为单链架构和平行链架构。前者仍然维护单一的线性区块链条，通过变色龙哈希函数等特定的技术手段实现区块数据的定点物理修改、或者在后续区块中追加修改后的新数据；后者则是维护独立运行的两条或多条平行链实现数据修改，例如两条平行的区块链或者两条平行的哈希链等。

按照数据编辑控制策略，需要详细规定涉及的数据范围(区块链数据的哪些数据内容、数据字段可以被编辑)、编辑权限(如何确定有请求编辑权限的实体、如何确定有验证新数据正确性权限的实体、以及如何确定有最终实施编辑操作权限的实体)、编辑流程(实施数据编辑操作有哪些具体步骤，操作执行的步骤是否需要序列化、原子化)、约束规则(区块链数据编辑过程中涉及哪些规则与约束条件)等要素。可编辑区块链编辑控制策略的设计与实施一般取决于实际场景的需求。

6.1.3 基于变色龙哈希函数的可编辑区块链方案

基于哈希运算生成的区块之间的哈希链路是区块链数据极难篡改的重要原因。常见的哈希函数具有抗碰撞性，即无法寻找到不同的数据替换使得区块间哈希关系仍然存在。同时，哈希函数具有高灵敏度，即使输入数据发生一个比特位的微小修改，输出哈希值也会发生明显的改变。因此，现有研究的基本思路是采用变色龙哈希函数，在不改变哈希函数输出结果的前提下实现区块数据的任意修改。

变色龙哈希函数是一种带陷门的单向哈希函数。如果掌握陷门信息，则可以轻易地计算任意输入数据的哈希碰撞，从而可以在不改变哈希函数输出的情况下，任意地改变哈希函数的输入。如果不掌握陷门信息，则变色龙哈希函数与传统的哈希函数一样具有抗碰撞性。因此，如果将区块链的哈希函数替换为变色龙哈希函数，并且人为地设置陷

门，则可以由陷门秘钥拥有者任意修改区块数据而不会破坏哈希链路的完整性。

变色龙哈希函数可以定义为：对于任意数据 x 和随机选择的参数 r，给定一个陷门 tk，可以找到消息对 (x,r) 和 (x',r') 使得 $CH(x,r)=CH(x',r')$，此处 CH 为变色龙哈希函数。在可编辑区块链应用中，x 和 x' 分别对应原区块数据和修改后的区块数据，且 $x \neq x'$。变色龙哈希函数通常有如下四个算法，即密钥生成算法 HG、哈希生成算法 CH、哈希验证算法 HV 以及哈希碰撞算法 HC，方案实现步骤如下：

1) 执行密钥生成算法 $HG(1^n)=(hk,tk)$：生成变色龙哈希的公钥 hk 和私钥（陷门）tk，其中 n 为安全性参数；

2) 执行哈希生成算法 $CH(hk,x;r)=(h,\varepsilon)$：给定公钥 hk、任意数据 x 和随机数 r，生成哈希值 h 和随机数 ε；

3) 执行哈希验证算法 $HV(hk,x,(h,\varepsilon))$：给定公钥 hk、任意数据 x、哈希值 h 和随机数 ε，如果 (h,ε) 是正确的哈希值，则输出 1，否则输出 0。

4) 执行哈希碰撞算法 $HC(tk,(h,x,\varepsilon),x')$：给定陷门 tk、三元组 (h,x,ε) 和修改后的新数据 x'，输出新随机数 ε'，使得验证时 $HV(hk,x,(h,\varepsilon))=HV(hk,x',(h,\varepsilon'))=1$，保持区块哈希链路的连续性。

显然, 掌握变色龙哈希陷门密钥就意味着拥有区块链的修改权，因此陷门密钥的管理对于变色龙哈希函数来说至关重要。在私有链结构中，陷门密钥一般由可信的中心化验证者掌握，可以实现任意链上数据的修改操作；而对于多中心和去中心化的联盟链和公有链来说，陷门密钥则必须在多个（固定的或者可变的）验证者之间共享，由验证者之间的共识过程决定是否参与和决策如何修改区块数据， 这种针对修改权的共识过程可以与验证区块数据的共识过程相互独立。从安全角度考虑，掌握陷门密钥和数据修改权的验证者必须是事前不可预测的，通过共识过程选举出拥有修改权的节点同样应该是不可预测的，以避免针对性的安全攻击。

6.1.4 可编辑区块链应用技术

1) 数据修改技术

2017 年，Ateniese 等学者最早提出了基于变色龙哈希函数的区块链数据修改方案，具有兼容性强、适合目前主流的区块链架构等优良特点，特别适用于带有少数可信验证者的授权区块链。该方案的思路是将区块链的内哈希函数 G 替换为变色龙哈希函数 CH，从而可以在不改变外哈希函数 H、不破坏哈希链路完整性的情况下，实现区块链上数据的物理修改。

对于中心化的私有链，存在唯一的中心化验证者集合 V 掌握变色龙哈希的陷门密钥 tk。对区块链上每个待修改的区块 $\boldsymbol{B_i}$，将变色龙哈希值(h_i,ε_i)增加到 $\boldsymbol{B_i}$ 中，验证者集合 V 将基于新数据 x_i' 计算内哈希函数的哈希碰撞，将旧数据 x_i 修改为新数据 x_i' 并生成新区块 $\boldsymbol{B_i'}$。所有区块修改完毕后，V 将新链 $\boldsymbol{C'}$ 广播至区块链系统中，其他节点必须同步到新链 $\boldsymbol{C'}$。

对于多中心或者去中心化区块链，可以通过秘密共享技术将陷门密钥安全地分发给预定义的验证者集合。该集合对于联盟链来说可以是固定的“联盟验证者”集合，对于公有链来说既可以是全部验证者(矿工)集合，也可以是按照特定规则选出的验证者集合(例如算力时间窗口内的矿工集合)。需要修改链上数据时，验证者将会启动安全多方计算协议，以分布式方式共同执行类似于私有链中的数据修改协议算法，从而实现链上数据修改。

2) 数据删除技术

链上数据删除技术主要包括本地数据删除和全局数据删除两种类型，前者是指分布式节点可以独立地删除其本地部分数据，以解决持续增长的区块链数据规模导致的存储瓶颈问题；而后者则是指分 布式节点通过共识算法来共同删除某些链上数据，主要致

力于解决清除链上不良信息的问题。

本地数据删除技术包括回收磁盘空间（RDS）、简化支付验证（SPV）和区块链剪枝技术。RDS 是指当区块链的最近一笔交易得到足够多的区块确认之后，在其之前的历史交易可以被删除以节省磁盘空间；SPV 技术则允许轻节点不必存储全部数据, 当其验证交易信息时, 可以通过区块链网络向其他全节点发起查询请求获得所需数据。区块链剪枝(Blockchain pruning)技术，致力于使验证者在其本地区块链账本中删除单个交易或数据项、或者是删除特定时间点之前的全部历史数据，例如选择性交易剪枝技术可以根据区块的状态可达性选择特定的不重要交易并删除。全局数据删除必须通过验证者的共识过程才能实现，实际上是将全局数据的删除操作视为一种修改操作，采用区块链链上数据修改技术来实现。

3) 数据插入技术

区块链数据插入技术的重点和难点是如何在去中心化的公有链中插入任意类型的数据。以比特币和以太坊等为代表的公有链主要存储以加密货币交易为核心的金融数据，其数据结构和语法相对固定、数据上下文相关性强、数据插入空间和规模有限，这些特点为插入任意类型的数据带来较高的难度。

现有的研究工作大多以比特币为原型来研究面向公有链的数据插入技术，其他加密货币的技术原理相近，只是在具体的数据插入位置和字段各有不同。以比特币系统为例，目前通常有四类数据插入手段，分别将任意数据插入到 Coinbase 交易、OP_RETURN 脚本、P2X 类型交易脚本以及非标准交易脚本中。

比特币 Coinbase 交易没有 ScriptSig 字段，可以由该区块的矿工在 Coinbase 交易中插入自定义的任意数据，最小 2 字节，最大 100 字节，该方法的局限性在于只能由矿工执行插入操作。OP_RETURN 是比特币交易的脚本操作码，是专门设计用来存储额外的

非金融数据的字段。OP_RETURN 脚本特别适合在比特币区块链上插入少量非金融数据，比特币用户和矿工均可以通过创建新交易实现方便快捷的数据插入操作。P2X (Pay to X)脚本是比特币系统的 P2PKH (Pay to public key Hash)、P2PK (Pay to public key)、MultiSig (多重签名)和 P2SH (Pay to script Hash)等标准交易脚本的统称，是将交易输出中锁定的比特币发送给由公钥、公钥哈希或者脚本等形式表示的接收者。P2X 方式是通过修改比特币交易输入(解锁脚本) 或输出(锁定脚本)中的公钥、公钥哈希、赎回脚本或其他脚本数据为自定义内容来插入任意数据。

4) 数据过滤技术

区块链数据过滤技术是指在数据实际写入到区块链之前，通过技术手段使得共识节点在共识过程中有效地检测和识别不良信息，或者通过经济手段提高不良信息上链的成本，以达到过滤和净化上链数据的目的。由于被过滤的数据并没有实际上链，因此不会涉及链上数据修改，也不会破坏区块之间哈希链路的完整性。

针对以金融交易数据为主的公有链，可以采取基于经济成本的数据过滤方法，通过调节交易费来限制带有大量数据的交易，从而提高任意数据上链的成本。针对联盟链和私有链，可以采取基于文本检测的数据过滤方法，在交易验证过程中展开语法检查，或者制定过滤策略执行语义检查，以阻止不良信息写入区块链。

目前，评价数据过滤技术的标准主要有四点，包括

a. 过滤质量：不良信息上链在计算开销和经济成本上都是难以承受的，可有效实现最大限度的数据过滤；

b. 可配置性：实施数据过滤技术不需要改变运行中的区块链系统，或者只有较低程度的协议改变；

c. 高可用性：数据过滤过程不会妨碍区块链的正常运行；

d. 低负载性：数据过滤不会使得区块链系统的性能发生严重下降。

5) 数据隐藏技术

为了保护用户隐私，在特定场景下链上数据也存在明显的数据隐藏的需求。例如不良信息上链后在一些特定场景下需要加以隐藏，使其不可读取，以避免产生负面影响。

第一种技术思路是以加密方式在链上存储数据的密文，并向授权用户或者 DAPP 开放解密密钥。这种方式相对灵活，但缺点是难以在多个实体之间有效地管理密钥，可能会出现由于密钥丢失或者泄露。

第二种技术思路是利用上述的链上数据修改技术，将需要隐藏的数据替换为适合公开的其他数据。然而，这种方式通常需要修改区块链协议来配置变色龙哈希、秘密共享等较为复杂的密码学工具, 因而在大规模区块链应用场景下可能存在性能瓶颈。

第三种技术思路是链下存储实际数据、链上仅存储时间戳和指向链下数据的哈希指针。当链下数据需要修改或者删除时，该数据在特定时间点存在的事实将会保留在区块链中，而且仅利用链上哈希数据 无法重构原始数据。这种方案的优势是较好地解决了区块链不可篡改性与数据隐藏需求的冲突，然而其缺陷是可能会降低区块链安全性并引入更多的受攻击面。

6.2 基于区块链+大数据风控的城市金融综合服务平台

6.2.1 城市金融综合服务平台

城市金融综合服务平台以“破解中小微企业融资难融资贵”为建设目标，坚持服务中小微企业的功能定位，实施“融资畅通工程”，推动提升金融服务实体经济质效。按照“统分结合、供需对接、信息共享、服务高效”的原则，围绕当前中小微企业融资对接过程中的难点、堵点、痛点，打造覆盖广泛、功能完备、产品丰富、过程监管的金融数据综

合服务产品，赋能中小微金融业务，最终建立可信、安全、实惠、高效的金融生态圈，实现可复制、可推广的中小微融资创新模式。城市金融综合服务平台的最大优势，是可以整合金融资源，打破信息孤岛壁垒，降低金融信息验证成本，同时提供统一的、可靠的、可追溯的数据采集、数据存储和数据处理方式。

6.2.2 城市金融综合服务平台功能架构

城市金融综合服务平台由以下五个层面的内容构成，如图 6.3 所示。其中，服务层面向用户提供服务，风控层面向各子平台和子系统提供一体化综合处理功能，可信层面向区块链事务，是数据安全可信的核心保障部分，数据层面向数据实体分为数据采集、数据存储和数据处理三个方面的内容，数据源是接口层，承接第三方征信、政府数据、企业数据等信息的输入。

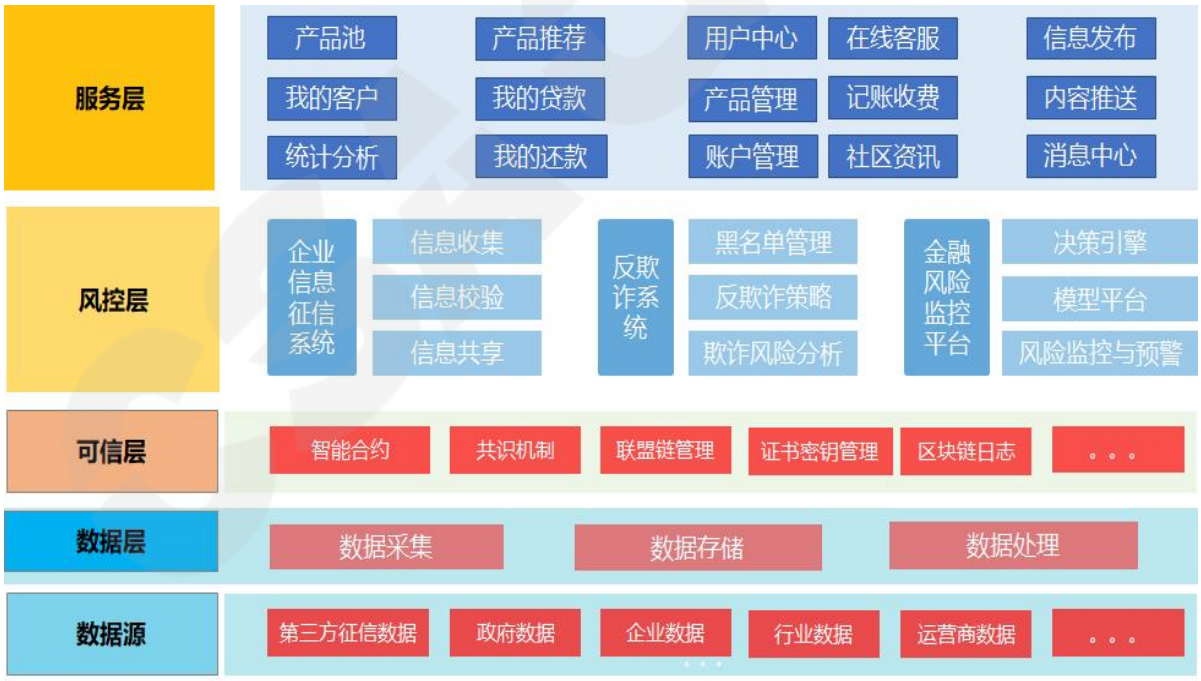

图6.3 城市金融综合服务平台功能架构

6.2.3 区块链数据安全保护技术

城市金融综合服务平台采用金融级别的安全标准。通过分级授权、确权机制、多重数据加密、数据脱敏、多重签名技术、防篡改技术，保证数据的完整性、保密性和可用性。主要通过以下五种方式结合区块链来保护数据的安全。

1) 分级授权原则

城市金融综合服务平台通过线上授权的方式授权信息上链，智能合约执行前将检查链上授权信息，得到授权则执行，没有授权则拒绝。合约根据用户角色进行分级授权机制，使得不同用户对链上数据访问的权限不同。授权流程如图 6.4 所示。

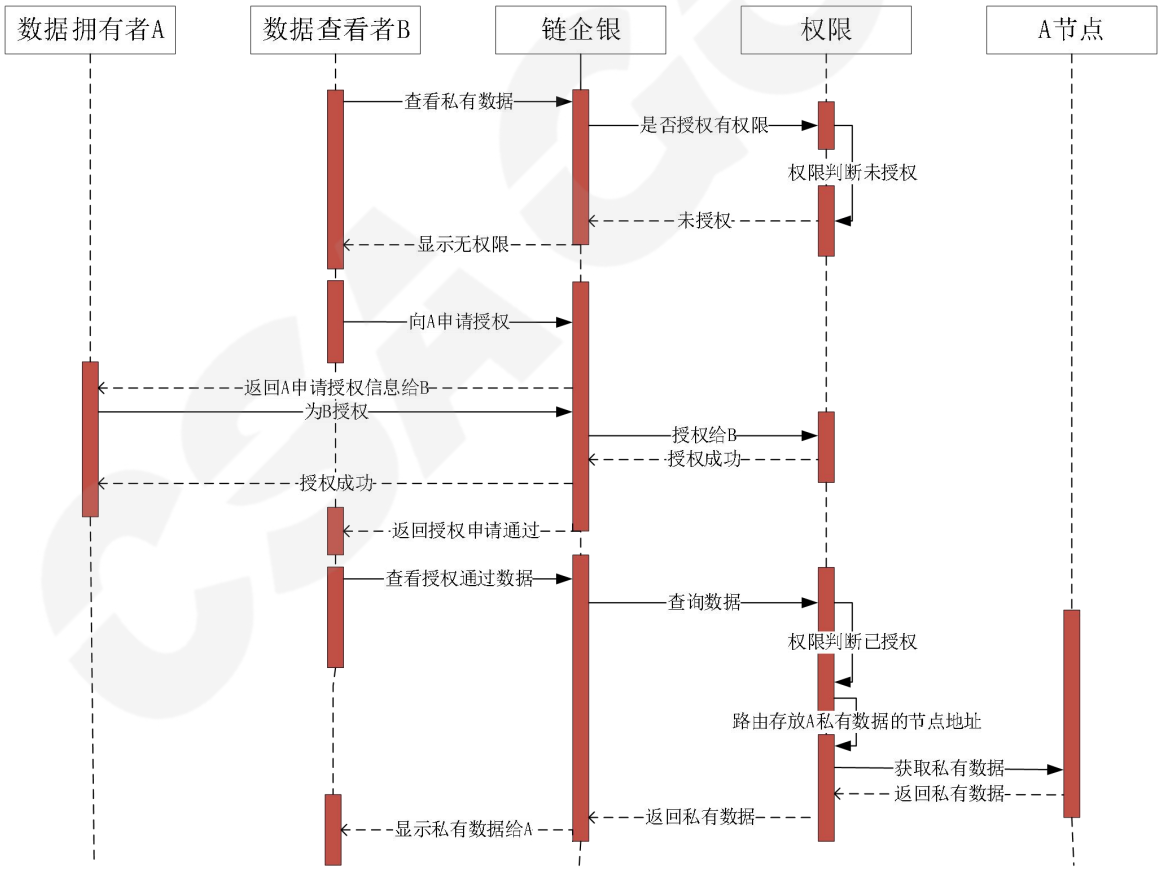

图 6.4 分级授权流程示意图

2) 数据加密技术

城市金融综合服务平台采取非对称加密算法（详细参见 4.3.1），采用一户一密，将密钥保存到密钥管理服务器 KMS。每个用户使用自身的密钥加密敏感数据，将加密后的密文保存到区块链上进行存储，对共享给具体某一方的数据，使用对方的公钥进行加密，保存到链上并共享给对方。数据加密技术流程图如图 6.5 所示。

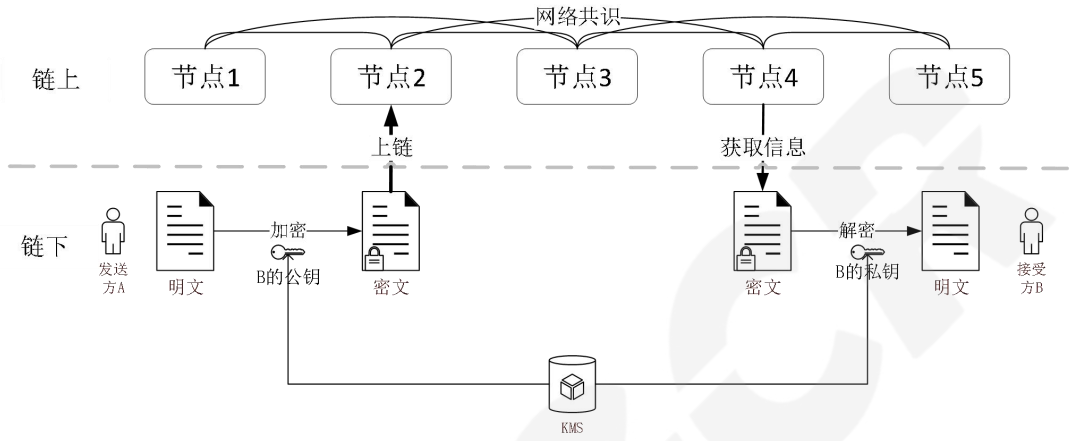

图 6.5 数据加密技术流程示意图

3) 多重签名技术

对关联多方的数据采用多重签名机制，多重签名机制是多方用户合作对同一个消息进行的签名，签名的长度与签名的人数无关。在这种数字签名中，每个用户用自己的私钥对消息签名，签名的验证者只需要用群体唯一的公钥就能验证签名的有效性。在城市金融综合服务平台中，采用有序多重签名机制（详情参见 4.3.3），对于某些重要的数据按规则只有得到多方用户的签名才认为是真实有效的。

4) 数据透明处理技术

以电子政务网的应用场景为例，风控引擎和链企银平台均部署在电子政务网，企业使用 APP/WEB 通过互联网访问部署在电子政务网的链企银平台，这样数据就始终在电子政务网内，满足数据可用不可见的要求。其中数据处理模块为每个部门提供单独接口，接收部门的原始数据，然后根据风控规则将原始数据进行处理，保存到 mysql 数据库，

将原始数据 hash 值上链存储，作为数据验证使用。风控引擎模块则负责接收链企银平台的调用请求，在 IDC 中进行计算，只将计算结果返回链企银平台。如图 6.6 所示。

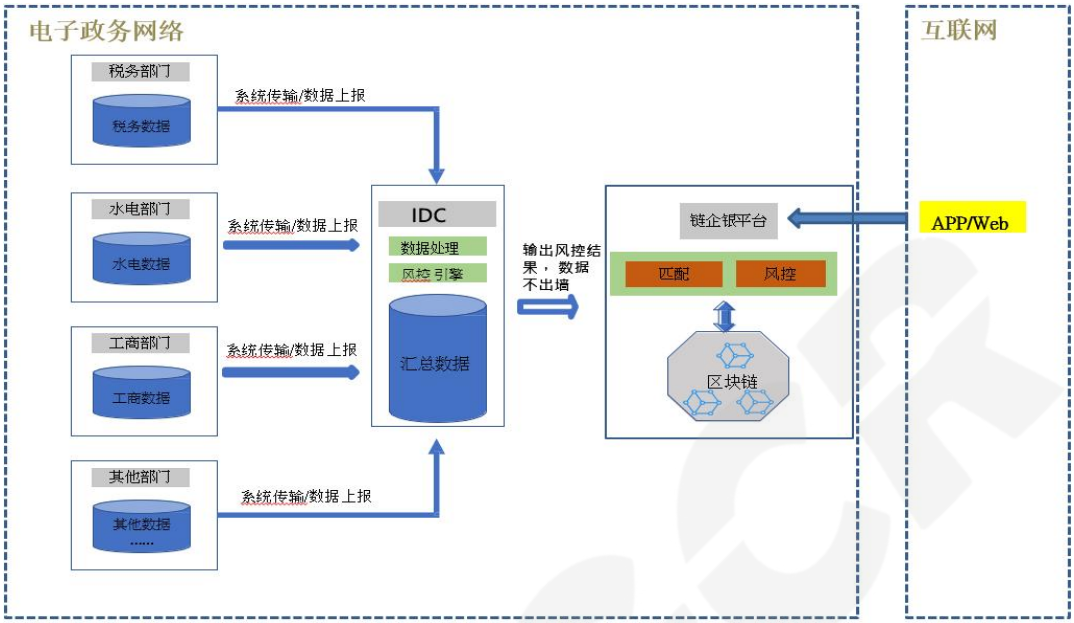

图6.6 数据透明处理

1) 数据验证和溯源技术

城市金融综合服务平台中所有的数据真实性、完整性都可以进行链上验证，同时也可以在需要时对数据进行溯源追踪，保证数据没有被篡改，数据验证和溯源流程如图 6.7 所示。

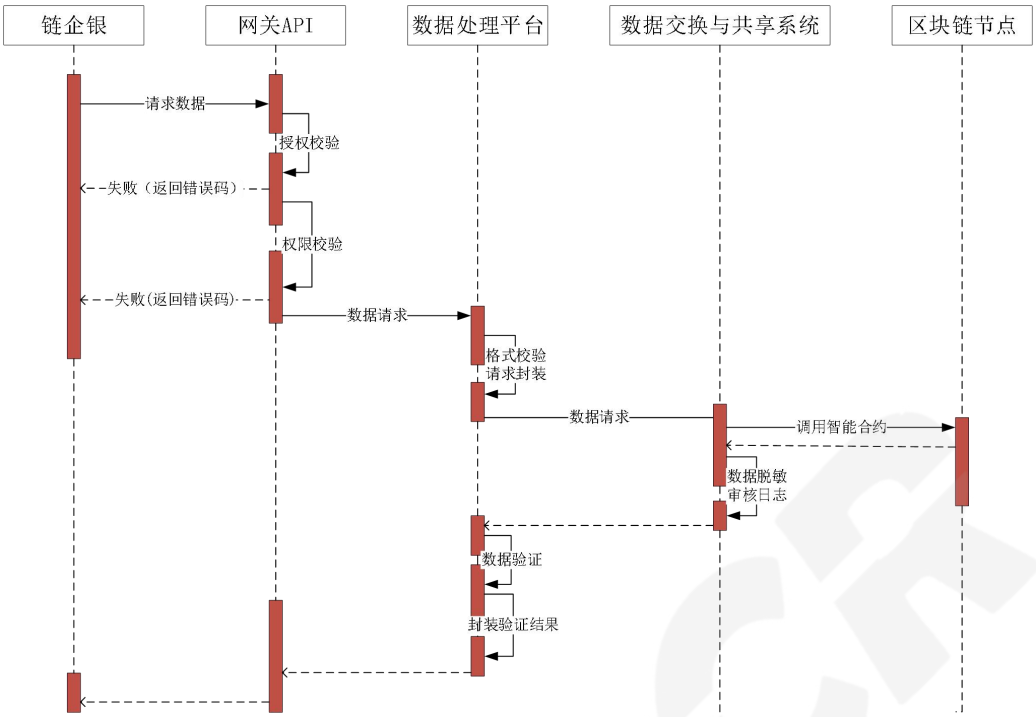

图 6.7 数据验证和数据追溯

6.3 基于区块链的司法存证方案

6.3.1 区块链存证技术背景

众所周知，区块链具有时间戳特性和不可篡改特性，这两个特性尤其适合用于数据存证。区块链存证的技术原理很简单，在用户签名和发送交易前，用户将要存证的数据（如果数据量小，而且不用担心隐私问题，可以直接存储正文(一般采取加密处理)，如果数据量大，则计算该数据的 Hash）附加到交易中，然后再进行签名广播。记账节点在验证了交易的合法性后，将该交易打包到区块中，并在区块中附加上时间戳信息。

使用区块链进行数据存证，需要满足以下几点基本要求：

1) 在存证所在区块的时间戳之前，该数据已经存在；

2) 拿到数据的内容，可以判断该数据在存证后是否更改；

3) 存证数据是由持有某私钥的人存证的，该人不可抵赖，别人也不可冒充。

虽然区块链存证具有以上的优势，但是比特币、以太坊等公有链毕竟不是为数据存证而设计的区块链，所以在存证上只有一个字段，对索引、扩展、引用功能都缺乏相应支持，需要第三方应用来实现。

6.3.2 e 签宝区块链司法存证方案

e 签宝是一家电子签名公司，提供全链路的电子签名服务，其中就包含存证服务，也就是客户签署完的合同备案。为了能保证证据链不被篡改，e 签宝使用了基于区块链的存证方法，主要是利用了区块链的不可篡改特性和时间戳特性。如果存证数据是保存在一个私有链、联盟链或者是 DPOS 公链上，那么从理论上来说，该区块链没有完全的去中心化，一旦区块链节点联合作恶，仍然存在历史数据被篡改的风险。

为了进一步加强存证所在链的安全性，e 签宝采用联盟链的方案，接入互联网法院，蚂蚁区块链等构建的包含“网通法链、可信电子证据平台、司法信用共治平台——一链两平台”在内的智慧信用生态体系。链上的节点全部都是法院、检察院、司法局、公证处、仲裁委等司法机构，由他们作为权威机构来证明本存证链没有被篡改，从而证明该链上的所有存证数据没有被联合作恶而篡改，e 签宝就是采用了此方式保证了本链数据的安全性。

图 6.8 是 e 签宝进行电子存证的示例。

图6.8　e签宝电子存证示例

6.3.3 司法存证中的数据安全保护

该架构的区块链数据安全主要包含：应用系统的安全、数据的网络传输安全、数据存储的安全、数据的访问留痕、数据的隐私保护等。

1)　应用系统安全

区块链司法存证应用系统都应该具有较强的安全能力，构建先进、主动的企业级安全威胁和漏洞检测体系，能够发现安全漏洞和安全事件，在系统遭到破坏后，可以快速恢复。

2)　网络传输安全

区块链跨机构网络传输过程应采用业内通用达到司法安全级别的加密方式，并且能验证数据来源，拥有抵御常规主动攻击或被动攻击的能力，保障数据传输过程中的真实性、完整性。

3)　数据存储安全

区块链司法存证应用系统中的机密数据应该加密存储，并且对密钥进行必要的保护；还需要保障用户数据隐私，只有被授权方才可以访问数据；通过联盟链保障数据不可以篡改，所有数据访问留痕。建立智能、全面的监控审计体系，全方位监控数据的存储和流动，保障数据安全。