基于区块链与边缘计算的物联网数据管理

程冠杰，黄诤杰，邓水光
（浙江大学计算机科学与技术学院，浙江 杭州 310007）

摘　要：智能设备的普及带动了物联网技术的应用和发展，而随之产生的海量物联网数据给传统集中式数据管理带来诸多挑战，如性能、隐私与安全的挑战。因此，提出了一种基于区块链与边缘计算的物联网数据管理架构来支持分布式的物联网数据管理，可以为物联网数据提供分布式存储和访问控制。同时设计了一种内置加密方案来保护数据的安全和隐私，并保障数据的所有权。通过引入边缘计算，解决了区块链系统的可扩展性瓶颈问题。给出了基于该架构的数据存储和数据访问流程，并详细说明了基于智能合约技术的系统实现算法。实验结果表明，与传统基于云的数据管理系统相比，基于该架构实现的物联网数据管理系统的性能更好。

关键词：物联网；区块链；边缘计算；数据管理
中图分类号：TP393.0
文献标识码：A
doi: 10.11959/j.issn.2096–3750.2020.00165

Data management based on blockchain and edge computing for Internet of things

CHENG Guanjie, HUANG Zhengjie, DENG Shuiguang
College of Computer Science and Technology, Zhejiang University, Hangzhou 310007, China

Abstract: The popularity of smart devices has driven the development of the application of Internet of things (IoT) technology, and the resulting massive amount of IoT data has brought challenges to traditional centralized data management methods, such as performance, privacy, and security. Therefore, a data management framework of IoT based on blockchain and edge computing was proposed to support the distributed IoT data management. The distributed storage and access control could be provided by the framework for the IoT data. At the same time, a set of the built-in encryption scheme was designed to protect the data security and privacy and safeguard the data ownership. By introducing edge computing, the scalability bottleneck of the blockchain the system was relieved. The processes of data storage and data access based on this framework were given, and the algorithm of the system implementation based on the smart contract technology was explained in detail. Experiments show that the IoT data management system based on this framework outperforms the traditional cloud-based data management systems.

Key words: Internet of things, blockchain, edge computing, data management

1　引言

物联网（IoT, Internet of things）是新一代信息技术的重要组成部分，也是信息化发展的重要标志[1]。通过连接智能设备，IoT 技术使得异构实体之间更容易收集、流通、处理和共享数据。目前，大多数 IoT 系统使用云服务提供存储和计算支持[2]，但是 IoT 海量、异构的数据特征以及多维、实时的服务

收稿日期：2020–03–25；**修回日期**：2020–04–26
通信作者：程冠杰，11821019@zju.edu.cn
基金项目：国家重点研发计划（No.2017YFB1400601）；浙江省重点研发计划（No.2015C01027，No.2017C01015）；国家自然科学基金资助项目（No.61772461）；浙江省自然科学基金资助项目（No.LR18F020003）
Foundation Items: The National Key Research and Development Program of China (No.2017YFB1400601), The Key Research and Development Project of Zhejiang Province (No.2015C01027, No.2017C01015), The National Science Foundation of China (No.61772461), The Natural Science Foundation of Zhejiang Province (No.LR18F020003)

请求等对基于云的数据管理架构产生了很大挑战[3]，极大地限制了 IoT 应用的发展。对基于云的数据管理架构产生的挑战具体包括如下 3 个方面。

1) 身份认证和访问控制

设备和用户的身份验证是 IoT 系统的第一层保障[4]。但是现有的身份验证机制一般完全依赖于第三方（如证书颁发机构），该机制缺乏可信性和稳定性[5]。此外，即使用户已通过认证，也无法排除其在进入系统后故意执行恶意行为的可能，如窃取数据和攻击网络。因此，需要制定高效的访问控制机制以实现对 IoT 用户的访问授权。

2) 数据安全与隐私

数据安全[6]和隐私保护[7]是 IoT 技术发展的核心问题。集中式的 IoT 数据管理架构容易受到各种网络攻击，如单点攻击和分布式拒绝服务攻击[8]，导致数据始终处于不安全的状态。因此，必须确保用户获取的 IoT 数据与数据源采集的数据完全一致，即未被篡改或丢失。另一方面，集中式的存储架构通常由中心化组织支配，使得生产者无法掌控数据的所有权。用户习惯用个人的数字资产来换取中心化组织提供的免费信息和服务，但是却忽略了由第三方掌控数据带来的安全和隐私威胁。以电商巨头阿里巴巴网络技术有限公司和北京京东世纪贸易有限公司为例，从“你可能还喜欢”的商品推荐到蚂蚁金服、京东白条的信贷评估模型，均利用了用户的购买记录数据进行筛选。为了更好地提取用户特征，数据管理者可以在用户不知情的情况下复制无数备份并将数据存储在组织内部的各个数据中心，若一个数据中心遭受攻击就会导致所有数据被泄露。因此，应该由数据生产者掌控数据的所有权，从而保障数据的安全和隐私。

3) 数据存储与处理性能

当前的 IoT 数据存储系统通常基于中心化架构构建，如由一个云服务提供商同时管理多个 IoT 应用[9]。IoT 从终端设备收集数据，然后发送到云服务器进行存储和进一步处理。但是这种基于云的中心化数据存储和处理模型无法适应 IoT 应用的扩展速度以及 IoT 场景的多样化需求[10]。面对大规模 IoT 应用，中心化架构将面临网络负载过重和网络传输时延不可预测等挑战，因此，海量异构 IoT 数据的处理性能无法得到保障。

综上所述，当前的 IoT 应用迫切需要一种新型的数据管理架构，从而能提供一种访问控制机制并确保 IoT 数据的安全和隐私，同时实现高效的数据存储和处理。学术界关于 IoT 数据管理问题的研究有很多，文献[11]针对 IoT 数据访问控制提出了一种基于身份的 IoT 认证机制，在该机制下，每个设备都有一个虚拟的网际协议第 6 版（IPv6, Internet protocol version 6）地址，在设备参与网络时作为身份证书。文献[12]设计了一种系统能够自动识别连接到 IoT 网络的设备类型，通过约束易受攻击设备之间的通信来尽量降低系统风险。文献[13]针对 IoT 数据隐私保护提出了一种基于生成源的数据分类机制，并创建了用户和访问控制列表，可有效防止 IoT 数据泄露。尽管上述工作已经为 IoT 数据管理提供了很多有效的解决方案，但基于云的本质并未改变，因此，仍面临中心化数据管理机制下的各种挑战和隐患。本文考虑使用区块链技术建立分布式的信任机制，为 IoT 数据管理提供了一种新的解决思路。

作为比特币的核心技术[14]，区块链的研究与应用呈现爆发式增长趋势。区块链本质上是一个分布式数据账本，区块链网络的每个参与者通过分布式共识算法维持存储数据的一致性，而不需要中心化机构的信用证书。区块链中记录的数据是不可篡改且可追溯的，只要系统中的诚实节点比所有攻击者节点拥有更多的算力，那么系统就是安全可靠的。此外，系统内置的数字签名技术和各种加密算法保障了链上数据的隐私和安全。因此，区块链技术为 IoT 数据管理提供了一种高效的解决方案。但是，区块链系统的整体性能受单个节点的性能上限的限制，导致存储和计算能力存在瓶颈，无法满足大规模 IoT 数据的可扩展性需求。边缘计算架构在移动网络边缘提供了充足的存储和计算能力，并且具备分布式、低时延和高带宽的特性[15]。通过引入边缘计算，可以很好地解决区块链的可扩展性瓶颈问题。此外，边缘计算可以满足 IoT 数据实时处理的需求，并为 IoT 数据提供了一个分布式存储架构。

因此，本文将区块链和边缘计算相结合，提出了一种新型 IoT 数据管理架构。基于内置的加密方案设计了一种主动访问控制机制，保护了数据的安全和隐私。基于边缘计算架构设计了一种新型分布式数据存储系统，详细说明了基于此架构的数据存储和数据访问流程，并对系统架构的性能进行了实验和分析。

2　相关工作

目前，已有一些研究工作将区块链与 IoT 进行结合。文献[16]提出了一种基于区块链的软件定义网络架构，实现了一种安全的分布式点对点（P2P, peer-to-peer）网络，其中，IoT 成员可以在没有可信第三方的情况下进行交互。文献[17]利用区块链为大型 IoT 系统提供访问控制，通过智能合约来注册、广播和撤销访问授权；IoT 设备拥有资源的所有权，而不是由中心机构进行监督。文献[18]提出了一种基于区块链的分布式访问控制方案，通过智能合约创建特定交易来定义访问控制策略。但是上述工作均基于私有区块链平台，通过牺牲去中心化来提升性能。此外，上述工作过分依赖于智能合约技术，而不是区块链架构本身。文献[4]提出了一种 IoT 身份验证机制，为每个用户分配一个唯一的 ID，然后将其记录到区块链上。此外，通过将重要数据进行哈希运算并存储到区块链中而提出一种数据保护机制，但是该机制忽略了区块链的存储瓶颈。

文献[19]和文献[20]所做的工作与本文的研究类似。文献[19]引入了联盟链来解决 IoT 数据的安全问题，然后采用了雾计算架构作为分布式环境部署区块链。但是，此方案的每个雾节点都需要存储完整的数据账本，导致存储负担很大。此外，所有 IoT 数据都存储在区块链上，导致系统缺乏可扩展性。在本文的数据管理架构中，区块链仅存储加密数据块的哈希值和一些重要文件，每台边缘服务器仅维护一部分状态信息，从而大幅度降低了存储开销并增强了系统的可扩展性。文献[20]提出了一种分布式的数据管理架构，将数据的哈希值存储在区块链中，并使用可信执行环境（TEE, trusted execution environment）将原始数据存储在可信硬件中。使用 TEE 会大幅度增加系统成本，并且数据的访问控制没有被纳入研究工作中。此外，大部分现有工作中的 IoT 用户都需要充当区块链节点才能参与网络，而同步区块将产生巨大的资源消耗。在本文提出的数据管理架构中，用户只需要与边缘服务器进行交互而不必加入区块链网络，因此，降低了用户使用系统的技术门槛和成本开销。

3　基于区块链与边缘计算的物联网数据管理

3.1　数据管理架构

基于区块链和边缘计算的 IoT 数据管理体系架构如图 1 所示，数据管理架构由 3 层构成，包括消费者层、由 IoT 设备和边缘计算网络组成的分布式存储层以及由边缘服务器维护的区块链层。1) 区块链层包含 IoT 数据的标识符和数据使用者的许可信息。区块链是整个数据管理方案的控制器，负责访问控制和数据验证，每个用户必须通过区块链上的访问控制才能继续访问 IoT 数据。在成功获取数据后，用户可通过区块链验证数据的一致性和完整性。2) 分布式存储层包含被大量 IoT 设备包围的边缘服务器。IoT 设备生成数据后，将数据经过加密和签名后存储在边缘服务器中。由数据所有者维护附录文件，为后续的数据请求和搜索提供相关信息。数据所有者会将数据生成的时间戳、加密数据块的数据摘要及其账户消息添加到附录文件中，并上传到区块链上，这种方式将 IoT 数据与其所有者相关联，保障了数据的所有权。3) 消费者层由想要访问 IoT 数据的实体组成。为了获得所需的数据，用户只需要与距离其最近的边缘服务器进行交互，而不需要考虑数据管理架构的具体实现细节，更不需要加入区块链网络。与用户交互的服务器将直接返回加密数据块与加密的对称密钥，用户通过解密即可获得原始数据。

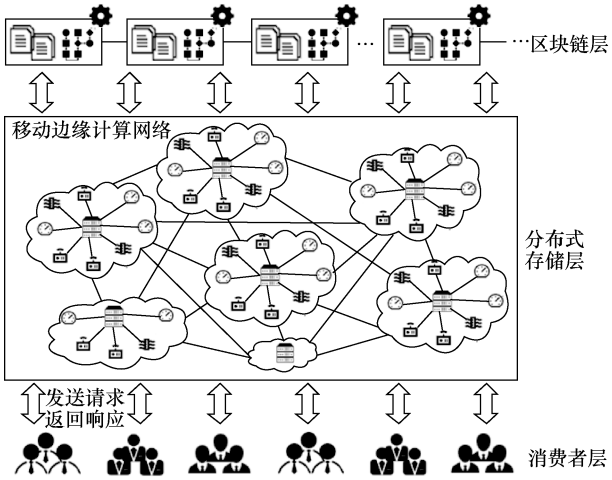

图 1　基于区块链和边缘计算的 IoT 数据管理体系架构

3.2　基于边缘计算的分布式存储

为了避免集中式存储系统带来的问题，本文利用边缘计算架构基于 Kademlia 算法[21]设计了一个分布式数据存储系统，由于 Kademlia 算法具有简单性、灵活性和安全性等特性，已成为 P2P 网络中数据存储和搜索的主流实现方式之一。为加入 Kademlia 网络的每台边缘服务器分配一个随机生

成的 160 bit 的节点身份（ID, identity）。将加密数据块的 160 bit 的哈希值作为编号，称为键，而加密数据块本身作为值，然后将数据块以键值对的形式存储在 ID 值与键相近的几台服务器上。Kademlia 网络中可容纳的最大节点数是 2^{160}，其存储容量远超过实际网络中所需设备的数量，因此，满足大规模 IoT 应用对可扩展性的需求。

分布式存储系统中的每台服务器只存储一部分加密数据，并不存储完整的数据账本。此外，服务器的状态信息通过 K-桶（K-bucket）[21]机制存储在各个节点中。Kademlia 算法通过异或操作来计算节点之间的距离。如 ID 为 01000000 的服务器与 ID 为 00000001 的服务器之间的距离为

$$01000000 \oplus 00000001 = 01000001 \tag{1}$$

将式(1)的结果转换成十进制值为 65。基于边缘计算的分布式存储结构如图 2 所示，每台边缘服务器都拥有 160 层的 K-桶机制表。对于 K-桶 i，服务器最多存储与其距离为$[2^{i-1}, 2^i)$的 k 个节点的状态消息。这些消息包括节点 ID、互联网协议（IP, Internet protocal）地址和访问端口。k 是系统级常数，可以根据存储系统动态设置，如比特流中使用的 Kademlia 算法将 k 设置为 8[22]。基于 K-桶机制的状态存储方式使得 n 台边缘服务器最多需要 lg$n$ 次查询就能找到目标信息。

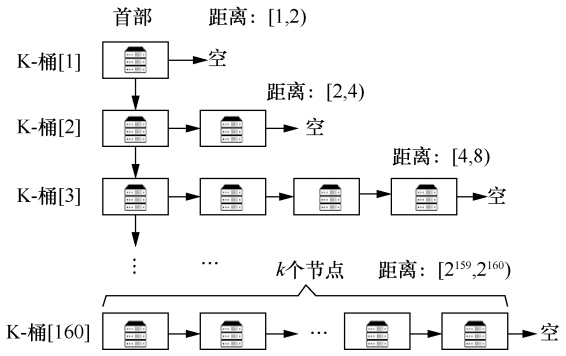

图 2　基于边缘计算的分布式存储结构

基于边缘计算的分布式存储架构有效地避免了传统分布式系统的两类常见问题。首先，分布式系统中节点的进/出非常频繁，当节点状态发生变化时，整个网络将广播地址更新并进行节点同步，从而导致网络拥塞并且大幅度降低了存储和搜索效率。在本文的存储方案中，每个节点仅维护部分服务器的消息，使得在任何节点发生状态改变时对整个网络的影响都降至最低。其次，在传统架构中，每个节点都维护整个网络的状态信息，一旦某个节点遭受攻击或故意作恶，则所有节点的状态信息都将被泄露。而本文利用 Kademlia 算法为存储系统提供了分区容错性，大幅度降低了信息被泄露的风险。

3.3　数据存储与请求的关键过程

3.3.1　*数据存储*

数据存储流程如图 3 所示。考虑安全性、效率和兼容性等因素[23]，本文选择高等级加密标准（AES, advanced encryption standard）作为对称加密算法。每个 AES 密钥（由 K_{AES} 表示）仅加密一个数据块，然后数据所有者立即将 AES 密钥进行本地存储。计算加密数据块的哈希值，然后将其附加到附录文件中并上传到区块链上，哈希值用来验证加密数据块的完整性。同时，数据所有者使用其私钥对加密数据块进行签名，从而提供数据身份验证，然后根据 Kademlia 算法将签名和加密数据块存储在边缘服务器中。数据所有者的公钥由 PK_{owner} 表示，私钥由 SK_{owner} 表示。本文假设边缘服务器是可信赖的，但是不排除服务器有宕机或其他硬件故障的风险。分布式存储算法如算法 1 所示。

算法 1　分布式存储算法

```
datablock ← (En[data], signature)
dataID = Hash(En[data])
for i = 1 → k do
    storageNode ← FINDNODE(dataID)
    if PING storageNode = True then
        STORE(dataID, datablock)
    end if
    dataID = dataID + 1
end for
```

首先，计算出加密数据块的哈希值并将其作为键。然后，将加密数据块和签名的副本以键值对的形式存储在 k 个节点 ID 值与哈希值较近的边缘服务器中。常数 k 由存储系统灵活设置，以适应 IoT 应用的动态需求。最后，数据所有者将时间戳、加密数据块的摘要及其账户消息添加到附录文件中，并将文件部署到区块链上以供查询。

3.3.2　*数据请求*

数据请求流程如图 4 所示，展示了从发送数据请求到最终获得原始数据的过程。

首先，消费者向距离其最近的边缘服务器发送数据请求。该请求包括数据所有者的部分账户信息、数据生成的时间戳和消费者的公钥（$PK_{consumer}$）。

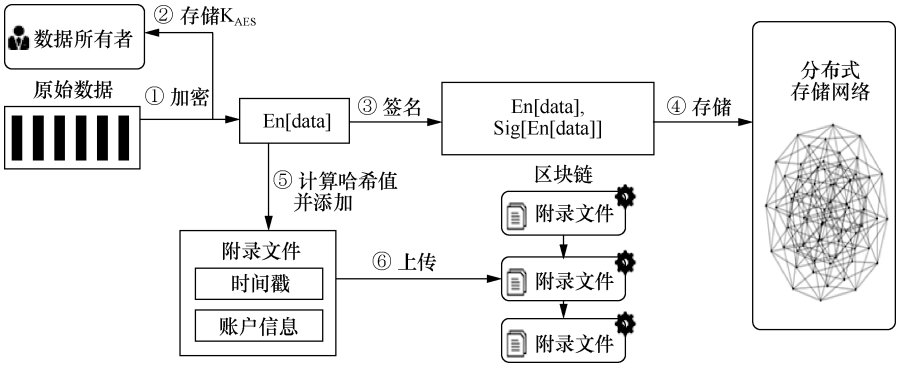

图 3　数据存储流程

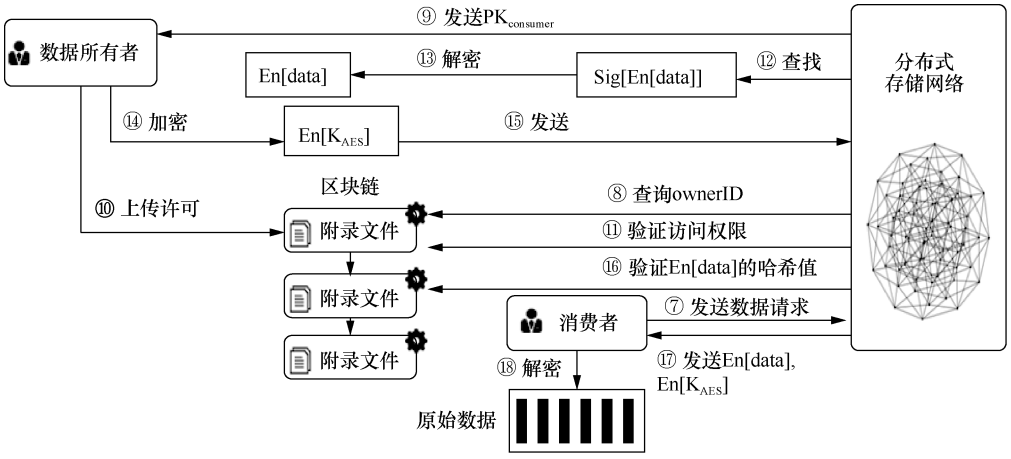

图 4　数据请求流程

边缘服务器通过时间戳和数据所有者信息在区块链上查询数据所有者 ID（ownerID），然后将 $PK_{consumer}$ 发送给数据所有者。如果数据所有者同意为用户授予访问权限，则上传许可文件到区块链上；如果拒绝用户访问，则上传拒绝文件。然后，边缘服务器通过区块链验证用户是否被成功授权。如果被拒绝访问，则边缘服务器将返回道歉声明。否则，服务器将在附录文件中找到加密数据块的摘要（datakey）。接下来，基于分布式搜索算法使用 datakey 来搜索键值对，分布式搜索算法如算法 2 所示。

算法 2　分布式搜索算法

key ← datakey
for $i = 1 \rightarrow k$ do
　　searchNode ← FINDVALUE(key)
　　if PING searchNode = True then
　　　　GET(dataID)
　　end if
　　close
end for
datablock ← Map(dataID)
　　find (En[data], signature)
　　verify the signature via PK_{owner}

可以通过 datakey 找到存储加密数据块和签名的特定服务器，然后服务器根据 PK_{owner} 验证数字签名，以确认数据块的所有权。数据所有者使用 $PK_{consumer}$ 对 K_{AES} 进行加密，然后将结果发送至与用户交互的边缘服务器。同时，服务器通过计算加密数据块的哈希值，将其与记录在区块链上的哈希标识符进行对比来验证数据的完整性。如果对比结果一致，则服务器将加密数据块和加密后的 K_{AES} 一起发送给用户。否则，服务器将向用户返回警告，指出数据已损坏。最后，用户解密数据块获取所需的 IoT 数据。

4　基于区块链与边缘计算的物联网数据管理原型实现

4.1　原型系统环境配置

区块链平台可分为公有链、联盟链和私有链。私有链通过牺牲去中心化来提升性能。与公有链相比，联盟链具有以下 3 个方面优势：1) 由于联盟节

点之间更容易达成共识，因此，联盟链中的交易速度很快。2) 联盟链的数据只能由授权实体访问，从而确保数据隐私。3) 联盟链是可控的。只要大多数联盟节点达成共识，就可以更改区块链中的规则，本文基于 Hyperledger Fabric 平台[24]实现数据管理系统。

数据管理架构中包含 3 种类型实体，即数据所有者、边缘服务器和 IoT 消费者。本文使用一台配备 Intel Core CPU i7-8700（3.2 GHz）的个人计算机（PC, personal computer）作为数据所有者，该计算机拥有 IoT 数据的所有权，承担收集数据并将其传输到边缘服务器的工作。为了运行 Fabric 区块链并执行数据存储，本文使用 3 台主机来模拟边缘服务器，分别命名为 Edge1、Edge2 和 Edge3。Edge1 和 Edge2 配备 Intel Xeon CPU E3-1220（3.00 GHz）和 32 GB 的随机存取存储器（RAM, random access memory），而 Edge3 配备 Intel Xeon CPU E5620（2.40 GHz）和 24 GB RAM；使用一台配备 Intel Core i9-9880H 和 16 GB RAM 的 MacBook Pro 作为 IoT 消费者。

4.2 基于智能合约的逻辑实现

本文使用智能合约技术来实现边缘服务器和区块链之间的交互逻辑。智能合约可以将实体之间的交互规则编码成程序，然后在对应条件触发时自动执行。如图 3 和图 4 所示，共有 5 个步骤需要与区块链系统进行交互。

1) 上传附录文件

基于智能合约的逻辑实现如算法 3 所示，数据所有者的账户名和数据块生成的时间戳被组合为 blockName，然后将 blockName 与加密数据块的哈希值、数据所有者的账户地址打包到附录文件中，最后将附录文件上传到区块链上。

2) 查询数据所有者 ID

IoT 消费者将数据请求发送到距离其最近的边缘服务器，然后服务器根据请求中与所需数据相关的信息在存储区块链上的附录文件中搜索出对应的数据所有者。

3) 上传许可文件

在此环节实现了 IoT 数据的访问控制机制，访问授权由数据所有者控制。对每个请求数据块都设置一个接受列表（acceptList）和拒绝列表（rejectList）。如果消费者被授予访问权限，则其账户地址将记录在 acceptList 中，以供后期验证；如果访问权限被拒绝，则消费者将被添加至 rejectList。这两个列表可以作为数据块访问记录，实现了访问的追溯和监管。

4) 验证访问权限

数据所有者做出对消费者的访问授权决策后，边缘服务器通过搜索消费者是否已被记录在对应数据块的 acceptList 来验证其访问权限并根据结果返回响应。

5) 验证哈希值

与区块链交互的最后一个操作是计算加密数据块的哈希值，并将计算结果与记录在区块链上的哈希标识符进行对比，以验证数据块的正确性和完整性。如果结果一致，则边缘服务器会将加密数据块和对称密钥一起发送给消费者；如果结果不一致，则退出访问。

算法 3 基于智能合约的逻辑实现

Step 1: 上传附录文件

blockName = (owner_account, data_timestamp)

blockHash = Hash(En[data])

owner_address = owner_account.address( )

blockAppendix←(blockName, blockHash, owner_address)

send blockAppendix to the blockchain

Step 2: 查询数据所有者 ID

send dataRequest to the closest edge server

integrate data_info of dataRequest into dataName

dataOwnerAddress = getDataOwnerAddress (dataName)

Step 3: 上传许可文件

dataOwner authorizes data access for the consumer

send blockPermission to blockchain

if blockPermission == Accept then

    acceptList = acceptList + consumerAddress

else

    rejectList = rejectList + consumerAddress

end if

Step 4: 验证访问权限

find requesterAddress from dataRequest

if requesterAddress in acceptList then

    continue

else

    break

end if

Step 5: 验证哈希值

Hash En[data] into HashInput

if blockAppendix.blockHash == HashInput then

send (En[data], En[K_{AES}]) to the consumer
else
return False
end if

5　实验与分析

5.1　实验结果与性能分析

本文从 3 个角度来评估数据管理架构的性能，即系统稳定性、用户体验质量（QoE, quality of experience）和系统计算开销。测试系统的网络时延和数据分组的超时率，并将其作为判断系统稳定性的指标；将发送数据请求到获得最终响应的总时间作为用户 QoE 的分析标准；由于加密和解密是系统中消耗计算资源的主要操作，因此，将方案中加密和解密操作的执行时间作为整个系统计算开销的量度。

在第一组实验中，将 Edge1 用作接受消费者数据请求的服务器，本文利用阿里云服务器作为基于云的数据管理策略来与本文的数据管理架构进行比较实验。阿里云服务器配有 Intel Skylake Xeon Platinum 8163（2.5 GHz）和 4 GB RAM（具有 64 bit Ubuntu 18.04 和 8 Mbit/s 带宽），将接收第一个网络响应的时间作为系统时延。本文执行 150 组实验，并计算得到两种数据管理方案下网络时延的平均值（Delay_avg）和标准差（Delay_sd）。此外，计算得到超时数据分组所占的比例，用来对比系统的稳定性。网络时延和超时率实验结果如表 1 所示，基于区块链与边缘计算的数据管理架构和云计算架构下的平均时延分别为 3.358 ms 和 5.719 ms，证明了前者具有更快的网络响应速度。此外，两种方案下的 Delay_sd 值相差较小，说明了本文所提方案的可行性和准确性。两种方案的数据分组超时率分别为 0.667%和 12.667%，表明本文的数据管理架构在系统稳定性方面显著优于集中式的数据管理方案。

第二组实验通过更改用户数据请求的大小，来观察基于云的方案和本文的数据管理架构的请求处理时间。处理时间包含从发送数据请求到获得最终结果的全部时间。两种方案的请求处理时间如图 5 所示（坐标值通过实际值取以 10 为底的对数取得），当数据请求大小设置为小于 10^4 B 时，两种方案的处理时间都非常短。但是随着数据请求大小的增加，处理时间会迅速拉大差距，可相差接近两个数量级。显然，本文的数据管理架构可以更快地处理用户的数据请求，因此，可以有效提高系统的 QoE，尤其适用于需要实时服务的 IoT 应用。

此外，本实验通过改变 IoT 数据块的大小计算出数据加密和解密的时间，并以此评估本文的数据管理系统中的计算开销。数据管理架构的计算开销如图 6 所示（坐标值通过实际值取以 10 为底的对数获得），实验结果表明，加密和解密算法对计算资源的消耗没有明显影响。此外，随着 IoT 数据块大小呈指数级增加，运行时间呈缓慢增长趋势，这表明本文方案的计算开销可控。另外，由图 6 可得，IoT 数据块大小最好不超过 10^5 B，从而减少系统的计算资源开销并保证计算效率。

表 1　网络时延和超时率实验结果

IoT 数据管理架构类型	Delay_avg/ms	Delay_sd/ms	超时率	最小值/ms	最大值/ms
基于区块链和边缘计算的架构	3.358	7.554	0.667%	2.263	95.313
基于云的架构	5.719	7.843	12.667%	4.560	95.042

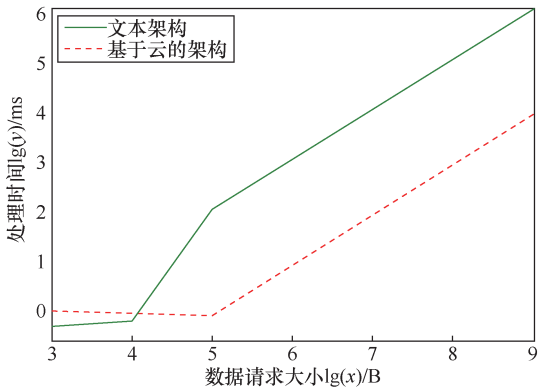

图 5　两种方案的请求处理时间

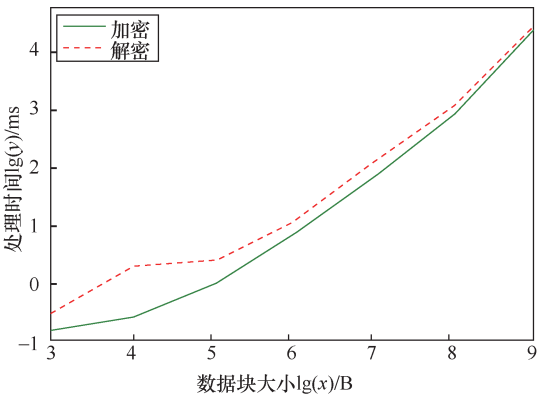

图 6　数据管理架构的计算开销

5.2　系统安全与隐私分析

5.2.1　安全性分析

本文使用信息安全三元组[25]来分析数据管理

系统的安全性，包含机密性、完整性和可用性。

1) 机密性

机密性规定了不同用户对 IoT 数据的访问控制权，只有授权用户可以获取数据并执行相关操作。具体来说，机密性要求系统设置对应的认证规则、访问控制和审查机制。本文的数据管理架构对于终端用户呈现出一个轻量级的接口，用户不需要加入区块链，通过直接与边缘服务器进行交互即可访问数据，消除了恶意用户从外部攻击管理系统的风险。此外，基于区块链的分布式共识和加密机制保证了系统内部的安全，因此，不需要制定冗杂的用户认证规则。在本文提出的数据管理架构中，访问控制权限完全由数据所有者决策。访问记录以许可文件的形式存储在区块链上，因此，本文的数据管理架构还提供了追溯和审计功能。

2) 完整性

完整性是指系统中的 IoT 数据无法被未经授权的用户非法使用和篡改，用户检索到的数据应该来自真实的数据源并且与原始数据一致。本文的数据管理架构通过使用哈希函数和数字签名来实现以下目标：原始数据经 AES 加密后由数据所有者使用私钥进行签名，实现了数据来源可验证，保证了数据的真实性；系统将加密数据块的哈希值存储在区块链中，获取数据块后计算得到哈希值并进行对比，基于哈希算法的单向性和抗碰撞性来验证数据的完整性。此外，在数据的存储和访问过程中，数据块均以加密状态进行传输，数据所有者以外的实体均无法获取真实数据，防止了数据被非法使用和篡改，进一步提高了数据的完整性。

3) 可用性

可用性是指授权用户可以随时访问和使用数据，具体包含两方面要求：首先，在系统遭受攻击时，要求能够继续提供可靠的服务；其次，用户的访问请求应该在有限时间内得到响应。本文的数据存储方案基于边缘计算架构和 Kademlia 算法，前者通过分布式的边缘服务器提供数据存储资源和计算能力，后者提供了数据存储的分布式拓扑结构，将边缘服务器通过 K-桶机制互相关联，两者为系统提供了分区容错性，即如果一台服务器遭受攻击或者发生故障，用户仍可以通过其他边缘服务器获取所需服务，不会影响系统的整体功能。本文系统以轻量级接口的形式对外提供服务，基于边缘计算架构搭建在网络边缘，因此，即使终端用户的需求激增，系统仍可以提供稳定、低时延的服务，及时响应用户的访问请求。因此，系统的可用性得到保障。

5.2.2 隐私保护分析

区块链系统中的隐私问题主要可以分为账户隐私和交易隐私[26]，分别对应于用户身份隐私和数据隐私。在本文的数据管理架构中，用户的身份信息、物理地址和 IP 地址不与系统中的公开信息（如用户的公钥和账户地址）相关联，任何节点都无法根据系统中的公开信息推测出用户的真实身份信息。系统运行基于智能合约的自动触发机制，降低了由人为干涉带来的隐私泄露风险。因此，用户身份隐私得到了很好的保护。数据隐私要求任何未经授权的节点都无法通过有效的技术手段获得关于 IoT 数据的任何信息，IoT 数据从数据源经 AES 加密，直到授权用户获取加密数据块和加密的 AES 密钥才能得到原始数据，数据在系统中均以加密状态存在，即使恶意攻击者获取加密数据块，也无法解密得到原始数据。本文基于区块链设计了一套主动访问控制机制，申请数据的用户得到数据所有者的访问授权才能读取加密数据块。访问权限将记录在区块链上以供查询，数据访问记录与访问用户形成映射，进一步保护了数据隐私。

6 结束语

本文提出了一种基于区块链与边缘计算的 IoT 数据管理方案。基于 Kademlia 算法设计了一种分布式数据存储方案，提高了系统的存储效率。此外，提出了一种基于区块链的主动访问控制机制，只有数据所有者授权的用户才能访问 IoT 数据。设计了一种内置的加密方案来保障数据安全和隐私，另外，基于智能合约在 Hyperledger Fabric 平台实现了系统原型。实验结果表明，与基于云的数据管理策略相比，本文提出的数据管理架构性能更高效。

参考文献：

[1] 邬贺铨. 物联网技术与应用的新进展[J]. 物联网学报, 2017, 1(1): 1-6.
WU H Q. Technology and application progress on Internet of things[J]. Chinese Journal on Internet of Things, 2017, 1(1): 1-6.

[2] 孙玉. 我国物联网产业发展趋势[J]. 物联网学报, 2017, 1(3): 1-5.
SUN Y. Development trend of IoT industry in China[J]. Chinese Journal on Internet of Things, 2017, 1(3): 1-5.

[3] ZHOU J, CAO Z F, DONG X L, et al. Security and privacy for cloud-based IoT: challenges[J]. IEEE Communications Magazine, 2017, 55(1): 26-33.

[4] LI D X, PENG W, DENG W P, et al. A blockchain-based authentica-

tion and security mechanism for IoT[C]//2018 27th International Conference on Computer Communication and Networks (ICCCN). IEEE, 2018: 1-6.

[5] HAMMI M T, HAMMI B, BELLOT P, et al. Bubbles of trust: a decentralized blockchain-based authentication system for IoT[J]. Computers & Security, 2018, 78: 126-142.

[6] ZHANG D. Big data security and privacy protection[C]//8th International Conference on Management and Computer Science (ICMCS 2018). Atlantis Press, 2018.

[7] ZYSKIND G, NATHAN O. Decentralizing privacy: using blockchain to protect personal data[C]//2015 IEEE Security and Privacy Workshops. IEEE, 2015: 180-184.

[8] LIU F F, GE Q, YAROM Y, et al. Catalyst: defeating last-level cache side channel attacks in cloud computing[C]//2016 IEEE International Symposium on High Performance Computer Architecture (HPCA). IEEE, 2016: 406-418.

[9] JOSEP A D, KATZ R A D, KONWINSKI A D, et al. A view of cloud computing[J]. Communications of the ACM, 2010, 53(4): 50-58.

[10] ZHANG B, MOR N, KOLB J, et al. The cloud is not enough: saving IoT from the cloud[C]//7th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 15). 2015.

[11] SALMAN O, ABDALLAH S, ELHAJJ I H, et al. Identity-based authentication scheme for the Internet of things[C]//2016 IEEE Symposium on Computers and Communication (ISCC). IEEE, 2016: 1109-1111.

[12] MIETTINEN M, MARCHAL S, HAFEEZ I, et al. IoT sentinel: automated device-type identification for security enforcement in IoT[C]//2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2017: 2177-2184.

[13] KALIYA N, HUSSAIN M. Framework for privacy preservation in IoT through classification and access control mechanisms[C]//2017 2nd International Conference for Convergence in Technology (I2CT). IEEE, 2017: 430-434.

[14] CROSBY M, NACHIAPPAN, PATTANAYAK P, et al. Blockchain technology: beyond bitcoin[J]. Applied Innovation Review, 2016, 2: 6-19.

[15] MAO Y Y, YOU C S, ZHANG J, et al. A survey on mobile edge computing: the communication perspective[J]. IEEE Communications Surveys & Tutorials, 2017, 19(4): 2322-2358.

[16] SHARMA P K, SINGH S, JEONG Y S, et al. DistBlockNet: a distributed blockchains-based secure SDN architecture for IoT networks[J]. IEEE Communications Magazine, 2017, 55(9): 78-85.

[17] XU R H, CHEN Y, BLASCH E, et al. BlendCAC: a blockchain-enabled decentralized capability-based access control for IoTs[C]//2018 IEEE International Conference on Blockchain. IEEE, 2018: 1027-1034.

[18] NOVO O. Blockchain meets IoT: an architecture for scalable access management in IoT[J]. IEEE Internet of Things Journal, 2018, 5(2): 1184-1195.

[19] FARHADI M, MIORANDI D, PIERRE G. Blockchain enabled fog structure to provide data security in IoT applications[J]. arXiv: 1901.04830, 2019.

[20] AYOADE G, KARANDE V, KHAN L, et al. Decentralized IoT data management using blockchain and trusted execution environment[C]//2018 IEEE International Conference on Information Reuse and Integration (IRI). IEEE, 2018: 15-22.

[21] MAYMOUNKOV P, MAZIERES D. Kademlia: a peer-to-peer information system based on the XOR metric[C]//International Workshop on Peer-to-Peer Systems. Springer, 2002: 53-65.

[22] COHEN B. Incentives build robustness in BitTorrent[C]//Workshop on Economics of Peer-to-Peer Systems. 2003, 6: 68-72.

[23] NECHVATAL J, BARKER E, BASSHAM L, et al. Report on the development of the advanced encryption standard (AES)[J]. Journal of Research of the National Institute of Standards and Technology, 2001, 106(3): 511.

[24] ANDROULAKI E, BARGER A, BORTNIKOV V, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains[C]//Proceedings of the 13th EuroSys Conference. 2018: 1-15.

[25] SAMONAS S, COSS D. The CIA strikes back: redefining confidentiality, integrity and availability in security[J]. Journal of Information System Security, 2014, 10(3): 21.

[26] 王宗慧, 张胜利, 金石, 等. 区块链数据隐私保护研究[J]. 物联网学报, 2018, 2(3): 71-81.
WANG Z H, ZHANG S L, JIN S, et al. Survey on privacy preserving techniques for blockchain[J]. Chinese Journal on Internet of Things, 2018, 2(3): 71-81.

[作者简介]

程冠杰（1996– ），男，江苏泰兴人，浙江大学计算机科学与技术学院博士生，主要研究方向为区块链、物联网和边缘计算。

黄诤杰（1996– ），男，浙江宁波人，浙江大学计算机科学与技术学院硕士生，主要研究方向为信息安全和隐私、区块链等。

邓水光（1979– ），男，湖南衡阳人，博士，浙江大学计算机科学与技术学院教授，主要研究方向为边缘计算、服务计算、移动计算和区块链。