计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2019,28(11):79−86 [doi: 10.15888/j.cnki.csa.007144]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于 CP-ABE 和区块链的数据安全共享方法①

黄　穗, 陈丽炜, 范冰冰

(华南师范大学 计算机学院, 广州 510631)
通讯作者: 范冰冰, E-mail: 2017022345@m.scnu.edu.cn

摘　要: 数据共享是打破大数据时代“数据孤岛”困境的有力途径, 而如何保证数据安全共享是当前面临的主要问题. 为此, 本文基于区块链技术和密文—策略基于属性的加密 (Ciphertext-Policy Attribute-Based Encryption, CP-ABE) 提出 DOB 框架, 使用智能合约和序列化方法将 CP-ABE 的系统公钥、用户属性、密文和用户密钥等存储在链数据库中, 同时设置数据库的访问权限和注册认证数据集, 实现数据的细粒度共享. 实验结果表明: 相比于 Jemel 等人提出的 Timely CPABE with Blockchain 方案, DOB 框架能进一步提高数据共享的安全性.

关键词: 大数据; 数据共享; 区块链; 密文-策略基于属性的加密; 智能合约

引用格式: 黄穗,陈丽炜,范冰冰.基于 CP-ABE 和区块链的数据安全共享方法.计算机系统应用,2019,28(11):79–86. http://www.c-s-a.org.cn/1003-3254/7144.html

Data Security Sharing Method Based on CP-ABE and Blockchain

HUANG Sui, CHEN Li-Wei, FAN Bing-Bing

(School of Computer Science, South China Normal University, Guangzhou 510631, China)

Abstract: Data sharing is a powerful way to break the “Data Isolation” dilemma in the era of Big Data and how to ensure the safe sharing of data is the main problem that we face at present. Therefore, in this study, we propose a DOB framework by using Blockchain technology and ciphertext-policy attribute-based encryption. The system public key, user’s attribute, ciphertext, and user secret key of the CP-ABE are stored in the Blockchain database through the smart contract and serialization method, and the access authority of the database and the registration authentication dataset are deployed to implement fine-grained sharing of data. The experimental results show that compared with the Timely CP-ABE with Blockchain scheme by Jemel et al, the proposed framework further improves the security of data sharing.

Key words: big data; data sharing; blockchain; CP-ABE; smart contract

1　引言

随着社会信息化的不断提高, 各类信息系统积聚着大量数据资源, 这些数据普遍存在难以融合关联的问题, 将会使产业间陷入“数据孤岛”式困境. 数据共享有效促进各类数据交叉引用, 提升数据的潜在价值, 带来巨大的社会和经济效益[1−4]. 然而, 越来越多的数据涉及用户个人隐私或相关隐私, 如医疗档案中的临床诊断结果、出生日期、医疗卡号等, 如果共享不当, 将不可避免产生隐私泄露问题[5,6]. 对于这部分数据, 一般采取细粒度的访问控制和数据加密技术[7], 在有限的范围和时间段内对特定的人或组织开放, 按照“特定用户—特定场景—特定资源”的方式进行细粒度共享, 同时对数据共享过程进行追踪. 区块链[8]具有不可篡改、可追溯和可编程的特性, 基于区块链构建安全共享框架可用于满足数据限制开放和透明监管的安全需求.

Xu 等人[9]根据隐私数据共享的应用需求, 提出一

① 基金项目: 广东省重大科技专项 (2016B030305003)
Foundation item: Science and Technology Major Program of Guangdong Province (2016B030305003)
收稿时间: 2019-04-08; 修改时间: 2019-05-08; 采用时间: 2019-05-16; csa 在线出版时间: 2019-11-06

种将区块链作为软件连接件的系统架构. 在区块链上实现访问控制、交易验证、数据注册和密钥分发等业务逻辑, 将数据存储、密钥生成、数据加密等操作放在区块链下进行, 链上与链下之间通过交易和智能合约进行数据交互. 但没有详细阐述具体的技术细节和算法思想. Di Francesco Maesa D 等人[10]利用比特币区块链实现一种基于属性的访问控制方案. 该方案通过策略创建交易 PCT (Policy Creation Transaction) 和权限转移交易 (Right Transfer Transaction) 实现策略的创建、更新、撤销和用户间访问权限转移, 区块链作为分布式数据库存储 XACML 策略和操作日志. 资源的访问权限可以由最后一个权限所有者在区块链上发起交易转移到合法请求者, 无需资源所有者处理. 此外, 任何用户都可以对交易记录进行审计, 实现对访问策略全周期透明管理. 该方案的区块链用于维护分布式节点数据的一致性算法是基于算力的, 因而系统运作存在较大的计算开销. 文献[11]提出一种结合分布式文件系统 IPFS、以太坊区块链和基于属性加密的访问控制框架. 在此框架下, 数据拥有者能够在区块链上通过交易的方式将密钥分发给数据请求者, 同时指定访问策略来加密共享数据. 此外, 作者通过智能合约实现了密文的关键字搜索功能, 解决了传统云服务器无法返回搜索结果和容易返回错误结果的问题. Jemel 等人[12]在区块链和密文-策略基于属性的加密 (CP-ABE, Ciphetext-Policy Attribute-Based Encryption) 基础上, 提出一种名为 Timely CP-ABE with Blockchian 的访问控制方案, 通过广播交易将带有时间属性的访问策略发布到区块链, 只有在特定时间内用户属性满足访问策略的请求者才能获取解密密钥. 该方案适合处理和解决开放共享环境下数据保护所面临的细粒度问题, 既有效减少由密钥撤回所带来的开销, 又可追踪用户发布和获取数据访问权限的操作记录. 然而存在以下不足: (1) 区块存储容量有限, 难以通过交易的形式实时部署众多的访问策略. (2) 数据请求者的属性集合被封装在交易信息, 并被广播到全网所有节点公开可见, 极易被恶意节点盗用生成正确的用户密钥.

本文通过借鉴上述方案的优点, 提出了一种基于 CP-ABE 和区块链的数据安全共享框架, 主要在以下方面进行改进:

(1) 摒弃文献[12]基于交易进行权限管理的方式, 以采用智能合约部署访问控制代替.

(2) 摒弃文献[12]将访问策略和用户属性公开存放在区块的方法, 以将其存储在区块链数据库上, 并设置访问限制代替.

(3) 将数据集元数据发布到区块链上, 对数据集进行注册认证, 为数据的确权提供保障的同时, 供数据请求者发现数据.

2 框架相关技术

2.1 区块链

区块链是一种以数据区块为基本单位的按时间顺序组合形成的链式数据结构, 并以密码学方式保证的不可篡改和不可伪造的分布式账本. 在区块链 1.0 时代, 区块链利用共识机制和 P2P 网络技术实现区块数据传输、验证以及冗余备份, 利用加密的带时间戳的链式区块结构存储数据, 通过后序区块对前序区块进行验证, 并从时间维度进行关联, 保证区块数据不可篡改且可追溯, 仅仅是一种被用来保证去中心化架构下数字货币交易可信性的技术方案的一部分[13]. 进入 2.0 时代, 区块链融合以智能合约为代表的链上脚本技术, 实现各种顶层复杂应用场景的业务逻辑, 为区块链可编程特性提供了基础, 极大拓宽了区块链的应用领域[14,15]. 区块链结构如图 1 所示.

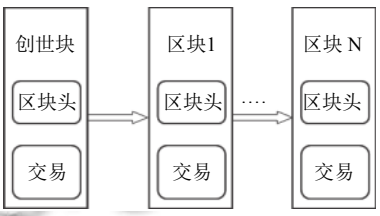

图 1 区块链结构示意图

2.2 智能合约

智能合约是一种在区块链上自动验证、不可逆转、可编程执行的计算机协议[16]. 最早是由密码学家 Nick Szabo 在 1994 年提出, 设想将传统合同条款转换成代码, 并将它们嵌入到可以自动执行的硬件或软件中, 以便最大限度地减少交易者之间对可信第三方的依赖. 区块链技术的诞生, 以及基于 C++、Solidity 等高级语言的智能合约开发平台的发布, 使设计具有可用性的智能合约变成现实. 智能合约本质上是一个状态机, 包括一组可执行函数、状态变量以及标识地址. 合约部署者指定相关的权限确认逻辑并把已完成编译的合约上传到区块链后, 其他用户可以通过标识地址向指定合约发起一笔事务 (包括执行函数所需的输入

参数), 从而触发对应合约中相应的执行函数, 返回执行结果并更新合约的状态. 因此, 智能合约赋予区块链计算处理能力, 开发者可以通过合理的逻辑函数管理和控制链上数据.

2.3 密文—策略基于属性的加密

密文—策略基于属性的加密运用密码机制保护数据, 由发送方规定访问密文的策略, 将属性集合与访问资源相关联, 接收方可以根据自己的授权属性访问密文信息, 适合隐私数据共享等访问控制类应用[17,18]. CP-ABE 主要由 4 个多项式算法组成:

(1) 初始化: 初始化算法为随机化算法, 一般在可信的密钥分发中心上执行. 如式 (1), 算法输入安全系数 λ 和属性空间 U, 生成系统公钥 PSK 和系统主密钥 MSK.

$$Setup(\lambda, U) \rightarrow (PSK, MSK) \tag{1}$$

(2) 密钥生成: 密钥生成算法为随机化算法, 一般由可信的密钥分发中心执行. 如式 (2), 根据系统公钥 PSK、系统主密钥 MSK 和数据请求者提交的属性集合 A, 为数据请求者生成与属性集合相关联的用户密钥 USK.

$$KeyGen(PSK, MSK, A) \rightarrow USK \tag{2}$$

(3) 加密: 加密算法为随机化算法, 由数据拥有者执行. 如式 (3), 算法输入系统公钥 PSK、待加密消息 T 和与访问策略相关联的访问控制结构 A_{cp}, 生成基于属性加密的密文 CT. 只有拥有满足访问策略的请求者才能解密密文 CT.

$$Encrypt(PSK, T, A_{cp}) \rightarrow CT \tag{3}$$

(4) 解密: 解密算法为确定性算法, 由数据请求者执行. 如式 (4), 算法的输入为系统公钥 PSK、用户密钥 USK 和密文 CT, 如果属性集合 A 满足访问策略, 算法自动解密密文并获得相应的明文数据 T.

$$Decrypt(PSK, USK, CT) \rightarrow T \tag{4}$$

3 DOB 框架

3.1 系统模型

DOB 框架由数据注册模块、密钥生成模块、策略管理模块和传输模块组成, 通过 4 个模块的相互协同, 实现数据的安全共享. DOB 框架包含 3 个实体:

1) DO: 数据拥有者, 实际上是拥有数据的个人或者机构. 主要负责发布数据集元数据, 设置访问策略, 分发密钥和传输基于属性加密的密文.

2) DR: 拥有部分属性集合的数据请求者, 只有属性集合符合访问策略才能拥有访问数据集的权限.

3) AC: 可信的密钥分发中心, 负责生成公开参数, 为 DO 和 DR 生成和分发密钥.

3.1.1 数据注册模块

数据注册模块将数据以相对成熟的元数据标准为核心进行统一描述后, 生成包含数据集哈希值、数据集所有者等特定标识字段的元数据, 通过智能合约将元数据发布到区块链网络上, 供所有节点浏览发现数据集, 利用区块日志不可篡改的特性为数据集来源追踪和数字资产确权提供保障.

3.1.2 密钥生成模块

密钥生成模块负责生成加密数据所需的密钥. 数据加密算法一般分为对称加密算法和非对称加密算法. 对称加密算法也称为单密钥算法, 要求发收信双方在建立通信之前, 商定一个用于加解密数据的密钥. 非对称加密算法需要公钥和私钥对数据进行加密和解密. 相比于非对称加密, 对称加密的计算量小, 加密速度快, 当加密大量数据时其实现效率要比非对称加密高 6–10 倍, 但密钥的分发缺乏安全, 算法的安全性在很大程度上依赖于密钥. 考虑到需要加密的共享数据集数量比较大, 为了保证高效的加解密效率, 同时有效避免恶意攻击者通过暴力破解的方式获取密钥, 本模块采用对称加密算法 AES-256 加密数据, 生成长度为 256 位的密钥, 该密钥由一对 256 位随机数进行加法运算后执行哈希算法 SHA256 得出. 密钥管理和分发交由策略管理模块和传输模块处理.

3.1.3 策略管理模块

策略管理模块主要实现密文—策略基于属性的加密. 在此模块里, AC 负责生成系统公钥、系统主密钥和用户密钥. DO 负责设置包含时间维度的访问策略, 并将密钥和访问策略嵌入到密文中, 密文只有在有效时间内才可被解密. DR 负责将用户密钥用于解密密文以获得密钥.

3.1.4 传输模块

传输模块主要利用智能合约设置用户节点的访问权限, 同时运用序列化技术, 实现系统公钥、用户属性、密文和用户密钥等在区块链上的安全高效传输.

3.2 运作流程

如图 2 所示, DOB 框架的运作流程为:

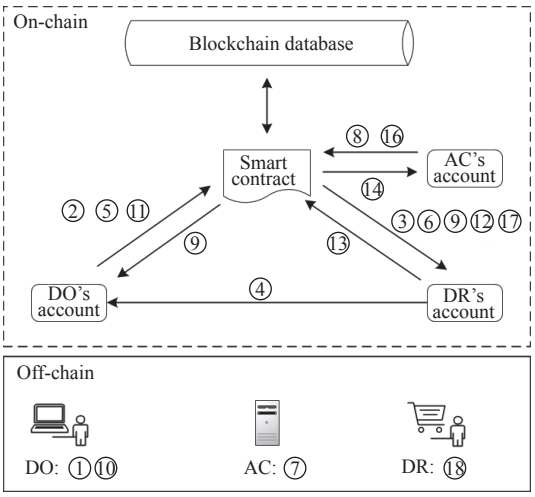

图 2 DOB 框架运作流程图

① DO 随机生成 256 位密钥对 k_1、k_2, 执行哈希算法获得对称加密密钥: k=SHA256($k_1 \oplus k_2$).

② DO 调用智能合约在区块链发布数据集元数据.

③ DR 调用智能合约浏览数据集元数据, 查看数据集详细描述信息.

④ DR 创建一笔交易向 DO 发送访问请求, 访问请求包括目标数据集标识名和目标数据集 MD5 值.

⑤ DO 收到访问请求后, 调用智能合约将密钥 k_1 存储在链数据库上, 设置访问权限仅对 DR 开放访问.

⑥ DR 调用智能合约从链数据库中获取密钥 k_1.

⑦ AC 执行 CP-ABE 初始化算法, 输入安全参数和属性空间 U, 生成系统公钥 *PSK* 和系统主密钥 *MSK*.

⑧ AC 通过智能合约将系统公钥 *PSK* 存储在链数据库上, 将访问权限默认设置为被所有节点账户访问.

⑨ DO 和 DR 调用智能合约从链数据库获取系统公钥 *PSK*.

⑩ DO 执行 CP-ABE 加密算法. 输入系统公钥 *PSK*、密钥 k_2 和访问策略 policy, 生成密文k_2'. 其中, 访问策略的有效时间 T 设置为: $T \geq 1552312182 \cup T \leq 1583848182$.

⑪ DO 调用智能合约将密文k_2'存储在链数据库上, 添加访问限制仅对 DR 开放.

⑫ DR 调用智能合约从链数据库获取密文k_2'.

⑬ DR 调用智能合约将其带有当前时间的属性集合 A 存储在链数据库上, 并添加访问限制仅对 AC 开放.

⑭ AC 调用智能合约从链数据库获取属性集合 A.

⑮ AC 执行 CP-ABE 密钥生成算法, 输入系统主密钥 *MSK*, 系统公钥 *PSK* 和属性集合 A, 生成用户密钥 *USK*.

⑯ AC 调用智能合约将 *USK* 存储在链数据库上, 添加访问限制仅对 DR 开放.

⑰ DR 调用智能合约从链数据库获取 *USK*.

⑱ DR 执行 CP-ABE 解密算法, 输入系统公钥 *PSK*、用户密钥 *USK* 和密文k_2', 若 DR 的属性集合满足密文中的访问策略 *policy*, 则自动解密密文获得密钥 k_2, 从执行哈希运算获得对称加密密钥: k=SHA256($k_1 \oplus k_2$). 否则, 无权访问数据集.

3.3 智能合约算法细节

DOB 框架调用的智能合约共有 12 个功能入口, 其算法描述如下:

(1) 发布元数据: 此算法只有 DO 才能执行. DO 创建一个多索引表, 定义元数据所有字段的数据类型, 将数据集标识名设置为多索引表的主索引, 最后把输入的元数据序列化, 存储在多索引表中.

算法 1. 发布元数据

输入: 元数据 (metadata), 数据集标识名 (dataset's identifier)
输出: 空值

```
1.  if vertify(DO's account)⇐false then
2.      throw;
3.  end
4.  else
5.      typedef a multi-index table
6.      set dataset's identifier of metadata as index
7.      serialize (metadata)
8.      break;
9.  end
```

(2) 查询元数据: 此算法由 DR 执行. DR 根据数据集标识名查找对应的多索引表中的数据对象, 如果查找成功, 返回数据集元数据. 否则, 查找失败.

算法 2. 查询元数据

输入: 数据集标识名 (dataset's identifier)
输出: 元数据 (metadata)

```
1.  metadata⇐find (dataset's identifier)
2.  if  metadata is null then
3.      throw;
4.  end
5.  else
6.      return metadata
7.  end
```

(3) 序列化密钥 k_1: 此算法由 DO 执行. DO 创建一个多索引表, 将 DO 节点账户名设置为多索引表的主

索引, 将输入的密钥 k_1 序列化, 存储在多索引表中, 最后设置多索引表访问权限仅对 DR 开放.

算法 3. 序列化密钥 k_1

输入: 密钥 k_1, DO 节点账户 (DO's account)
输出: 布尔值 (bool)

```
1.    typedef a multi-index table
2.    set DO's account as index
3.    serialize (k1)
4.    authorizeUsers(DR's account)⇐true
5.    return true;
6.  end
```

(4) 提取密钥 k_1: 此算法只有 DR 才能执行. 算法初始会验证运行节点账户是否属于 DR, 若验证不成功, 则终止算法. 若验证成功, DR 根据 DO 节点账户名查找对应的多索引表中的密钥 k_1, 如果查找成功, 提取密钥 k_1. 否则, 提取失败.

算法 4. 提取密钥 k_1

输入: DO 节点账户 (DO's account)
输出: 密钥 k_1

```
1.    if vertify (DR's account)⇐false then
2.        throw;
3.  end
4.  else
5.      k1⇐find (DO's account)
6.      if k1 is null then
7.          throw;
8.    else
9.          return k1
10.   end
11. end
```

(5) 序列化系统公钥: 此算法由 AC 执行. AC 创建一个多索引表, 将其节点账户名设置为多索引表的主索引, 并将系统公钥序列化, 存储在多索引表.

算法 5. 序列化系统公钥

输入: 系统公钥 (*PSK*), AC 节点账户 (AC's account)
输出: 空值

```
1.        typedef a multi-index table
2.        set AC's account as index
3.        serialize (PSK)
4.        break;
5.  end
```

(6) 提取系统公钥: 此算法由 DO 和 DR 执行. 根据 AC 节点账户名查找对应多索引表的系统公钥, 若查找成功则提取系统公钥. 否则, 提取失败.

算法 6. 提取系统公钥

输入: AC 节点账户 (AC's account)
输出: 系统公钥 (*PSK*)

```
1.    PSK⇐find (AC's account)
2.    if PSK is null then
3.        throw;
4.    end
5.    else
6.        return PSK
7.  end
```

(7) 序列化密文k_2': 此算法由 DO 执行. DO 创建一个多索引表, 将其节点账户名设置为多索引表的主索引, 并把密文k_2'序列化, 存储在多索引表中. 最后设置多索引表访问权限仅对 DR 开放.

算法 7. 序列化密文k_2'

输入: 密钥 k_1, DO 节点账户 (DO's account)
输出: 布尔值 (bool)

```
1.    typedef a multi-index table
2.    set DO's account as index
3.    serialize (k2')
4.    authorizeUsers (DR's account)⇐true
5.    return true;
6.  end
```

(8) 提取密文k_2': 此算法只有 DR 才能执行. 初始情况下, 算法会验证运行节点账户是否属于 DR, 若验证不成功, 则终止算法. 若验证成功, DR 根据 DO 节点账户名查找对应的多索引表中的密文k_2', 如果查找成功, 提取密文k_2'. 否则, 提取失败.

算法 8. 提取密文k_2'

输入: DO 节点账户 (DO's account)
输出: 密文k_2'

```
1.  if vertify (DR's account)⇐false then
2.      throw;
3.  end
4.  else
5.      k2'⇐find (DO's account)
6.      if k2' is null then
7.          throw;
8.      else
9.          return k2'
10.   end
11. end
```

(9) 序列化用户属性: 此算法由 DR 执行. DR 创建一个多索引表, 定义用户属性所有字段的数据类型, 将

其节点账户名设置为多索引表的主索引, 最并将用户属性序列化, 存储在多索引表中. 最后设置多索引表访问权限仅对 AC 开放.

算法 9. 序列化用户属性

输入: DR 节点账户 (DR's account), 用户属性 (attribute)
输出: 布尔值 (bool)

```
1.      typedef a multi-index table
2.      set DR's account as index
3.      serialize (attribute)
4.      authorizeUsers(AC's account)⇐true
5.      return true;
6.  end
```

(10) 提取用户属性: 此算法只有 AC 才能执行. 算法初始会验证运行节点是否为 AC 节点, 若验证不成功, 则终止算法. 若验证成功, AC 根据 DR 节点账户名查找对应的多索引表中的用户属性, 如果查找成功, 提取用户属性. 否则, 提取失败.

算法 10. 提取用户属性

输入: DR 节点账户 (DR's account)
输出: 用户属性 (attribute)

```
1.  if vertify (AC's account)⇐false then
2.      throw;
3.  end
4.  else
5.      attribute⇐find (DR's account)
6.      if attribute is null then
7.          throw;
8.      else
9.          return attribute
10.     end
11.  end
```

(11) 序列化用户密钥: 此算法由 AC 执行. AC 创建一个多索引表, 将自身节点账户名设置为该多索引表的主索引, 并将用户密钥序列化, 存储在多索引表中. 最后设置多索引表访问权限仅对 DR 开放.

算法 11. 序列化用户密钥

输入: AC 节点账户 (AC's account), 用户密钥 (*USK*)
输出: 布尔值 (bool)

```
1.      typedef a multi-index table
2.      set AC's account as index
3.      serialize (USK)
4.      authorizeUsers (DR's account)⇐true
5.      return true;
6.  end
```

(12) 提取用户密钥: 此算法只有 DR 才能执行. 算法初始会验证运行节点是否为 DR 节点, 若验证不成功, 则终止算法. 若验证成功, DR 根据 AC 节点账户名查找对应多索引表中的用户密钥, 如果查找成功, 提取用户密钥. 否则, 提取失败.

算法 12. 提取用户密钥

输入: AC 节点账户 (AC's account)
输出: 用户密钥 (*USK*)

```
1.  if vertify (DR's account)⇐false then
2.      throw;
3.  end
4.  else
5.      USK⇐find (AC's account)
6.      if USK is null then
7.          throw;
8.      else
9.          return USK
10.     end
11.  end
```

4 实验结果及分析

4.1 实验准备

实验部署在 1 台 PC 机上, 其配置如下: Intel Xeon(R) E5-2407 v2(2.4 GHz)8 核 CPU, 16 GB 内存. 操作系统采用 Ubuntu 16.04.10 LTS desktop, 区块链采用 EOSIO 平台部署, 智能合约采用 C++开发, 密钥采用 Openssl 工具产生, 密文—策略基于属性的加密采用基于双线性对的密码函数库 (Pairing-Based Cryptography library, PBC) 实现.

实验选取 1999 年 1 月至 2008 年 12 月期间的 "130 家医院糖尿病患者病例"数据集[19], 包含一个 diabetic_data.csv 文件, 存有 10 万条糖尿病患者的医疗档案, 包括医疗卡号、性别、年龄、治疗情况等隐私信息. 根据 diabetic_data 数据集的特点, 将数据集元数据分为外部描述核自定义的内部描述两部分, 外部描述按照数据目录词汇表标准[20]定义, 如表 1. 内部描述选用自定义描述词汇, 如表 2. diabetic_data 数据集元数据, 如图 3.

初始条件下, 在区块链网络上设置两个用户节点和一个 AC 节点, 每个用户节点既是数据拥有者, 又是数据请求者. 实验通过控制变量, 在同一环境上分别运行 Timely CPABE with Blockchain 方案和 DOB 框架, 二者皆采用相同的访问策略和属性集合. 实验一共分为 7 次, 每次实验比上次增加 5 种属性数量, 每次实验重复 10 遍, 最后取得平均值.

表 1 diabetic_data 数据集外部描述词汇

词汇	前缀	类型
title	dc	数据集标题
theme	dcat	所属主题
description	dc	描述说明
Identifier	dc	数据集的唯一标识
keyword	dcat	关键词
format	dc	数据集格式
accessURL	dcat	访问链接
byteSize	dcat	数据集大小
license	dc	许可说明
Rights	dc	使用权限信息
Issued	dc	初始发布时间
modified	dc	最后修改时间

表 2 diabetic_data 数据集内部描述词汇

词汇	类型	说明
account_name	string	节点账户
Copyrightowner	string	数据集所属人
Field	string	数据集类别
MD5	string	数据集哈希值


```
[Metadata of diabetic_dataset]
"account_name": "Bob",
"CopyrightOwner": "Bob",
"identifier": "diabetic_dataset",
"title": "Diabetes 130-US hospitals for years 1999-2008 Data Set",
"theme": "This data has been prepared to analyze factors related to
        readmission as well as other outcomes pertaining to
        patients with diabetes.",
"keyword": "Diabetes;1999-2008;Medical cases;US",
"issued": "October 18th,2018 Thursday 20:20:17 CST",
"modified": "December 15th,2018 Saturday 15:20:01 CST",
"accessURL":"127.0.0.1/medical/datasets/Diabetes+130-US+hospitals
           +for+years+1999-2008",
"format": "csv",
"byteSize": "19,161,930 bytes",
"description": "The dataset represents 10 years (1999-2008) of clinical
              care at 130 US hospitals and integrated delivery networks.
              It includes over 50 features representing patient and
              hospital outcomes.",
"rights": "Authorized Access",
"license": "MIT",
"field": "Medical",
"MD5": "f22425753cefbc18e321825450ec0f00"
```


图 3 diabetic_data 数据集元数据

4.2 实验结果及分析

4.2.1 运行时间分析

经实验测试, 两种方案的系统运行时间如图 4.

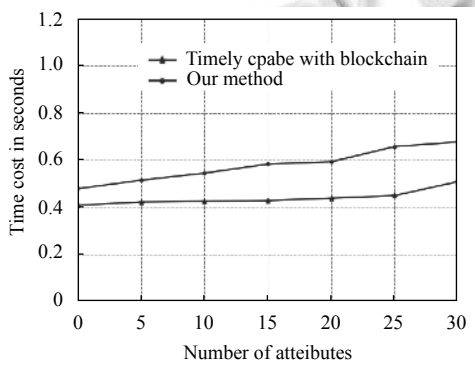

图 4 系统运行时间示意图

从图 4 可以看出, 随着属性数量的增加, 所需传输的密文和用户密钥的空间大小逐渐增大, 使得二者的系统运行时间逐步增加. 相对于 Timely CPABE with Blockchain 方案, DOB 框架的平均系统运行时间增加了 31.18%, 主要原因是采用序列化方法在链数据库上存储和传输密文、相关密钥和用户属性, 增加链数据库的读写次数. 但是, 系统运行时间远远小于 1 秒, 仍处于合理范围之内.

4.2.2 安全性分析

Timely CPABE with Blockchain 方案将合法请求者的用户属性公开存取在区块, 恶意节点可以盗用合法者的用户属性生成 *USK*, 从而将获得的密钥 k_2 与密钥 k_1 结合生成数据加密密钥 k. DOB 通过在链数据库上设置访问权限, 只有授权的数据请求者才能提取出密文、用户密钥 *USK* 和密钥 k_1, 因此 DOB 框架能降低 *USK* 被盗用的风险.

5 结论与展望

根据数据共享的安全需求, 本文提出一种基于 CP-ABE 和区块链的数据安全共享框架, 通过带有访问权限的智能合约部署密文—策略基于属性的加密, 提高了获取和跟踪共享数据访问许可的自动化程度, 使数据的管理使用权真正掌握在数据拥有者手中. 实验结果表明, DOB 框架的综合性能比 Jemel 等人提出的 Timely CPABE with Blockchain 方案表现得要好, 具有一定的积极意义. 由于访问策略是数据拥有者事先定义的, DOB 框架的访问机制相对固定, 适用于数据集自身更新频度小、共享范围相对确定的场景, 以避免数据加密的密钥频繁更换. 当然, 方案仍存在一些值得完善的地方, 例如属性灵活撤销、高效访问结构的设计以及访问策略动态更新, 以应用于动态共享场景, 将是下一步的研究方向.

参考文献

1 刘海房, 莫世鸿, 范冰冰. 开放数据最新进展及趋势. 情报杂志, 2016, 35(9): 163–167. [doi: 10.3969/j.issn.1002-1965.2016.09.029]

2 金泳, 徐雪松, 王刚, 等. 基于区块链的电子政务大数据安全共享研究. 信息安全研究, 2018, 4(11): 1029–1033. [doi: 10.3969/j.issn.2096-1057.2018.11.011]

3 Ahmadi Zeleti F, Ojo A, Curry E. Exploring the economic value of open government data. Government Information Quarterly, 2016, 33(3): 535–551. [doi: 10.1016/j.giq.2016.01.008]

4 中华人民共和国国务院. 促进大数据发展行动纲要. 成组技术与生产现代化, 2015, 32(3): 51–58. [doi: 10.3969/j.issn.1006-3269.2015.03.012]
5 冯登国, 张敏, 李昊. 大数据安全与隐私保护. 计算机学报, 2014, 37(1): 246–258.
6 戚学祥. 区块链技术在政府数据治理中的应用: 优势、挑战与对策. 北京理工大学学报 (社会科学版), 2018, 20(5): 105–111.
7 郭兵, 李强, 段旭良, 等. 个人数据银行——一种基于银行架构的个人大数据资产管理与增值服务的新模式. 计算机学报, 2017, 40(1): 126–143. [doi: 10.11897/SP.J.1016.2017.00126]
8 何蒲, 于戈, 张岩峰, 等. 区块链技术与应用前瞻综述. 计算机科学, 2017, 44(4): 1–7, 15. [doi: 10.11896/j.issn.1002-137X.2017.04.001]
9 Xu XW, Pautasso C, Zhu LM, *et al*. The blockchain as a software connector. Proceedings of 13th Working IEEE/IFIP Conference on Software Architecture. Venice, Italy. 2016. 182–191.
10 Di Francesco Maesa D, Mori P, Ricci L. Blockchain based access control. Proceedings of the 17th IFIP WG 6.1 International Conference, DAIS 2017, Held as Part of the 12th International Federated Conference on Distributed Computing Technique. Neuchâtel, Switzerland. 2017. 206–220.
11 Wang SP, Zhang YL, Zhang YL. A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems. IEEE Access, 2018, 6: 38437–38450. [doi: 10.1109/ACCESS.2018.2851611]
12 Jemel M, Serhrouchni A. Decentralized access control mechanism with temporal dimension based on blockchain. Proceedings of the IEEE 14th International Conference on E-business Engineering. Shanghai, China. 2017. 177–182.
13 袁勇, 王飞跃. 区块链技术发展现状与展望. 自动化学报, 2016, 42(4): 481–494.
14 刘敖迪, 杜学绘, 王娜, 等. 区块链技术及其在信息安全领域的研究进展. 软件学报, 2018, 29(7): 2092–2115. [doi: 10.13328/j.cnki.jos.005589]
15 Dannen C. Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners. New York: Apress, 2017.
16 Clack CD, Bakshi VA, Braine L. Smart contract templates: Foundations, design landscape and research directions. arXiv preprint arXiv: 1608.00771, 2016.
17 Waters B. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. Proceedings of the 14th International Conference on Practice and Theory in Public Key Cryptography. Taormina, Italy. 2011. 53–70.
18 苏金树, 曹丹, 王小峰, 等. 属性基加密机制. 软件学报, 2011, 22(6): 1299–1315.
19 Strack B, Deshazo JP, Gennings C, *et al*. Impact of HbA1c measurement on hospital readmission rates: Analysis of 70000 clinical database patient records. BioMed Research International, 2014: 781670.
20 于梦月, 翟军, 林岩. 我国地方政府开放数据的核心元数据研究. 情报杂志, 2016, 35(12): 98–104. [doi: 10.3969/j.issn.1002-1965.2016.12.018]