doi :10.3969/j.issn.1671-1122.2020.08.003

# 基于区块链与代理重加密的数据共享方案

李莉[1]，曾庆贤[1]，文义红[2]，王士成[2]

（1. 武汉大学国家网络安全学院，武汉 430072；2. 中国电子科技集团公司第五十四研究所，石家庄 050081）

**摘　要：**在不可信环境下实现安全的数据共享一直是一个难题，传统中心化方案存在数据容易被泄露、数据容易被篡改、数据去向难以追踪、监管难度大等问题。公钥体制下的数据共享方案则存在通信代价、计算开销大，实用性差等问题。针对上述问题，文章提出一个基于区块链的数据共享方案。该方案通过区块链维护一个可信账本来保证数据的可追溯性，同时也保证了访问控制权限的不可篡改。在此基础上构建了基于 Schnorr 的代理重加密方案，通过一个代理重加密密钥实现了数据的安全共享。文章所提方案相比传统方案有较好的安全性和可追溯性，且已成功应用于医疗数据共享项目中。

**关键词：**区块链；代理重加密；数据共享；Schnorr 签名

**中图分类号：**TP309　**文献标志码：**A　**文章编号：**1671-1122（2020）08-0016-09

中文引用格式：李莉，曾庆贤，文义红，等 . 基于区块链与代理重加密的数据共享方案 [J]. 信息网络安全，2020，20（8）：16-24.

英文引用格式：LI Li, ZENG Qingxian, WEN Yihong, et al. Data Sharing Scheme Based on the Blockchain and the Proxy Re-encryption[J]. Netinfo Security, 2020, 20(8): 16-24.

## Data Sharing Scheme Based on the Blockchain and the Proxy Re-encryption

LI Li[1], ZENG Qingxian[1], WEN Yihong[2], WANG Shicheng[2]

(1. *School of Cyber Science and Engineering*, *Wuhan University*, *Wuhan* 430072, *China*;2. *The* 54th *Research Institute of China Electronics Technology Group Corporation*, *Shijiazhuang* 050081, *China*)

**Abstract:** Achieving secure data sharing in an untrusted environment is always a difficult problem. Traditional centralized solutions have problems that data is easily leaked, data is easily tampered, data destination is difficult to track, and supervision is difficult. Data sharing scheme based on public-key system has some problems such as high communication cost, high computation cost and poor practicability. To solve the above problems, this paper proposes a data sharing scheme based on blockchain. The scheme maintains a credible ledger through the blockchain to ensure the traceability of data and immutability of the access control authority. On this basis, a proxy re-encryption scheme based on Schnorr is constructed, which realizes the secure data sharing by a proxy re-encryption secret key. Compared with the traditional

**收稿日期：**2020-6-28

**基金项目：**国家重点研发计划 [2018YFC1315404]

**作者简介：**李莉（1976—），女，湖北，副教授，博士，主要研究方向为应用密码安全、区块链、物联网安全等；曾庆贤（1996—），男，江西，硕士研究生，主要研究方向为区块链、物联网安全；文义红（1977—），男，湖北，高级工程师，博士，主要研究方向为遥感图像视频分析处理、航天区块链应用、航天系统仿真及应用等；王士成（1976—），男，河北，研究员，硕士，主要研究方向为人工智能与大数据、空间态势感知、航天地面应用等。

**通信作者：**李莉 lli@whu.edu.cn

schemes, the proposed scheme has better security and traceability, and has been successfully applied in the medical data sharing project.

**Key words:** blockchain; proxy re-encryption; data sharing; Schnorr algorithm

## 0 引言

数据安全一直是一个比较重要的话题[1]。随着大数据的应用发展，数据共享需求对数据安全提出了更多挑战。如何在保证数据隐私的同时，安全、可追溯地共享所需数据成为研究的热点。传统的中心化方案通常存在数据缺乏可信度、数据流向不透明、事后难以追责的问题[2]，这些问题成为设计数据共享系统的最大障碍。

本文针对上述问题，提出一个基于区块链的数据共享系统。该系统由多个组织参与，每个组织控制一定数量的节点。数据拥有者在上传数据的同时，通过元数据（Metadata）分配数据的访问权限，保证了拥有者对数据权限有较高的控制力。相较于其他区块链方案，本文方案使用基于Schnorr算法的代理重加密技术保证数据共享的安全可靠。区块链和代理重加密的结合使得可以实现安全、完整、可追溯的数据共享。

## 1 相关工作

当前已有一些数据共享方案。例如，文献[3]使用粒子群算法对数据进行加工，保护其中的敏感数据；针对垂直数据集中的敏感数据问题，文献[4]采用基于安全多方计算保护决策树的方案；文献[5]提出基于云计算的数据共享方案。这些尝试都在一定程度上做到了对隐私数据的保护，但中心化数据本身的安全问题却难以解决。中心化系统必须建立在一个诚实的第三方之中，并且能够抵御外在的攻击，其内部成员也要有严格的权限控制。这些问题大大增加了建立中心化数据系统的成本。

区块链的出现为问题的解决提供了一个全新的思路。区块链是一个由多个对等节点参与的分布式账本[6,7]，通过智能合约来保证记账的安全性与完整性。在区块链上，一系列对等节点通过智能合约提供的接口完成访问控制和数据读写操作，并将这些操作的日志永久保存在链上，使得数据流向清晰透明。现今已有一些区块链在数据共享上的研究和应用。文献[8]针对用户可穿戴设备场景提出一个以用户为中心的隐私数据可控的数据共享方案。文献[9]利用对称可搜索技术，在区块链上建立了一个数据可搜索平台，在保证数据安全的同时实现了数据的可搜索性。针对用户之间的数据安全传输、共享，文献[10]也提出了基于区块链的方案，并且使用PoW（Proof of Work，工作量证明）机制来防止恶意节点对区块链的攻击。

## 2 相关技术

### 2.1 区块链技术

区块链技术是从比特币底层抽象出来的一种新兴技术[11]。它是由全网节点维护的一个分布式公开账本，数据结构类似于C语言中的链表，链上的每个节点保存了最近的交易记录以及上一个区块的哈希值，因为任何对数据的改动都会使得哈希值发生巨大变化。该特性带来的结果是，除非节点具有超过全网51%的算力，否则无法对整条链进行分叉，这就避免了常见的“双花问题”，也因此保证了区块的不可更改。区块链保存在所有参与的节点中，也避免了单点失效的问题，且能防止恶意节点对数据进行篡改。区块链是一个去中心化的网络，不需要一个权威的第三方对其进行监管和控制，通过分布式协议和智能合约技术完成相应的功能。

区块链根据开放程度分为3个类型：公有链、私有链以及联盟链。公有链对所有人公开，根据一定的算法，每个人都能成为其中的一个节点并参与计算和读写。私有链是高度集中的区块链，通常只有一个组织，只有组织信任的节点才能参与区块链进行账本的读写。联盟链介于两者之间，通常由几个组织构成，每个组织管理了一些节点，这些节点维护了区块链的各项功能。因为联盟链提供的模式能够满足数据共享的许多

应用场景，所以其被选为很多项目的底层区块链模式。

### 2.2 Fabric

Fabric是由不同组织参与开发的一个联盟链项目[12]，是开源项目超级账本中比较成熟的一个，面向企业级区块链开发。其网络主要由3种节点构成：背书节点、确认节点和排序节点，此外还有一个可插拔的认证中心（Certificate Authority，CA）节点。

背书节点负责对交易进行背书，根据已有规则对数据进行读写操作。在收到交易的提案之后，背书节点首先对其中的签名进行验证，并检查提案的发起者是否有权限；然后根据自己的状态数据库进行模拟交易，把结果生成读集（Read Set）和写集（Write Set），它们代表了对Fabric状态数据库的读写操作集合；最后把执行结果背书并返回给交易的发起者。确认节点负责对背书节点发来的交易进行检查，验证其数据结果。排序节点负责对收到的交易进行排序，打包成区块，并将结果返回给交易对应的节点。CA节点是Fabric中PKI体系下的核心节点，主要负责用户的注册以及证书的颁发和撤销等。

Fabric中比较引人注目的功能是它支持多条链（Channel），每条链支持连接不同的组织，运行独立的链码（Chaincode）。这使得在一个区块链上运行几条独立的链成为可能，也能够隔离不同链在相同节点上的交易。

### 2.3 代理重加密

代理重加密是BLAZE[13]等人提出的一个概念。该机制允许在系统中有一个半诚实的代理者，通过一个代理密钥，将Alice加密的密文转换为Bob可解密的密文。在这个过程中，代理者无法获得任何明文信息[14]。在当前背景下，多数系统的数据存储功能都是由一个不诚实或者半诚实的云服务端实现，代理重加密的特性正好可以避免云服务端的恶意行为。

代理重加密可用于如下场景。Alice每次都通过邮箱接收用自己公钥加密的邮件，假设某天Alice因为时间问题不能及时处理邮件，因此她想将邮件转发给自己的助理Bob，让Bob来处理这些邮件。因为邮件的内容是利用Alice的公钥加密的密文信息，因此Bob无法获取邮件的明文信息。一个简单的办法就是Alice直接把自己的私钥发送给Bob，但这会产生一些问题：首先如果直接发送密钥明文，那邮件的加密就失去了意义，一旦Bob的邮件服务器被攻破，邮件的明文就会被泄露；其次Alice可能不希望自己的私钥被别人直接获取。在这种场景下，代理重加密就有了用武之地。Alice可以根据Bob的公钥和自己的私钥产生一个代理重加密密钥，放在自己的邮箱服务器的数据库中。在收到密文邮件后，邮箱服务器可以利用这个代理重加密密钥，将密文进行一次重加密，将重加密后的密文转发给Bob，Bob收到这条重加密的密文消息后，就可以利用自己的私钥对密文进行解密，进而还原出明文消息。

代理重加密分多种形式。根据密文转换方向的不同可分为单向代理重加密和双向代理重加密。单向代理重加密只允许将Alice能解密的密文转换为Bob能解密的密文。双向代理重加密除了允许将Alice能解密的密文转换为Bob能解密的密文，同时也可以将Bob能解密的密文转换为Alice能解密的密文。根据重加密密钥转换次数的不同，可以分为单跳密文转换和多跳密文转换。本文使用的是基于单跳单向的代理重加密方案，即只允许对一个密文进行从Alice能解密到Bob能解密的密文转换。

### 2.4 Schnorr 签名

Schnorr签名是一种基于离散对数难题的签名方案。这是一种简单、高效的签名算法，能够在一些资源受限的环境下发挥良好的性能。此外，Schnorr签名还具有可证明安全性，能够在随机预言机模型下很好地证明Schnorr签名的安全性。

## 3 系统架构

本文引入区块链，并基于Fabric链码设计了一个数据共享平台系统方案。系统的主要角色有数据拥有者、数据请求者和半诚实第三方代理。

数据拥有者是数据的原始持有者，他将共享数据密文、相关密钥以及元数据上传至代理服务器。数据请求

者搜索数据的元数据，并对感兴趣的数据申请下载。半诚实第三方代理负责保存数据拥有者上传的数据，并遵循上传下载等协议为数据拥有者和请求者提供数据存储和使用等服务。半诚实第三方代理虽然遵循协议执行，但却可能保存协议的中间计算状态。第三方代理首先确认数据请求者的下载资格，如果请求者具有资格，则代理方会发送密文以及重加密后的对称密钥的密文给数据请求者,数据请求者通过自己的私钥解密数据,获得明文。

本文方案在代理服务器端通过密码算法保证数据一直以密文形式存储，数据请求者只有在成功请求数据后,才能解密得到数据明文。系统功能包括上传文件、搜索文件列表和下载对应文件。系统架构如图1所示，主要分为3个模块：前端模块、代理服务器模块以及区块链模块。

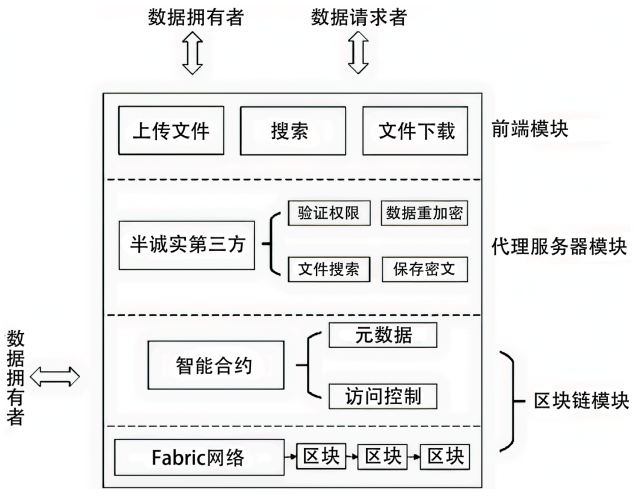



图 1 系统架构

前端模块主要为数据拥有者和数据请求者提供可视化的操作接口，方便用户进行数据上传、搜索以及下载等相关操作。

代理服务器模块扮演半诚实第三方的角色，主要有4个功能：1）保存数据拥有者上传的数据密文和相应的元数据。2）根据用户提供的关键字在元数据中搜索对应文件，并将结果返回给请求者。3）接受对文件的下载请求，并将请求者信息和文件信息发送给区块链模块，由区块链模块验证用户是否有对应这个文件的下载权限。区块链模块验证之后，会对此次请求结果生成日志上链。4）若从区块链模块确认用户有权限下载文件，则通过一个代理重加密密钥$rk$对经对称密钥加密的密文进行重加密。

区块链模块分为两个部分：智能合约和底层数据区块。智能合约是一组定义了区块链系统逻辑的核心合约代码，由管理员编写，在系统初始化阶段部署在各个节点上，提供远程接口供用户调用[15]。这种合约不仅能够免除因为人工失误而造成的错误，还能避免恶意操作者对数据进行恶意修改和破坏。在Fabric中，智能合约就是链码（Chaincode）。本文系统中，智能合约主要负责管理3种数据：元数据、访问控制数据以及日志信息。系统通过调用智能合约的API将数据的元数据以及访问控制数据上链。底层数据区块主要负责存储相关数据，也就是链上数据。它是一个类似链表的数据结构，每个节点保存了上一个节点的哈希值，以保证链上数据的不可更改性。

## 4 方案设计

### 4.1 主要流程

基于第3章的系统架构，本文设计了基于区块链与代理重加密的数据共享方案。数据共享流程如图2所示，主要包括数据加密及上传、数据下载并解密两个过程。

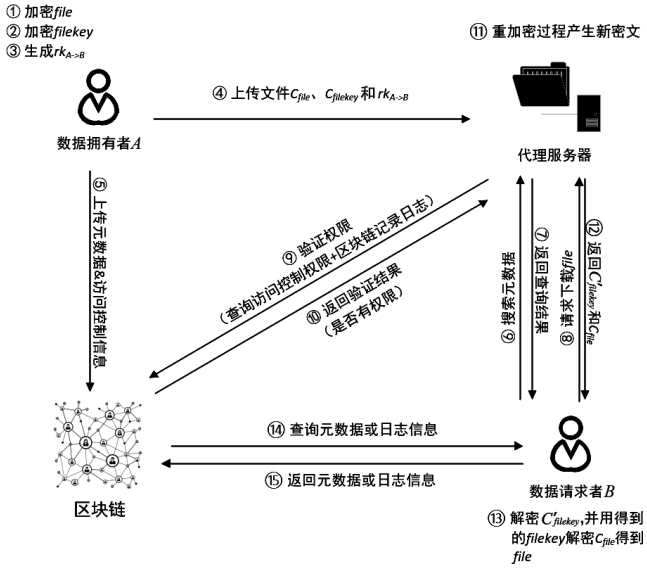



图 2 基于区块链与代理重加密的数据共享流程图

详细描述过程之前，先给出方案涉及的符号、密钥和密码操作，分别如表1、表2和表3所示。

**表 1 符号及其含义**

| 符号 | 含义 |
|---|---|
| $A$ | 数据拥有者 |
| $B$ | 数据请求者 |
| CA | 证书颁发机构 |
| $l$ | 系统安全参数 |
| $q$ | $l$ bit 的大素数 |
| $G$ | 阶为 $q$ 的加法循环群 |
| $P$ | 群 $G$ 的生成元 |
| $\mathbf{F}_q^*$ | 模 $q$ 的正整数域 |
| $id_A$，$id_B$ | 用户 $A$、$B$ 的身份标识，可以唯一确定用户的公钥 |

**表 2 密钥及其含义**

| 密钥 | 含义 |
|---|---|
| $P_{\mathrm{CA}}$ | CA 的公钥 |
| $P_A$ | 用户 $A$ 的公钥 |
| $P_B$ | 用户 $B$ 的公钥 |
| $d_{\mathrm{CA}}$ | CA 的私钥 |
| $d_A$ | 用户 $A$ 的私钥 |
| $d_B$ | 用户 $B$ 的私钥 |
| $file$ | 待加密数据文件 |
| $filekey$ | 用于加密 $file$ 的对称密钥，与 $file$ 一一对应 |
| $rk_{A\to B}$ | 用户 $A$ 授权给用户 $B$ 的重加密密钥 |

**表 3 密码操作及其含义**

| 密码操作 | 含义 |
|---|---|
| $E_{filekey}(file)$ | 使用 $filekey$ 对 $file$ 进行加密保护，生成密文 $C_{file}$ |
| $E_{P_A}(filekey)$ | 使用 $P_A$ 对 $filekey$ 进行加密保护，生成密文 $C_{filekey}$ |
| $DEC_{d_B}(C'_{filekey})$ | 使用 $d_B$ 解密密文 $C'_{filekey}$，得到 $filekey$ |
| $DEC_{filekey}(C_{file})$ | 使用 $filekey$ 解密 $C_{file}$，得到明文 $file$ |
| $Re\text{-}Enc(C_{filekey})$ | 重加密 $C_{filekey}$，得到 $C'_{filekey}$ |

## 4.2 系统初始化

数据拥有者和数据请求者首先要在CA注册，获取自己的证书。CA建立且颁发证书的相关流程本文不再赘述，这里只给出其中的两个主要流程，具体如下：

1）生成系统参数和系统主公私钥

（1）输入安全参数$l$,CA选择一个$l$bit的大素数$q$，生成一个$q$阶加法循环群$G$，令$P$为$G$的一个生成元；

（2）CA选择一个随机数$d_{\mathrm{CA}}\in\mathbf{F}_q^*$作为主私钥，计算$P_{\mathrm{CA}}=d_{\mathrm{CA}}P$为系统主公钥；

（3）CA选择两个安全杂凑函数：$H_1$:$\{0,1\}^{64}\times G\to\mathbf{F}_q^*$，$H_2$:$G\times\{0,1\}^{64}\times\{0,1\}^{64}\to\mathbf{F}_q^*$；

（4）CA秘密保存系统主私钥$d_{\mathrm{CA}}$，公开系统参数$params=\{G,q,P,P_{\mathrm{CA}},H_1,H_2\}$。

2)用户$U$生成公私钥对，CA为用户$U$生成证书

（1）用户$U$产生一个随机数$d_U\in\mathbf{F}_q^*$作为私钥，并计算公钥$P_U=d_UP$，将$(id_U,P_U)$发送给CA，其中，$id_U$为用户的身份标识；

（2）CA利用私钥$d_{\mathrm{CA}}$为用户$U$的公钥$P_U$生成证书$Cert_U$。

## 4.3 加密及上传数据

1）数据拥有者加密$file$

该步骤对应图2中的①。为了保证数据的安全共享，首先使用对称加密算法对$file$进行加密，得到$C_{file}=E_{filekey}(file)$。将$C_{file}$保存在代理服务器上，供数据请求者下载。这一过程称为Sym_ENC。

2）数据拥有者加密$filekey$

该步骤对应图2中的②。给定$filekey$和时间戳$T_0$，用户$A$利用自身公钥$P_A$对$filekey$加密，具体描述如下：

（1）产生相关的元数据$meta=(id_A||T_0)$；

（2）输入待加密的消息$M=filekey$；

（3）产生随机数$r\in\mathbf{F}_q^*$，并计算点$R=rP$；

（4）计算第一部分密文$C_A=M\oplus H_1(meta||rP_A)$；

（5）计算第二部分密文$h_A=H_2(R||M||meta)$；

（6）计算Schnorr签名部分$z_A=r+h_Ad_A$，若$z_A=0$，返回步骤（3）；

（7）输出$C_{filekey}=(C_A,meta,h_A,z_A)$。

上述过程称为Asy_ENC_with_Schnorr。

3）数据拥有者生成重加密密钥$rk_{A\to B}$

该步骤对应图2中的③。给定原始密文$C_{filekey}=(C_A,meta,h_A,z_A)$，用户$B$的身份$id_B$和证书$Cert_B$，用户$A$产生授权给用户$B$的$rk_{A\to B}$的过程如下：

（1）恢复随机数$r=z_A-h_Ad_A$；

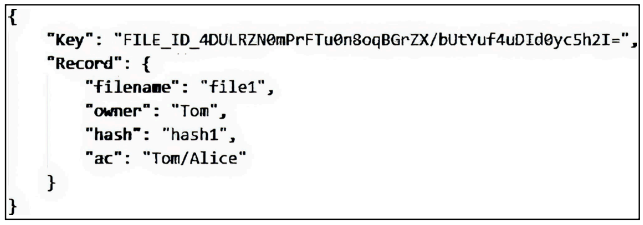
（2）用户$A$利用$P_{\mathrm{CA}}$验证$Cert_B$是否有效,若有效，则从$Cert_B$中获取$P_B$；

（3）计算$rk_{A\to B}=H_1(meta||rP_A)\oplus H_1(meta||rP_B)$。

上述过程称为Gen_ReEncKey。

4）数据拥有者上传数据

该步骤对应图2中的④和⑤。数据拥有者将数据访问控制信息与*file*的元数据通过图1中的区块链模块上传至区块链，将$C_{file}$、$C_{filekey}$以及$rk_{A\to B}$传送至代理服务器。元数据结构如图3所示。



```
{
    "Key": "FILE_ID_4DULRZN0mPrFTu0n8oqBGrZX/bUtYuf4uDId0yc5h2I=",
    "Record": {
        "filename": "file1",
        "owner": "Tom",
        "hash": "hash1",
        "ac": "Tom/Alice"
    }
}
```


图 3 元数据的 JSON 对象结构体

上链保存过程如代码1所示。

**代码1** 元数据上链保存

```
func (s *SmartContract) uploadFile(APIstub shim.ChaincodeStubInterface, args []string) sc.Response {
    var metadata = MetaData{Filename: args[1], Owner: args[2], Hash: args[3], AC: args[4]}
    metadataAsBytes,_ := json.Marshal(metadata)
    APIstub.PutState(args[0], metadataAsBytes)// 保存元数据
    tt := time.Now().Unix()
    log_id := "LOG_"
    log_id += args[0]
    log_id += "_"
    log_id += strconv.FormatInt(tt,10)// 添加日志信息
    var log = Log{Time: strconv.FormatInt(tt,10), User: args[2], Operation:" upload", Result:" success"}
    logAsBytes,_ := json.Marshal(log)
    APIstub.PutState(log_id, logAsBytes)// 将日志信息上链
    return shim.Success(nil)
}
```

## 4.4 下载并解密数据

该步骤对应图2中的⑥~⑮。密文被上传至代理服务器之后，数据请求者便可以通过代理服务器提供的搜索功能进行搜索，进而下载并解密所需的文件。具体流程如下：

1）数据请求者在代理服务器上通过搜索关键词寻找想要的文件，代理服务器返回对应的查询结果。

2）数据请求者在查询结果中寻找自己需要的文件，向代理服务器发出下载文件的请求。代理服务器在区块链上验证该数据请求者是否有相应权限，如代码2所示，并在链上记录一个日志信息，如图4所示，以表示有数据请求者请求了该数据。

**代码2** 验证权限

```
func (s *SmartContract) validateRequest(APIstub shim.ChaincodeStubInterface, args []string) sc.Response {
    metadataAsBytes,_ := APIstub.GetState(args[0])
    metadata := MetaData{}
    json.Unmarshal(metadataAsBytes,&metadata)
    tt := time.Now().Unix()
    log_id := "LOG_"
    log_id += args[0]
    log_id += "_"
    log_id += strconv.FormatInt(tt,10)
    var log = Log{Time: strconv.FormatInt(tt,10), User: args[1], Operation: args[2], Result:" "}
    logAsBytes,_ :=json.Marshal(log)
    if strings.Contains(metadata.AC, args[1]){
        // 有访问权限
        log.Result = "success"
        logAsBytes,_ = json.Marshal(log)
        APIstub.PutState(log_id, logAsBytes)
        return shim.Success(nil)
    }else{
        // 无访问权限
        log.Result = "fail"
        logAsBytes,_ = json.Marshal(log)
        APIstub.PutState(log_id, logAsBytes)
        return shim.Error("Access Denied!" )
    }
}
```

```
{
    "Key": "LOG_FILE_ID_hash10_1582479569",
    "Record": {
        "Time": "1582479569",
        "User": "Tom",
        "Operation": "upload",
        "Result": "success"
    }
}
```


图 4 日志信息的 JSON 对象结构体

3）区块链将验证结果返回给代理服务器。在确认数据请求者有相应权限后，代理服务器利用数据拥有者上传的$rk_{A\to B}$将$C_{filekey}$转换为密文$C'_{filekey}$。具体操作如下：

（1）计算部分重加密密文$C_B = rk_{A\to B} \oplus C_A$；

（2）输出重加密密文$C'_{filekey}=(C_B, meta, h_A, id_B, z_A)$。

上述过程称为Re_ENC。

4）代理服务器将$C_{file}$以及$C'_{filekey}$发还给数据请求者$B$，数据请求者$B$利用自己的私钥将$C'_{filekey}$还原成*filekey*，具体操作如下：

（1）利用$P_{CA}$验证$Cert_A$，并从$Cert_A$中获取$P_A$。

（2）计算点$R'=z_AP-h_AP_A$。

（3）计算消息$M'=C_B\oplus H_1(meta\|d_BR')$。

（4）计算部分密文$h'_A=H_2(R'\|M'\|meta)$。

（5）判断$h'_A$是否与$h_A$相等。

（6）若相等，则接受$M'$；否则，拒绝$M'$。

上述过程称为Asy_DEC_with_Schnorr。此时数据请求者$B$得到了*filekey*=$M'$，数据请求者$B$使用*filekey*解密$C_{file}$获得*file*，这一过程称为Sym_DEC。数据请求者$B$在获得*file*之后，便可在区块链上验证元数据和日志信息，验证通过之后，整个数据共享过程结束。

综上所述，图5给出本文方案的加解密流程图。

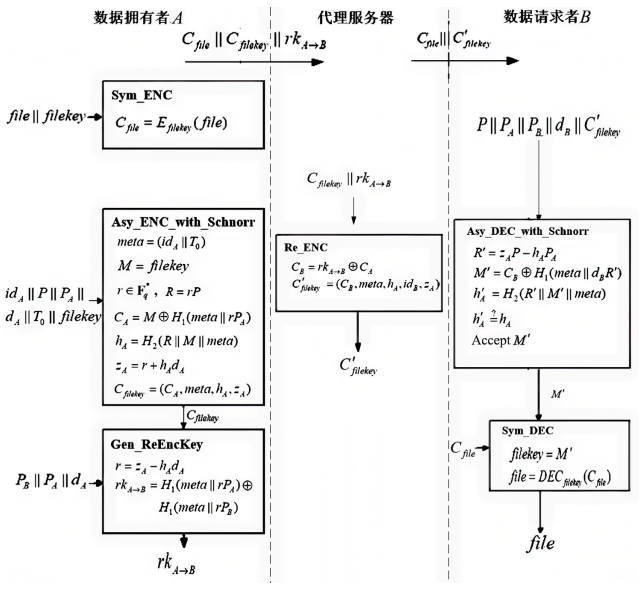



图 5 本文方案加解密流程图

## 5 可行性与安全性评估

### 5.1 可行性评估

1）正确性

通过验证$h_A$和$h'_A$是否相等来验证4.4节中的解密步骤是否正确，具体分析如下。

因为$h_A=H_2(R\|M\|meta)$，$h'_A=H_2(R'\|M'\|meta)$。其中，*meta*是相等的，故只要$R=R'$，即可通过验证$h_A=h'_A$说明$M=M'$。

由$P_A=d_AP$，$z_A=r+h_Ad_A$，$R=rP$，可以得出

$$R'=z_AP-h_AP_A=(r+h_Ad_A-h_Ad_A)P=rP=R \quad (1)$$

故可以通过验证$h_A=h'_A$说明$M=M'$。

下面验证在计算过程正确的情况下，有$M$=$M'$。因为$M'=C_B\oplus H_1(meta\|d_BR')=rk_{A\to B}\oplus C_A\oplus H_1(meta\|d_BR')=H_1(meta\|rP_A)\oplus H_1(meta\|rP_B)\oplus C_A\oplus H_1(meta\|d_BR')$。由公式（1）可得$d_BR'=d_BR=d_BrP=r(d_BP)=rP_B$。因此，$H_1(meta\|rP_B)=H_1(meta\|d_BR')$，且

$$M'=H_1(meta\|rP_A)\oplus C_A=H_1(meta\|rP_A)\oplus M\oplus H_1(meta\|rP_A)=M \quad (2)$$

由此验证了在计算过程正确的情况下，有$M=M'$，即本文方案是正确的。

2）机密性

在本文方案中，通过对称加密算法将明文转换成密文保存在代理服务器上，并通过非对称加密算法加密对称密钥。这就使得在密钥不被泄露的情况下，代理服务器和第三方无法将密文恢复成明文，数据机密性得到了保障。

3）完整性

本文方案将数据的哈希值在区块链中进行存储，这就使得数据请求者在通过重加密恢复出密钥进而恢复出明文时，可以通过区块链来验证恢复数据的完整性。

### 5.2 区块链安全性评估

本文方案中，用户通过元数据以及访问控制信息上链这个步骤，将访问控制以及数据完整性的安全保障交托给了区块链。攻击者想要对这个流程进行攻击，就必须能够篡改链上数据。

假设当前区块链的高度为$\alpha$，有攻击者试图对存在高度为$\beta$的区块上的区块信息进行篡改，进而修改访问控制信息，获得文件的下载权限。设该区块距离最新区块的高度差为$h$。因为区块链中的每个区块都保

存了上一个区块的哈希值，因此攻击者为了篡改这个区块，就必须重新计算并生成这个区块之后的所有区块，并得到全网的验证。攻击者的主要攻击对象是区块历史，即过去生成的区块，对新区块的产生不会有影响，因此新区块的产生难度不会增加。为了方便计算，假设没有新节点在攻击时加入进来。对于诚实节点（即未被攻击的节点）来说，假设它每秒生成一个区块的概率为 $w$，对于攻击者来说，假设这个概率为 $u$。根据定义，攻击者攻击的区块与最新区块的高度差为 $h$，则在每秒内，$h$ 可能有 3 种结果[16]：

1）结果 $X_1$。攻击者没有产生区块，诚实节点产生了区块，概率为 $P_1=w(1-u)$。

2）结果 $X_2$。攻击者产生了区块，诚实节点没有产生区块，概率为 $P_2=u(1-w)$。

3）结果 $X_3$。攻击者和诚实节点均产生或未产生区块，概率为 $P_3=1-P_1-P_2$[17]。

假设在 $t$ 秒内有 $t$ 个结果，$X_1$ 发生了 $n$ 次，$X_2$ 发生了 $m$ 次，$X_3$ 发生了 $t-m-n$ 次。每秒内 $h$ 的变化的概率分布符合多项分布。在 $t$ 秒内，攻击者产生区块的长度如果想追上诚实节点产生区块的长度，需要满足条件 $m>n+h$，即需要满足条件 $n\in[0,\frac{t-h-1}{2}]$, $m=n+h+k$ 并且 $1\leqslant k\leqslant t-2n-h$。满足该条件的概率为

$$P_h(t)=\sum_{n=0}^{(t-h-1)/2}\sum_{k=1}^{t-2n-h}\frac{t!}{n!(n+h+k)!(t-h-2n-k)!}\cdot P_1^n P_2^{n+h+k} P_3^{t-h-2n-k} \quad (3)$$

$P_h(t)$ 即攻击者成功篡改区块数据的概率。图 6 为攻击者算力为诚实节点的 50% 时，攻击者成功篡改区块数据的概率。图 7 为攻击者算力与诚实节点相同时，攻击者成功篡改区块数据的概率。其中，横坐标为时间 $t$，单位为诚实节点产生一个区块的平均时间；纵坐标为 $P_h(t)$；$h$ 代表了被篡改区块与最新区块的高度差。

从图 6 和图 7 可以看出，$h$ 越大，攻击者成功的概率越低。攻击者算力越大，其成功概率也随之增加。当攻击者的算力为诚实节点的 50% 时，其成功概率不超过 0.5%；当攻击者的算力与诚实节点相同时，其成功概率在 35% 左右。

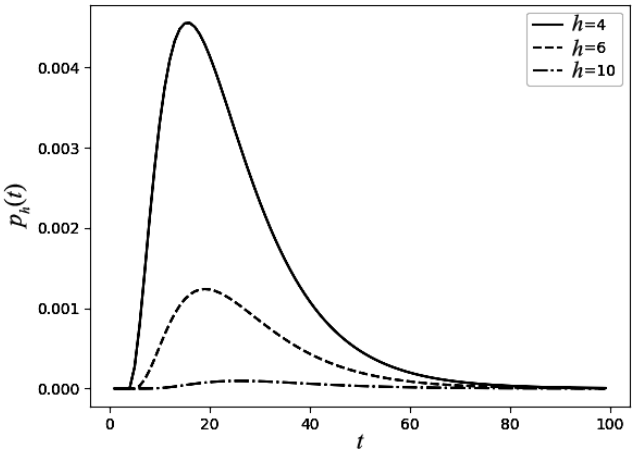



图 6 攻击者算力为诚实节点的 50% 时的攻击成功概率

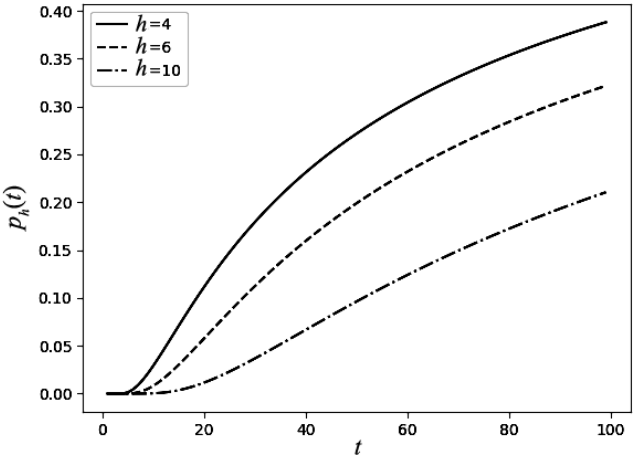



图 7 攻击者算力与诚实节点相同时的攻击成功概率

在实际数据共享应用中，数据的上传往往是大批量的，并且这个过程不断进行，这就导致攻击者想要篡改的区块与最新区块的高度差通常是成百上千个区块，因此攻击者很难依靠有限的算力达到篡改区块的目的。

## 6 方案实现

基于智能合约本文方案在医疗数据共享应用中得到了实现。前端主要为用户提供上传文件、搜索以及文件下载的功能。页面展示主要使用了 HTML、CSS 以及 JS，前端框架为 Vue，前端 UI 框架为 ElementUI，通过 RESTful 接口风格的 API 与代理服务器进行交互。后端主要为用户提供重加密、搜索以及存储数据的功能，使用 Spring Boot 框架以及 MongoDB 数据库。

## 7 结束语

区块链技术为数据的共享模式给出了一个全新的思路。本文基于区块链在数据共享方面的国内外研究现状，设计与实现了基于区块链技术的数据共享系统。该系统有如下优势：1）利用区块链保存访问控制信息以及日志信息；2）利用对称加密算法对数据进行加密；3）利用非对称加密算法对对称密钥进行加密；4）使用重加密技术保证数据拥有者完全控制数据的流向。该系统既保证了数据的安全性与完整性，又使得数据在共享过程中具有较高的可追溯性。本文方案已经成功应用在医疗数据共享项目中。未来工作将研究完全去中心化的区块链数据共享方案，以支持在没有第三方服务器的情况下实现数据的安全共享。（责编 马珂）

**参考文献：**

[1] TRUONG N B, SUN Kai, LEE G M, et al. GDPR-Compliant Personal Data Management: A Blockchain-based Solution[J]. IEEE Transactions on Information Forensics and Security, 2019, 15: 1746-1761.

[2] TANG Wen-an, WANG Hui. Scheme and Platform of Trusted Fund-raising and Donation Based on Smart Contract[J]. Journal of Computer Applications, 2020, 40(5): 1483-1487.
谭文安，王慧 . 基于智能合约的可信筹款捐助方案与平台 [J]. 计算机应用，2020，40（5）：1483-1487.

[3] BONAM J, REDDY A R, KALYANI G. Privacy Preserving in Association Rule Mining by Data Distortion using PSO[C]// Springer. ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India-Vol II, December 13 - 15, 2013, Visakhapatnam, India. Switzerland: Springer, Cham, 2014: 551-558.

[4] SHEELA M A, VIJAYALAKSHMI K. A Novel Privacy Preserving Decision Tree Induction[C]//IEEE. 2013 IEEE Conference on Information & Communication Technologies, April 11-12, 2013, Thuckalay, Tamil Nadu, India. NJ: IEEE, 2013: 1075-1079.

[5] ZHANG Hanlin, YU Jia, TIAN Chengliang, et al. Cloud Storage for Electronic Health Records Based on Secret Sharing with Verifiable Reconstruction Outsourcing[J]. IEEE Access, 2018, 6: 40713-40722.

[6] YUAN Yong, WANG Feiyuan. Blockchain: The State of the Art and Future Trends[J]. Acta Automatica Sinica, 2016, 42(4): 481-494.
袁勇，王飞跃 . 区块链技术发展现状与展望 [J]. 自动化学报，2016，42（4）：481-494.

[7] DUBOVITSKAYA A, XU Zhigang, RYU S, et al. Secure and Trustable Electronic Medical Records Sharing Using Blockchain[J]. Amia Annual Symposium Proceedings, 2017, 2017: 650-659.

[8] LIANG Xueping, ZHAO Juan, Shetty S, et al. Integrating Blockchain for Data Sharing and Collaboration in Mobile Healthcare Applications[C]// IEEE. 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), October 8-13, 2017, Montreal, QC, Canada. New York: IEEE, 2017: 1-5.

[9] GAO Mengjie, WANG Huaqun. Blockchain-based Searchable Medical Data Sharing Scheme[J]. Journal of Nanjing University of Posts and Telecommunications(Natural Science Edition), 2019, 39(6): 94-103.
高梦婕，王化群 . 基于区块链的可搜索医疗数据共享方案 [J]. 南京邮电大学学报（自然科学版），2019，39（6）：94-103.

[10] MEI Ying. The Utilizing Blockchain-based Method of the Secure Storage of Medical Records[J]. Journal of Jiangxi Normal University(Natural Science Edition), 2017, 41(5): 484-490.
梅颖 . 安全存储医疗记录的区块链方法研究 [J]. 江西师范大学学报（自然科学版），2017，41（5）：484-490.

[11] NAKAMOTO S. Bitcoin: Apeer-to-peer Electronic Cash System[EB/OL]. http://bitcoin.org/bitcoin.pdf, 2016-6-12.

[12] ANDROULAKI E, BARGER A, BORTNIKOV V, et al. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains[C]// ACM. The Thirteenth EuroSys Conference, April 23-26, 2018, Portugal Porto. New York: ACM, 2018: 1-15.

[13] BLAZE M, BLEUMER G, STRAUSS M. Divertible Protocols and Atomic Proxy Cryptography[C]//Springer. International Conference on the Theory and Applications of Cryptographic Techniques, May 31 - June 4, 1998, Espoo, Finland. Heidelberg: Springer-Berlin, 1998: 127-144.

[14] LANG Xun, WEI Lixian, WANG Xu-an, et al. Cryptographic Access Control Scheme for Cloud Storage Based on Proxy Re-encryption[J]. Journal of Computer Applications, 2014, 34(3): 724-727, 741.
郎讯，魏立线，王绪安，等 . 基于代理重加密的云存储密文访问控制方案 [J]. 计算机应用，2014，34（3）：724-727，741.

[15] NOR R M, RAHMAN M M H, RAHMAN T, et al. Blockchain Sadaqa Mechanism for Disaster Aid Crowd Funding[EB/OL]. http://irep.iium.edu.my/61375/, 2020-5-25.

[16] TAN Haibo, ZHOU Tong, ZHAO He, et al. Archival Data Protection and Sharing Method Based on Blockchain[J]. Journal of Software, 2019, 30(9): 2620-2635.
谭海波，周桐，赵赫，等 . 基于区块链的档案数据保护与共享方法 [J]. 软件学报，2019，30（9）：2620-2635.

[17] SHENG Nianzu, LI Fang, LI Xiaofeng, et al. Data Capitalization Method Based on Blockchain Smart Contract for Internet of Things[J]. Journal of Zhejiang University(Engineering Science), 2018, 52(11): 2150-2158.
盛念祖，李芳，李晓风，等 . 基于区块链智能合约的物联网数据资产化方法 [J]. 浙江大学学报（工学版），2018，52（11）：2150-2158.