计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.20210587
59(12):2750-2759,2022

可控、可追责的敏感数据共享方案

张正昊　李　勇　张振江
(北京交通大学电子信息工程学院　北京　100044)
(19120172@bjtu.edu.cn)

Controllable and Accountable Sensitive Data Sharing Scheme

Zhang Zhenghao, Li Yong, and Zhang Zhenjiang
(*School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing* 100044)

Abstract　In the era of big data, the sharing of massive amounts of data is a prerequisite for fully mining the value of data. Some of these data are sensitive data involving user privacy, and special attention should be paid to the data sharing process. However, traditional data sharing methods have defects such as unclear data flow and difficulty in accountability. To solve these problems, a blockchain-based sensitive data controllable sharing solution that supports regulation is proposed. By using the dynamic accumulator technology to achieve access control of sensitive data, the data owner can flexibly grant or revoke the access rights of other participants to the data and realize the controllability of the data by the data owner. A regulator is set up to check the data request process. The regulator will issue a regulatory certificate to the data requester after the check is approved. Only the data requester who has the regulatory certificate and is authorized by the data owner can get the data. To protect the privacy of the data requester, unrelated third parties cannot obtain the identity information of the data requester by using strong designated verifier signature (SDVS). Blockchain technology is used to record data requests and responses. The record can only be read by the regulator, which realizes the regulation of the whole data sharing process. The security analysis proves that the scheme satisfies the privacy of the data requester, the controllability of the data owner, and accountability. The simulation experiment proves the feasibility of the scheme.

Key words　data sharing; controllability; privacy protection; accountability; blockchain

摘　要　在大数据时代下,海量数据之间的共享是充分挖掘数据价值的前提.对涉及用户隐私的敏感数据,需要对其共享过程特别关注,而传统的数据共享方式存在数据流向不明确、难以追责等缺陷.针对这些问题,基于区块链提出了一种支持监管的敏感数据可控共享方案.通过使用动态累加器技术实现敏感数据的访问控制,数据拥有方可以灵活地授予或者撤销其他参与方对数据的访问权限,实现数据拥有方对数据的可控性.设置监管方对数据请求过程进行审核,监管方将为通过审核的数据请求方颁发监管凭证,只有拥有监管凭证且获得数据拥有方授权的数据请求方才能获得数据.为保护数据请求方的隐私,通过强指定验证者签名技术,使无关第三方无法获得数据请求方的身份信息.使用区块链技术记录数据的请求和响应情况,该记录只有监管方可以读取,从而实现了监管方对数据共享全流程的可监管性.安全性分析表明,方案满足数据请求方隐私性、数据拥有方可控性、可监管性,仿真实验验证了方案的可行性.

收稿日期:2021-06-10;**修回日期**:2021-11-05
基金项目:国家重点研发计划项目(2018YFC0832300,2018YFC0832303)
This work was supported by the National Key Research and Development Program of China (2018YFC0832300, 2018YFC0832303).
通信作者:李勇(liyong@bjtu.edu.cn)

关键词 数据共享;可控;隐私保护;可追责;区块链

中图法分类号 TP391

在当今大数据时代,人们的日常生活中会产生大量数据,这些数据代表着一个人的日常习惯、行为动态、生活轨迹等,是极其重要的隐私数据.数据产生的源头可能多种多样,不同的公司、企业、部门之间都持有用户数据.在大数据背景下,这些蕴涵着巨大价值的数据必然走向共享、开放[1],即数据只有流动起来才能发挥其最大效能,进一步挖掘出数据背后的隐藏信息,为企业和个人提供更好的服务质量[2-3].如果不同数据只存放于独立的机构内部,就会形成"数据孤岛",数据得不到充分利用,造成数据浪费.

然而,数据共享过程中存在的突出问题是,数据拥有方往往难以控制数据的流向,数据是否能得到合法使用也难以保证.例如 2020 年 11 月,国内某快递公司的 40 余万条客户个人信息被内部用户获取之后倒卖,被泄露的信息包括客户的收发件地址、姓名、电话等,这是一起严重的数据泄露事件.因此有必要研究隐私数据在各方之间共享时的安全性,实现数据的可控共享.另外,许多信息泄露事件都是内部人员所为,因此数据共享也需要设置监管方对数据共享进行监管,确保所有的数据获取请求和数据响应合规合法,最大程度避免数据泄露情况的发生.

1 相关工作

针对敏感数据共享过程中的痛点和难点,文献[4]基于大数据平台提出了一种安全的敏感数据共享框架.该框架从平台中数据的生命周期出发,分别从数据提交、存储、使用和销毁 4 个方面考虑数据的安全共享问题.文献[4]的作者设计了一种异构代理重加密方案,支持从身份基加密[5]到公钥加密[6]之间的转换.另外为了避免隐私信息的泄露,数据请求者将在基于虚拟机监视器(virtual machine monitor,VMM)[7]的私有进程中对数据进行下载操作,所有的密钥将保存在私有进程的内存中.框架为所有隐私数据设置了一种基于租约的机制,在租约到期后,明文和密钥将从云中彻底销毁.但是该框架对请求者的隐私问题考虑不充分,任何发生在平台上的数据共享行为都会被无关第三方得知,并且其访问控制功能的实现主要依赖于数据拥有方是否为请求方生成了重加密密钥,对数据请求方的权限控制相对较弱.

2008 年,Nakamoto[8]提出比特币概念,区块链开始引起关注.由于区块链具有分布式、防篡改等特性,一些学者开始使用区块链进行访问控制,以设计可控的敏感数据共享方案.文献[9]针对云环境下的电子病历共享场景,设计了一个轻量可扩展的联盟链,所有的用户实体都将经过授权之后才能进入共享系统,并且所有的共享行为都会被实时记录.该方案利用区块链分布式与防篡改等特性,构造了比较强的访问控制方案.文献[10]研究了医疗数据在无信任环境下的数据共享问题,基于区块链技术提出了医疗数据管理系统 MeDShare.在该系统内,对数据的操作行为都将以防篡改的方式记录下来,并且通过访问控制机制进行用户权限的管理.但是其访问控制通过访问控制表实现,验证用户权限时开销比较大.文献[11]基于联盟链并使用可搜索加密技术[12]和秘密分享技术[13],设计了一个基于区块链的医疗数据共享方案.在该方案中,希望得到共享数据的多个用户将获得搜索凭证的密钥份额,并提供给联盟链进行验证,验证通过后会得到搜索凭证,进而向数据服务器请求相应的密文,密文会被提交给智能合约进行解密,从而得到数据明文.此方案考虑到了云存储服务器并非完全可信的问题,并使用秘密分享技术来支持多个用户的共享,但该方案的交互过程相对复杂,联盟链、云服务器、数据拥有方和请求方之间都有交互,系统耦合性比较高.

随着使用区块链进行访问控制的相关研究不断深入,学者们开始注意到数据共享过程中的监管问题.文献[14]使用代理重加密技术[15]进行数据共享,同时保证数据拥有方对数据有一定程度的可控性.当用户有下载文件的权限时,第三方代理就会将数据密文和密钥的重加密密文发给用户,用户用自己的私钥解密出密钥,从而获得数据明文.该方案使用区块链对用户权限进行验证,并且记录请求内容,实现一定程度的监管性.文献[16]从时间维度的访问控制出发,结合属性基加密[17]和区块链,设计了医疗数据的共享方案.数据拥有方在给其他用户授权时,会部署一个限时智能合约,只有用户同时满足访问控制结构和时间范围才能获得数据.该方案虽然设置了监管中心对用户的身份进行管理,但是监管中心无法监管共享请求,也不知道是否有非法共享行为发生.另外,从时间维度对数据共享进行限制

的灵活性不高,只要用户符合访问控制结构,那么在相应的时间段内用户可以随意获得数据,拥有方无法撤回授权.

从上述分析中可知,敏感数据共享过程仍存在待解决的问题:1)数据拥有方无法灵活地进行权限更新和撤销.2)数据共享本身缺乏监管性,有方案确实设置了监管中心,但监管性比较弱.鉴于敏感数据的重要性,必须有一个强监管方来保证数据共享过程的合法合规.3)共享过程中的用户身份隐私问题也需要考虑,共享本身是双方之间的交互过程,不相关的第三方不能获取到参与方的身份信息.针对这3个问题,本文以区块链技术为背景,提出了一个可控、可追责的敏感数据共享方案,旨在提高数据共享的隐私性、可控性和可监管性.该方案可以确保数据拥有方对数据的控制权,设置监管方以监督共享全过程,避免非法行为的出现.并且数据共享的行为将记录在区块链上,利用区块链不可篡改等特性,监管方可以追溯数据的使用过程,在非法行为出现时找到相关参与方进行问责.引入强指定验证者签名技术,保护了数据请求方的身份隐私.在访问控制方面使用动态累加器,使得数据拥有方可以灵活地对授权用户进行管理,包括权限的更新和撤销等,增强了拥有方对自己数据的可控程度.

2 相关假设

2.1 强 RSA 假设

设 $n=pq$ 为 RSA 模,$c\in\mathbb{Z}_n^*$,强 RSA 假设是指在不知道 n 的分解的前提下,计算 $a,b\in\mathbb{Z}_n^*$,$b\geqslant 2$,使得 $a^b=c \bmod n$ 是困难的.

2.2 离散对数(discrete logarithm, DL)假设

对于给定的群 $\mathbb{G}$,设 $y\in\mathbb{G}$,DL 假设是指对于任何运行在多项式时间 t 内的算法 $\mathcal{A}$,求解出 $x\in\mathbb{Z}_q$,使得 $y=g^x \bmod q$ 的概率是可以忽略的.

2.3 计算性 Diffie-Hellman(computational Diffie-Hellman, CDH)假设

给定生成元为 g 的 q 阶循环群 $\mathbb{G}$,对于 $x,y\in_R\mathbb{Z}_q^*$,已知(g,g^x,g^y),CDH 假设是指对于任何运行在多项式时间 t 内的算法 $\mathcal{A}$,正确计算 g^{xy} 的概率是可以忽略的.

2.4 判定性 Diffie-Hellman(decisional Diffie-Hellman, DDH)假设[18]

给定生成元为 g 的 q 阶循环群 $\mathbb{G}$,对于 $x,y,z\in_R\mathbb{Z}_q^*$,已知(g,g^x,g^y,g^z),DDH 假设是指对于任何运行在多项式时间 t 内的算法 $\mathcal{A}$,正确判断是否有 $z=xy \bmod q$ 成立的概率是可以忽略的.

2.5 间隙性 Diffie-Hellman(gap Diffie-Hellman, GDH)假设[19]

给定生成元为 g 的 q 阶循环群 $\mathbb{G}$,对于 $x,y\in_R\mathbb{Z}_q^*$,已知(g,g^x,g^y),GDH 假设是指借助于 DDH 预言机的前提下,对于任何运行在多项式时间 t 内的算法 $\mathcal{A}$,正确计算 g^{xy} 的概率是可以忽略的.

3 方案定义

本文方案包含的算法有 10 个:

1) $Setup(1^k)\rightarrow params$ 为系统初始化算法.以安全性参数 1^k 为输入,输出公共参数 $params$.

2) $KeyGen(params)\rightarrow(sk_A,pk_A,sk_B,pk_B,sk_R,pk_R)$为密钥生成算法.输入公共参数 $params$,输出数据拥有方密钥对(sk_A,pk_A)、数据请求方密钥对(sk_B,pk_B)和监管方密钥对(sk_R,pk_R).

3) $AccInit(params)\rightarrow(Acc,C)$为累加器初始化算法,由数据拥有方执行.为属于自己的数据设置一个访问控制结构,输出累加器 Acc 和授权集合 C.

4) $ShareRequest(params,sk_B,pk_B,pk_A)\rightarrow req_{Share}$为共享请求算法,由数据请求方执行.输入公共参数 $params$、数据请求方的密钥对(sk_B,pk_B)和数据拥有方的公钥 pk_A,输出共享请求 req_{Share}.

5) $Authorization(params,sk_A,req_{Share},Acc,C)\rightarrow(0\mid witness_B,witness)$为授权算法,由数据拥有方执行.输入公共参数 $params$、数据拥有方的私钥 sk_A、共享请求 req_{Share}、当前累加器 Acc 和授权集合 C,数据拥有方根据共享请求判断是否授权,如果不授权则输出 0,如果授权则输出数据请求方证据 $witness_B$ 和其他授权用户的新证据 $witness$.

6) $RegulateRequest(params,msg,pk_A,sk_B,pk_B,pk_R,witness_B)\rightarrow req_{Regulate}$为监管请求算法,由数据请求方执行.输入公共参数 $params$、数据拥有方公钥 pk_A、请求方自己的密钥对 sk_B,pk_B、监管方公钥 pk_R、请求内容 msg 和授权凭证 $witness_B$,请求方为监管方生成一个强指定验证者签名,并附带其他附加信息后输出 $req_{Regulate}$.

7) $RegulateVerify(params,pk_B,sk_R,pk_R,req_{Regulate})\rightarrow(0\mid\sigma_R)$为监管验证算法,由监管方执行.输入公共参数 $params$、请求方公钥 pk_B、监管请求和监管方密钥对 sk_R,pk_R,监管方验证信息的正确性和合法性,如果认为该请求不合法,则输出 0,如

果认为请求可以执行,则输出签名 σ_R,并将请求上传到区块链中.

8) $DataResponse(params, pk_B, pk_R, \sigma_R, w_B, msg) \rightarrow (0|res_{Data})$ 为数据响应算法,由数据服务器执行.输入公共参数 $params$、数据请求方发来的信息 pk_B,σ_R,w_B 和监管方公钥 pk_R,数据服务器验证所有凭证是否成立,如果验证通过,则返回响应的数据,输出 res_{Data},并将响应信息上传到区块链,否则输出 0.

9) $Revoke(params, Acc, C) \rightarrow (Acc_{new}, witness)$ 为权限撤销算法,由数据拥有方执行.输入公共参数 $params$ 和当前累加器 Acc,数据拥有方对在权限集合 C 中的某个用户进行权限撤销,输出新累加器 Acc_{new} 和其他还存在于授权集合中的用户的证据 $witness$.

10) $RegulateSim(params, pk_B, sk_R, pk_R, req_{Regulate}) \rightarrow \sigma_{Sim}$ 为副本模拟算法,由监管方在有需要时执行.为了保护数据请求方的隐私,输入公共参数 $params$、请求方公钥 pk_B、监管请求 $req_{Regulate}$ 和监管方密钥对 sk_R,pk_R,输出签名副本 σ_{Sim}.

4 安全模型

在本文方案中,安全性需求描述为 4 方面:

1) 数据拥有方可以控制自己数据的流向,即可以灵活地将数据访问权限授权给其他用户,也可以撤销对某个用户的授权;

2) 数据请求方在获得数据拥有方的授权之后,需要进一步获得监管方的授权,在与监管方通信的过程中,数据请求方的隐私得到保护;

3) 监管方可以对数据的请求、响应进行全流程的把控,一旦发现非法行为的出现,监管方将追踪到相应的共享参与方;

4) 在数据请求方与监管方交互的过程中,监管方不能破坏请求方的隐私,也不能冒充数据请求方,向数据拥有方非法请求数据.

本文假设数据服务器是可信的,且各参与方之间与数据服务器之间的信道是安全的.下面给出本文方案的安全性定义.

定义 1. 数据请求方匿名性.对于一个挑战者 $\mathcal{C}$ 和敌手 $\mathcal{D}$ 之间的游戏 $Game_{anony}$,如果不存在任何概率多项式时间敌手能以不可忽略的优势赢得该游戏,则该方案具有数据请求方匿名性.$Game_{anony}$ 定义为:

1) 对于签名者 S_0,S_1 和验证者 V,挑战者 $\mathcal{C}$ 为其生成密钥对——(sk_{S_0}, pk_{S_0}),(sk_{S_1}, pk_{S_1}),(sk_V, pk_V),并将公钥 $(pk_{S_0}, pk_{S_1}, pk_V)$ 发送给敌手 $\mathcal{D}$.

2) 敌手 $\mathcal{D}$ 在多项式时间内向下列预言机发出查询.

$\mathcal{O}_{sign}$.输入待签名消息 M,pk_{S_d} $(d \in \{0,1\})$,pk_V,输出对消息 M 的签名 σ.

$\mathcal{O}_{sim}$.输入待签名消息 M,pk_{S_d} $(d \in \{0,1\})$,pk_V,输出对消息 M 的签名 σ.

$\mathcal{O}_{ver}$.输入待签名消息 M,σ,pk_{S_d} $(d \in \{0,1\})$,pk_V,当 σ 有效时输出 1,否则输出 0.

3) 敌手 $\mathcal{D}$ 输出消息 M^*,挑战者 $\mathcal{C}$ 抛出硬币选择一个 $b \in \{0,1\}$,运行指定验证者签名算法输出 $\sigma^* = Sign(sk_{S_b}, pk_{S_b}, pk_V, M^*)$,将 σ^* 发送给敌手 $\mathcal{D}$.

4) 敌手 $\mathcal{D}$ 向 $\mathcal{O}_{sign}$ 和 $\mathcal{O}_{sim}$ 发出如步骤 2) 中的查询,然后输出 b',当 $b'=b$ 时,敌手 $\mathcal{D}$ 赢得这个游戏.

定义 2. 数据拥有方可控性.如果满足以下条件,则数据拥有方可以控制自己数据的流向,即方案具备可控性:

$$\begin{aligned}&Pr[Setup(1^k) \rightarrow params, KeyGen(params) \rightarrow \\ &(sk_A, pk_A, sk_B, pk_B, sk_R, pk_R), \\ &AccInit(params) \rightarrow (Acc, C), \\ &ShareRequest(params, sk_B, pk_B, pk_A) \rightarrow \\ &req_{Share}: Authorization(params, sk_A, \\ &req_{Share}, Acc, C) \rightarrow 0] \geqslant 1 - v(\lambda),\end{aligned}$$

或者

$$\begin{aligned}&Pr[Setup(1^k) \rightarrow params, KeyGen(params) \rightarrow \\ &(sk_A, pk_A, sk_B, pk_B, sk_R, pk_R), \\ &AccInit(params) \rightarrow (Acc, C), \\ &ShareRequest(params, sk_B, pk_B, pk_A) \rightarrow req_{Share}: \\ &Authorization(params, sk_A, req_{Share}, Acc, C) \rightarrow \\ &(witness_B, witness)] \geqslant 1 - v(\lambda),\end{aligned}$$

其中,$v(\lambda)$ 为可忽略函数,即数据拥有方不给数据请求方授权的概率接近于 1,或数据拥有方向数据请求方授权,并为其他授权用户更新凭证的概率接近于 1.

定义 3. 可监管性.如果满足以下条件,则监管方可以审核系统中的共享请求和响应,即方案具有可监管性:

$$\begin{aligned}&Pr[Setup(1^k) \rightarrow params, KeyGen(params) \rightarrow \\ &(sk_A, pk_A, sk_B, pk_B, sk_R, pk_R), \\ &AccInit(params) \rightarrow (Acc, C), \\ &ShareRequest(params, sk_B, pk_B, pk_A) \rightarrow req_{Share},\end{aligned}$$

$$Authorization(params, sk_A, req_{Share}, Acc, C) \rightarrow (witness_B, witness), RegulateRequest(params, msg, pk_A, sk_B, pk_B, pk_R, witness_B) \rightarrow req_{Regulate}: RegulateVerify(params, pk_B, sk_R, pk_R, req_{Regulate}) \rightarrow 0] \geqslant 1-v(\lambda),$$

或者

$$Pr[Setup(1^k) \rightarrow params, KeyGen(params) \rightarrow (sk_A, pk_A, sk_B, pk_B, sk_R, pk_R), AccInit(params) \rightarrow (Acc, C), ShareRequest(params, sk_B, pk_B, pk_A) \rightarrow req_{Share}, Authorization(params, sk_A, req_{Share}, Acc, C) \rightarrow (witness_B, witness), RegulateRequest(params, msg, pk_A, sk_B, pk_B, pk_R, witness_B) \rightarrow req_{Regulate}: RegulateVerify(params, pk_B, sk_R, pk_R, req_{Regulate}) \rightarrow \sigma_R] \geqslant 1-v(\lambda),$$

其中,$v(\lambda)$为可忽略函数,即当数据请求方的请求有问题时,监管方否定该请求的概率接近于 1,或者当数据请求方的请求正确时,监管方同意该请求并生成监管凭证的概率接近于 1.

5 方案构造

通过引入动态累加器方案和强指定验证者签名方案,基于区块链技术设计了一种适用于敏感数据共享的方案.在该方案中,各数据共享方拥有对自己数据的控制权,在将数据上传时就设置好访问控制结构,只有满足访问控制结构的数据请求方才有机会获得数据.监管方负责验证每一次数据的请求过程,数据的请求和响应情况将被上传到区块链中,这些记录只有监管方可以读取,并且由于区块链具有不可篡改等特性,监管方可以在出现非法行为时追溯数据的共享全过程,定位相关责任方,进而采取必要措施.

在本节中,假设有 2 个参与方 A 和 B,A 将数据上传到数据服务器中,B 请求获得 A 的某些数据,共享流程如图 1 所示.其中,本文假设数据服务器与各参与方之间的信道是安全可信的,其余信道不可靠,需要借助如 SSL/TLS 等技术进行传输.本文方案约定 $Enc_{pk}(m)=c$,$Dec_{sk}(c)=m$ 为非对称加/解密算法,$Sig_{sk}(m)=\sigma$,$Ver_{pk}(\sigma)=0$ 或 1 为数字签名和签名验证算法,具体计算过程不再赘述.方案构造 9 个算法:

1) $Setup(1^k) \rightarrow params$.输入安全参数 1^k,得到一个长度为 k 的随机数 N,使得 $N=pq$,其中 $p=2p'+1$,$q=2q'+1$,p 和 q 的长度相等,并且 p,q,p',q'都是素数,从 N 的二次剩余循环群 QR_N 中选择一个随机数 u.接着选取一个乘法群$\mathbb{G}$,阶为 m,生成元为 g,$H:\{0,1\}\times\mathbb{G}^3\rightarrow\mathbb{Z}_m$为一个抗碰撞的 Hash 函数.输出公共参数 $params=(N,u,\mathbb{G},m,g,H)$.

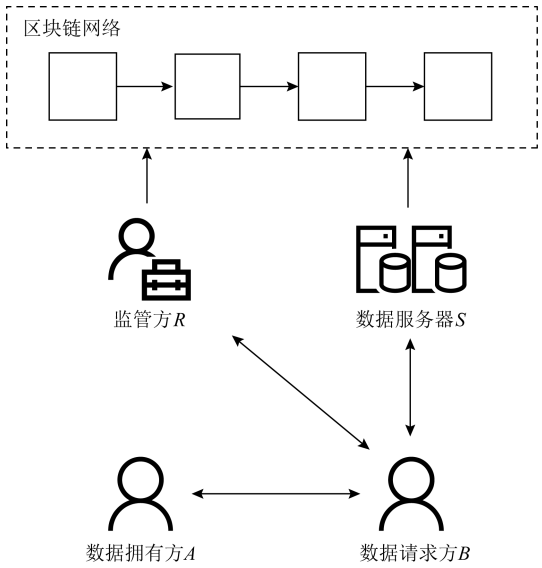

Fig. 1 Data requester B requests shared from data owner A

图 1 数据请求方 B 向数据拥有方 A 请求共享数据

2) $KeyGen(params) \rightarrow (sk_A, pk_A, sk_B, sk_R, pk_R)$.参与数据共享的双方 A 和 B 各自生成一个随机数 $x_1, x_2 \in \mathbb{Z}_m$,令 $sk_A=x_1$,$sk_B=x_2$,$pk_A=g^{sk_A} \bmod m$,$pk_B=g^{sk_B} \bmod m$,A 的密钥对为(sk_A, pk_A),B 的密钥对为(sk_B, pk_B).监管方 R 选取随机数$x\in\mathbb{Z}_m$,监管私钥 $sk_R=x$,$pk_R=g^{sk_R} \bmod m$,得到监管密钥对(sk_R, pk_R).A,B,R 分别保管各自的私钥,并公开公钥.

3) $AccInit(params) \rightarrow (Acc, C)$.$A$ 在本地维护一个授权集合 $C=\{c_1, c_2, \cdots, c_n\}$,其中$\{c_1, c_2, \cdots, c_n\}$为 n 个授权用户的公钥,计算累加器 $Acc=u^{c_1 c_2 \cdots c_n} \bmod N$,输出 Acc.

4) $ShareRequest(params, sk_B, pk_B, pk_A) \rightarrow req_{Share}$.设 B 想要获取的内容为 msg,当前系统时间为 T,计算 $r_1=Enc_{pk_A}(pk_B, msg, T)$,$r_2=Sig_{sk_B}(pk_B, msg, T)$,输出共享请求 $req_{Share}=\{r_1, r_2\}$发送给 A.

5) $Authorization(params, sk_A, req_{Share}, Acc, C) \rightarrow (0 | witness_B, witness)$.当 A 收到 req_{Share}后,计

算 $Dec_{sk_A}(r_1)=\{pk_B,msg,T\}$,通过 $Ver_{pk_B}(r_2)$验证签名的正确性.若验证不通过则输出 0,若验证通过,则考虑是否授予 B 权限,如果决定授权,则将累加器更新为 $Acc_{new}=Acc^{pk_B} \bmod N$,将 B 的公钥 pk_B 加入授权集合 C 中形成新的授权集合 $\widetilde{C}$.为了证明 $c_B=pk_B\in\widetilde{C}$,计算证据 $w_B=u^{\prod_{\substack{i=1\\c_i\neq c}}^{n}c_i} \bmod N$,$c_i=\widetilde{C}$,生成密文 $Enc_{pk_B}(w_B)$,生成签名 $Sig_{sk_A}(w_B)$,令 $witness_B=\{Enc_{pk_B}(w_B),Sig_{sk_A}(w_B)\}$.对于原先就在授权集合中证据为 w 的用户,A 为其计算 $w_{new}=w^{c_B} \bmod N$ 作为新证据,生成 $Enc_{pk}(w_{new})$ 和 $Sig_{sk_A}(w_{new})$,令 $witness=\{Enc_{pk}(w_{new}),Sig_{sk_A}(w_{new})\}$.公开累加器 Acc_{new},将 $witness_B$ 发送给 B,向其他用户发送 $witness$.

6) $RegulateRequest(params,msg,pk_A,sk_B,pk_B,pk_R,witness_B)\rightarrow req_{Regulate}$.$B$ 收到 $witness_B$ 后,计算 $Dec_{sk_B}(Enc_{pk_B}(w_B))$,$Ver_{pk_A}(Sig_{sk_A}(w_B))$验证签名的正确性.若验证失败,则算法终止,回到 $ShareRequest$;若验证成功,则生成强指定验证者签名(strong designated verifier signature, SDVS)[20].为了避免恶意监管方收到 w_B 后,冒充数据请求方 B 向数据服务器 S 请求数据,B 不能将 w_B 直接发送给 R,而是应该发送随机化后的 w_B,过程有 4 个步骤.

① 随机生成 $r,z,t,r'\in\mathbb{Z}_m$;

② 计算 $R_1=g^{r'}$,$R_2=g^{z}\cdot pk_R^{-t}$,$K=pk_R^{sk_B}$,$\widetilde{w}_B=w_B^{r}$;

③ 令 $l=H(msg,\widetilde{w}_B,K,R_1,R_2)$,计算 $t'=l-t$,$z'=r'+t'\cdot sk_B$;

④ 输出签名 $\sigma_{DVS}=(t,z,t',z')$.

令 $y_1=Enc_{pk_R}(msg,\widetilde{w}_B)$,$y_2=\sigma_{DVS}$,$B$ 将 $req_{Regulate}=(y_1,y_2)$发送给监管方.

7) $RegulateVerify(params,pk_B,sk_R,pk_R,req_{Regulate})\rightarrow(0|\sigma_R)$.监管方 R 收到 $req_{Regulate}=(y_1,y_2)$之后计算 $Dec_{sk_R}(y_1)=(msg,\widetilde{w}_B)$,然后对 σ_{DVS} 进行验证,验证过程为:

① 将 σ_{DVS}解析为(t,z,t',z');

② 计算 $R_1=g^{z'}\cdot pk_B^{-t'}$,$R_2=g^{z}\cdot pk_R^{-t}$,$K=pk_B^{sk_R}$;

③ 验证 $t+t'?=H(msg,\widetilde{w}_B,K,R_1,R_2)$.

若不相等则算法终止,若相等则验证通过,然后 R 将验证累加器证据的正确性.由于 R 收到的是 $\widetilde{w}_B$ 而非原本的累加器证据,无法直接进行验证,因此 B 需要通过交互式零知识证明过程向 R 证明:自己持有随机数 r,可以使得 $e(\widetilde{w}_B,g^{pk_B})=e(Acc,g^{r})$.从而使得 B 可以在不暴露 w_B 的情况下,证明 w_B 的有效性.具体过程如算法 1 所示.当算法 1 输出 1 时,R 进行如下计算:对请求方 B 的公钥 pk_B 生成签名 $\sigma_R=Sig_{sk_R}(pk_B)$,将该签名返回给 B.

算法 1. 关于累加器证据的零知识证明.

① B 和 R 分别计算 $v_1=e(\widetilde{w}_B,g^{pk_B})$,$v_2=e(Acc,g)$;

② B 生成随机数 $s\in\mathbb{Z}_m$,计算 $w=v_2^{s}=e(Acc,g)^{s}$,并将 w 发送给 R;

③ R 生成随机数 $\eta\in\mathbb{Z}_m$,并将 η 发送给 B;

④ B 计算 $\mu=s+\eta r$,并将 μ 发给 R;

⑤ R 验证 $v_2^{\mu}?=w\cdot v_1^{\eta}$,若等于,则输出 1,否则输出 0.

8) $DataResponse(params,pk_B,pk_R,\sigma_R,w_B,msg)\rightarrow(0|res_{Data})$.$B$ 构造 $dataReq=(pk_B,\sigma_R,w_B,msg)$发送给数据存储服务器 S,S 将计算 $Ver_{pk_R}(\sigma_R)$,当验证通过时计算 $Acc'=w_B^{pk_B} \bmod N$,如果 Acc'等于当前的累加器 Acc,则最终验证通过,返回 B 所请求的数据.服务器构建 $res_{Data}=(msg',pk_B,T')$,其中 msg'为服务器返回的内容,T'为当前的系统时间,生成 $Enc_{pk_R}(res_{Data})$后上传到区块链.

9) $Revoke(params,Acc,C)\rightarrow(Acc_{new},witness)$.当 A 想要撤销 B 的权限,即从授权集合 $C=\{c_1,c_2,\cdots,c_n\}$中删掉 $c_B=pk_B$ 时,A 更新累加器为 $Acc_{new}=Acc^{c_B^{-1} \bmod (p-1)(q-1)} \bmod N$,对于尚存在于授权集合中证据为 w 的参与方,更新其证据为 $w_{new}=w^{b}Acc_{new}^{a} \bmod N$,其中 a,b 为 2 个整数,满足 $av+bc_B=1$,v 为参与方的公钥,然后 A 公开 Acc_{new},向各参与方发送 $witness=\{Enc_{pk}(w_{new}),Sig_{sk_A}(m)\}$.

6 方案分析

6.1 安全性分析

本节分析本文方案的安全性.

定理 1. 如果 Hash 函数 H 为随机预言机,且 GDH 假设在群G上成立,则该方案满足监管过程中的数据请求方隐私性.

证明. 设存在一个敌手 $\mathcal{D}$,以$(1/2+\delta)$的概率攻破了监管过程中数据请求方的隐私性,即攻破了强指定验证者签名 SDVS 的隐私性,下面使用 $\mathcal{D}$ 作为子程序,构建算法 $\mathcal{E}$ 解决 GDH 假设.

步骤 1. $\mathcal{E}$ 初始化 GDH 参数($\mathbb{G},g,q,g^u,g^w$)和 DDH 预言机,$\mathcal{E}$ 选择一个随机数 $u'\in\mathbb{Z}_q$,令 $pk_{S_0}=g^u$,$pk_{S_1}=g^ug^{u'}$,$pk_V=g^w$,将(pk_{S_0},pk_{S_1},pk_V)发给 $\mathcal{D}$,$\mathcal{E}$ 生成初始为空的 2 个表格 HT 和 ST.

步骤 2. $\mathcal{D}$ 向 $\mathcal{E}$ 进行如下查询,其中,本文假设 $\mathcal{D}$ 不进行重复查询.

1) Hash 查询.输入(M,K,R_1,R_2),如果 $\mathcal{E}$ 发现 HT 中已经存在对应的记录((M,K,R_1,R_2),l),则直接返回 c,否则,将(g,pk_{S_0},pk_V,K)和(g,pk_{S_1},pk_V,K)输入 DDH 预言机,分别得到 b_0 和 b_1.如果 $b_0=1$,则输出 K 并中断;如果 $b_1=1$,则输出 $K/(g^w)^{u'}$ 并中断.如果 $b_0=b_1=0$,则生成随机数 $\tau\in\mathbb{Z}_q$,将 τ 返回给 $\mathcal{D}$ 并在 HT 中增加((M,K,R_1,R_2),l).

2) 签名查询.输入比特 d 和消息 M,$\mathcal{E}$ 随机生成 $t'_M,t_M,z'_M,z_M\in\mathbb{Z}_q$,计算 $R_{M,1}=g^{z'_M}pk_{S_d}^{-t'_M}$,$R_{M,2}=g^{z_M}\cdot pk_V^{-t_M}$,$l_M=t'_M+t_M$,如果 l_M 重复,则重新生成随机数进行上述计算,否则将(($M,\perp,R_{M,1},R_{M,2}$),l_M)存储在 HT 中,返回 $\sigma_M=(t'_M,t_M,z'_M,z_M)$,将($d,M,\sigma_M$)存储在 ST 中.

3) 验证查询.输入比特 d、消息 M 和某个 $\sigma_M=(t'_M,t_M,z'_M,z_M)$,$\mathcal{E}$ 检索 ST,如果(d,M,σ_M)已经存在,则返回 1,否则返回 0.

步骤 3. $\mathcal{D}$ 输入消息 M^*,$\mathcal{E}$ 随机抛出硬币选择一个 $b\in\{0,1\}$,随机选择 $t'_{M^*},t_{M^*},z'_{M^*},z_{M^*}\in\mathbb{Z}_q$,计算 $R_{M^*,1}=g^{z'_{M^*}}pk_{S_d}^{-t'_{M^*}}$,$R_{M^*,2}=g^{z^*_v}pk_V^{-t_{M^*}}$,$l_{M^*}=t'_{M^*}+t_{M^*}$,同样,如果 l_{M^*} 重复,则重新生成随机数进行上述计算,否则将(($M^*,\perp,R_{M^*,1},R_{M^*,2}$),$l_{M^*}$)存储在 HT 中,返回 $\sigma_{M^*}=(t'_{M^*},t_{M^*},z'_{M^*},z_{M^*})$,将($b,M^*,\sigma_{M^*}$)存储在 ST 中.

步骤 4. $\mathcal{D}$ 输出 b' 作为对 b 的猜测.

在步骤 2 中,$\mathcal{D}$ 要想在验证查询阶段得到返回值为 1,则需要其输入的签名 σ_{M^*} 合法,而合法的签名将通过 Hash 查询获得,但 $\mathcal{D}$ 有可能不通过 Hash 查询而猜出一个合法的签名,这种情况的概率为 $1/q$.因此,通过 Hash 查询获得合法签名的概率最小为 $(1-1/q)$.而 $\mathcal{D}$ 赢得 $Game_{anony}$ 的概率为 $(1/2+\delta)$,但是 $\mathcal{D}$ 有可能没有通过 Hash 查询,只是随机选择 l_{M^*} 使其赢得 $Game_{anony}$,这种情况的概率为 $(1/2+1/q)$,因此 $\mathcal{D}$ 经过上述步骤得到正确结果的概率为 $((1/2+\delta)-(1/2+1/q))$,所以 $\mathcal{E}$ 解决 GDH 问题的概率为:

$$\varepsilon_{GDH}\geqslant\left(1-\frac{1}{q}\right)\left(\left(\frac{1}{2}+\delta\right)-\left(\frac{1}{2}+\frac{1}{q}\right)\right)=\left(1-\frac{1}{q}\right)\left(\delta-\frac{1}{q}\right)>\delta-\frac{2}{q}.$$

由上式,ε_{GDH} 不可忽略,与 GDH 假设矛盾. 证毕.

定理 2. 在数据共享过程中,只有持有数据拥有方凭证的其他参与方,才有可能获得数据,即数据拥有方具备数据的可控性.

证明. 标识为 c_j 的数据请求方在请求授权时,数据拥有方更新累加器为 $Acc_{new}=Acc^{c_j}\bmod N$,将授权集合 C 更新成 $\widetilde{C}$,并生成证据 $w=u^{\prod_{\substack{i=1\\c_i\neq c_j}}^{n}c_i}\bmod N$,$c_i=\widetilde{C}$,数据服务器验证证据的过程为

$$w^{c_j}=(u^{\prod_{\substack{i=1\\c_i\neq c_j}}^{n}c_i})^{c_j}\bmod N=u^{\prod_{i=1}^{n}c_i}\bmod N,c_i\in\widetilde{C}.$$

而最新的累加器为 $Acc_{new}=u^{\prod_{i=1}^{n}c_i}\bmod N,c_i\in\widetilde{C}$,即 $Acc_{new}=w^{c_j}$,该条件为数据服务器返回数据的必要条件,而当 w 不是正确的授权证据时,根据强 RSA 假设,Acc_{new} 与 w^{c_j} 不相等,无法通过数据服务器的验证.因此,方案具有可控性. 证毕.

定理 3. 对于系统中的所有数据共享请求,监管方能够审核共享请求与响应,即方案具有可监管性.

证明. 数据请求方需要向监管方发送 $req_{Regulate}=(y_1,y_2)=\{Enc_{pk_R}(msg,\tilde{w}_B),\sigma_{DVS}\}$,监管方计算 $Dec_{sk_R}(y_1)=(msg,\tilde{w}_B)$,以验证请求是否合法,然后通过交互式协议,监管方最终验证 v_2^{μ} 是否等于 $w\cdot v_1^{\eta}$,其中

$$v_2^{\mu}=e(A,g)^{s+\eta r},$$

$$w\cdot v_1^{\eta}=e(A,g)^{s}e(\tilde{w}_B,g^{pk_B})^{\eta}=e(A,g)^{s}e(w_B^{r},g^{pk_B})^{\eta}=e(A,g)^{s}e(A,g)^{\eta r}=e(A,g)^{s+\eta r},$$

从而验证授权凭证 $\tilde{w}_B$ 是否有效,若有效则会将请求以 pk_R 加密上传到区块链,并且数据服务器的响应情况将用 pk_R 加密上传到区块链,监管方可以随时解密以跟踪数据使用情况. 证毕.

定理 4. 监管方权力限制.监管方无法破坏数据请求方隐私性,也不能冒充获得授权的数据请求方非法获得数据.

证明. 在本文方案中,数据请求方将用 SDVS 来请求监管方的审核,由于该 SDVS 方案具有非授权性和非传递性的特征,因此没有第三方可以验证或者生成合法的签名.除此之外,在监管方验证授权

凭证时,通过交互式证明协议,在不泄露凭证本身的前提下证明了正确性,因此监管方无法冒充合法的被授权方去获取数据. 证毕.

定理 5. 监管方可以通过对区块链上存储的信息进行读取,获取到某个敏感数据共享过程的相关信息,在发现问题时进行追责.

证明. 在本文方案中,数据服务器 S 在返回给数据请求方数据时,需要构建 $res_{\text{Data}}=(msg', pk_{\text{B}}, T')$,并计算 $Enc_{pk_{\text{R}}}(res_{\text{Data}})$ 上传至区块链中.res_{Data} 包含数据共享过程的参与方、共享内容等信息,监管方 R 可以解密这些信息从而进行追责. 证毕.

6.2 比较分析

将本文方案与文献[4,9,11,14]中的方案进行安全性对比,如表 1 所示:

Table 1 Comparison of Security

表 1 安全性对比

方案	隐私性	可控性	可监管性	监管方权力限制
文献[4]	有	强	无	无
文献[9]	无	强	弱	无
文献[11]	无	无	无	无
文献[14]	无	弱	无	无
本文方案	有	强	强	有

由表 1 可以看到,本文方案对比其他方案,在实现敏感数据可控的前提下还具有一定程度的隐私性,对于共享过程,监管方可以进行监管,同时监管方的权力将受到限制.

文献[10]同样使用访问控制技术进行权限的更新与撤销,将本文方案与文献[10]中的方案进行复杂度的对比,结果如表 2 所示:

Table 2 Comparison of Complexity

表 2 复杂度对比

方案	权限验证时间复杂度	权限更新时间复杂度	空间复杂度
文献[10]	$O(n)$	$O(1)$	$O(n)$
本文方案	$O(1)$	$O(n)$	$O(1)$

由于文献[10]中通过访问控制表进行访问控制,当需要验证用户权限时,需要遍历整个访问控制表,所以其时间复杂度为 $O(n)$,且需要存储整张表,空间复杂度为 $O(n)$.而本文方案中使用累加器进行访问控制,只需要用户自己的凭证,结合公开的累加器即可验证,其时间复杂度为 $O(1)$,并且验证方只需要累加器本身即可进行验证,无需额外存储信息,空间复杂度为 $O(1)$.所以本文方案的访问控制结构比较高效.并且本文方案相比文献[10],在做到高效访问控制的同时,还实现了一定的隐私性和追责性.

文献[21]使用代理重加密和可搜索加密技术进行医疗数据共享,其数据共享阶段包括陷门生成、搜索、解密 3 个阶段.表 3 展示了常用密码学计算步骤的计算成本,将本文方案与文献[21]进行比较,2 个方案的计算成本如表 4 所示.

Table 3 Computational Cost of Common Cryptographic Algorithms

表 3 常用密码算法的计算成本

符号	含义	时间/ms
T_{e}	指数运算时间	0.341 8
T_{e2}	二次指数运算时间	0.413 9
T_{m}	群上乘法运算时间	0.001 9
T_{p}	双线性映射时间	13.673 6
T_{h}	Hash 运算时间	0.006 0
T_{i}	循环群中求逆运算时间	0.025 6

Table 4 Computational Cost of Schemes

表 4 方案计算成本

方案	数据加密	数据共享
文献[21]	$2T_{\text{p}}+8T_{\text{e}}+7T_{\text{h}}$	$5T_{\text{e}}+2T_{\text{p}}+3T_{\text{h}}+4T_{\text{m}}+5T_{\text{i}}$
本文方案	$3T_{\text{e}}+T_{\text{m}}$	$12T_{\text{e}}+2T_{\text{p}}+3T_{\text{e2}}+2T_{\text{h}}+4T_{\text{m}}+2T_{\text{i}}$

综合表 3 和表 4 可得,本文方案在数据加密阶段的计算成本明显低于文献[21].在数据共享阶段,文献[21]的计算成本约为 29.209 8 ms,本文方案的计算成本约为 32.761 3 ms,计算成本增加了约 12%,这主要是因为本文方案在数据共享阶段设置监管方对数据共享过程进行监管,提高了方案的安全性,所以计算成本有所增加,但在可接受的范围内.

在方案仿真部分,本实验使用 Java 语言,调用 java.math.BigInteger 库与第三方 jpbc 库,针对本文方案中的各种算法进行了仿真.实验环境配置如表 5 所示:

Table 5 Experimental Environment Configuration

表 5 实验环境配置

硬件	版本/型号
操作系统	Windows 10
内存	16 GB RAM
CPU	Intel Core i5 2.5 GHz

实验分别设置安全参数长度为 512 b 和 1 024 b，记录每个算法的运行时间，为了保证实验数据的可靠性，每一个算法的运行时间均多次重复运行后取平均值，不同安全参数下各算法的运行时间如表 6 所示.

从表 6 中可以看出，在安全性参数长度为 512 b 时，*Setup* 算法和 *AccInit* 算法由于需要生成一系列的参数所以耗时比较大，但是这 2 个算法往往只在系统启动时执行一次，所以不会对系统整体效率产生影响.其余算法运行时间均为毫秒级别，效率比较高.当提高安全性参数时，只有 *Setup* 和 *AccInit* 两个算法受到比较大的影响，其余算法均比较稳定.综上，本文方案在保证了较高安全性的同时，还具有良好的运行效率.

Table 6 Running Time of Algorithms

表 6 算法运行时间 ms

安全性参数长度/b	*Setup*	*KeyGen*	*AccInit*	*Share Request*	*Authorization*	*Regulate Request*	*Regulate Verify*	*Data Response*	*Revoke*
512	1 483.125	7.500	1 105.875	25.750	167.125	4.375	23.875	3.250	140.000
1 024	14 947.240	28.429	1 639.190	29.381	213.619	19.190	40.333	11.857	167.400

7 结束语

本文基于动态累加器、强指定验证者签名和区块链技术，提出了一种适用于敏感数据共享的方案，该方案具有隐私性、可控性、可监管性、可问责性和监管约束性的特点.通过对比分析证明了本文方案的安全性.本文的主要贡献为：

1) 通过动态累加器技术，数据拥有方可以灵活地授予用户权限或者撤销已经发出的授权；

2) 设置监管方对整个敏感数据共享过程进行监管，监管方将审核每一个数据共享请求和响应情况，从而避免出现非法共享行为；

3) 引入强指定验证者签名技术，使得数据请求方在请求监管方审核时能够保护自身的身份隐私，从而使得方案具有一定的隐私性；

4) 敏感数据共享过程的有关信息将由数据服务器以监管方公钥加密后上传至区块链，监管方可以获得这些信息来对数据共享过程进行追责；

5) 引入交互式零知识证明技术，在不影响监管方发挥监管职责的前提下，限制监管方能够获得的信息，避免恶意监管方破坏系统的安全性.

未来工作将主要聚焦在减弱本文方案的假设，比如考虑在数据存储服务器半可信或者不可信前提下的共享方案，以及考虑当多个用户同时发起共享请求时，如何提高系统效率的方法.

作者贡献声明：张正昊完成了论文所提方案的设计、仿真工作，以及论文初稿撰写；李勇、张振江与张正昊一起讨论所提方案的可行性，并在方案框架、方案仿真与分析方面进行指导.

参 考 文 献

[1] Guo Zijing, Luo Yuchuan, Cai Zhiping, et al. Overview of privacy protection technology of big data in healthcare [J]. Journal of Frontiers of Computer Science and Technology, 2021, 15(3): 389-402 (in Chinese)
(郭子菁, 罗玉川, 蔡志平, 等. 医疗健康大数据隐私保护综述[J]. 计算机科学与探索, 2021, 15(3): 389-402)

[2] He Peiyu. Research on the application model and value of big data based on Internet finance [J]. China Business and Market, 2017, 31(5): 39-46 (in Chinese)
(何培育. 基于互联网金融的大数据应用模式及价值研究[J]. 中国流通经济, 2017, 31(5): 39-46)

[3] Sima Hong. Establish a unified data standard system and promote the formulation of standards for data interconnection across industries [J]. China Standardization, 2019 (7): 14-14 (in Chinese)
(司马红. 建立统一数据标准体系 推进跨行业数据互联互通标准制定[J]. 中国标准化, 2019 (7): 14-14)

[4] Dong Xinhua, Li Ruixuan, He Heng, et al. Secure sensitive data sharing on a big data platform [J]. Tsinghua Science and Technology, 2015, 20(1): 72-80

[5] Shamir A. Identity-based cryptosystems and signature schemes [G] //LNCS 196: Proc of the CRYPTO 1984. Berlin: Springer, 1984: 47-53

[6] Diffie W, Hellman M. New directions in cryptography [J]. IEEE Transactions on Information Theory, 1976, 22(6): 644-654

[7] Rosenblum M, Garfinkel T. Virtual machine monitors: Current technology and future trends [J]. Computer, 2005, 38(5): 39-47

[8] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system [EB/OL]. 2008 [2021-10-08]. https://bitcoin.org/bitcoin.pdf

[9] Qi Xia, Sifah E B, Smahi A, et al. BBDS: Blockchain-based data sharing for electronic medical records in cloud environments [J]. Information, 2017, 8(2): 44-59

[10] Qi Xia, Sifah E B, Asamoah K O, et al. MeDShare: Trust-less medical data sharing among cloud service providers via blockchain [J]. IEEE Access, 2017, 5: 14757-14767

[11] 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 (in Chinese)
(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)

[12] Song Xiaodong, Wagner D, Perrig A. Practical techniques for searches on encrypted data [C] //Proc of the 21st IEEE Symp on Security and Privacy. Piscataway, NJ: IEEE, 2000: 44-55

[13] Shamir A. How to share a secret [J]. Communications of the ACM, 1979, 22(11): 612-613

[14] Li Li, Zeng Qingxian, Wen Yihong, et al. Datasharing scheme based on the blockchain and the proxy re-encryption [J]. Netinfo Security, 2020, 20(8): 16-24 (in Chinese)
(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)

[15] Blaze M, Bleumer G, Strauss M. Divertible protocols and atomic proxy cryptography [G] //LNCS 1403: Proc of the EUROCRYPT 1998. Berlin: Springer, 1998: 127-144

[16] Zhou Zhengqiang, Chen Yuling, Li Tao, et al. Medical data security sharing scheme based on consortium blockchain [J]. Journal of Applied Sciences, 2021, 39(1): 123-134 (in Chinese)
(周正强, 陈玉玲, 李涛, 等. 基于联盟链的医疗数据安全共享方案[J]. 应用科学学报, 2021, 39(1): 123-134)

[17] Sahai A, Waters B. Fuzzy identity-based encryption [G] //LNCS 3494: Proc of the EUROCRYPT 2005. Berlin: Springer, 2005: 457-473

[18] Boneh D. The decision Diffie-Hellman problem [G] //LNCS 1423: Proc of the 3rd Int Symp on Algorithmic Number Theory. Berlin: Springer, 1998: 48-63

[19] Okamoto T, Pointcheval D. The gap-problems: A new class of problems for the security of cryptographic schemes [G] //LNCS 1992: Proc of the Public Key Cryptography-PKC'01. Berlin: Springer, 2001: 104-118

[20] Huang Qiong, Yang Guomin, Wong Duncan S, et al. Efficient strong designated verifier signature schemes without random oracle or with non-delegatability [J]. International Journal of Information Security, 2011, 10(6): 373-373

[21] Niu Shufen, Chen Lixia, Li Wenting, et al. Electronic medical record data sharing scheme based on blockchain [J/OL]. Acta Automatica Sinica, 2021 [2021-10-08]. https://doi.org/10.16383/j.aas.c190801 (in Chinese)
(牛淑芬, 陈俐霞, 李文婷, 等. 基于区块链的电子病历数据共享方案[J/OL]. 自动化学报, 2021 [2021-10-08]. https://doi.org/10.16383/j.aas.c190801)

Zhang Zhenghao, born in 1997. Master. Student member of CCF. His main research interests include blockchain and information security.

张正昊,1997 年生.硕士.CCF 学生会员.主要研究方向为区块链和信息安全.

Li Yong, born in 1973. PhD, associate professor, PhD supervisor. Senior member of CCF. His main research interests include cryptography, cloud computing security, and blockchain.

李 勇,1973 年生.博士,副教授,博士生导师.CCF 高级会员.主要研究方向为密码学、云计算安全和区块链.

Zhang Zhenjiang, born in 1973. PhD, professor, PhD supervisor. His main research interests include edge computing, IoT and authentication.

张振江,1973 年生.博士,教授,博士生导师.主要研究方向为边缘计算、物联网、身份认证.