专题：内生安全

基于区块链的云数据匿名确定性删除方法

王双星[1]，罗劲瑭[2]，帅莉莎[1]，张佳敏[1]，张敏[1]，阳小龙[1]
（1. 北京科技大学计算机与通信工程学院，北京 100083；
2. 国网四川省电力公司经济技术研究院，四川 成都 610041）

摘　要：现有确定性删除方案忽略了用户数据与用户身份的关联性，使用户的删除行为暴露给攻击者或云服务提供商。为解决此问题，提出一种基于区块链的云数据匿名确定性删除方法。该方法改进了可链接环签名方案，使用户可以通过控制签名中的链接标记在匿名情况下实现确定性删除；同时，它利用区块链记录删除行为，使其具有不可抵赖性。理论分析和实验表明：该方法不仅能满足用户数据的确定性删除要求，而且具有匿名性可以切断用户数据与其身份的关联，从而有效避免攻击者或云服务提供商对用户行为的追踪分析。

关键词：确定性删除；区块链；可链接环签名；隐私保护
中图分类号：TP393
文献标识码：A
doi: 10.11959/j.issn.1000−0801.2021049

Cloud data anonymous assured deletion approach based on blockchain

WANG Shuangxing[1], LUO Jintang[2], SHUAI Lisha[1], ZHANG Jiamin[1], ZHANG Min[1], YANG Xiaolong[1]
1. School of Computer and Communications Engineering, University of Science & Technology Beijing, Beijing 100083, China
2. Economic Technology Research Institute of Sichuan Electric Power Company, Chengdu 610041, China

Abstract: The previous assured deletion schemes ignored that the privacy is threated by the relevance between data and user identity, exposing user behavior to the cloud service provider or attackers. Aiming at the above problems, an assured deletion approach based on blockchain was proposed. The improved linkable ring signature scheme was adopted to make user achieve anonymous assured deletion by setting specific link symbol in the signature. Moreover, the blockchain technology was leveraged to guarantee the nonrepudiation of assured deletion. Theoretical analysis and experimental results show that the proposed approach can meet the assured deletion requirements of user data in cloud storage, and its anonymity can effectively prevent attackers from tracking and analyzing user behavior.

Key words: assured deletion, blockchain, linkable ring signature, privacy-preserving

收稿日期：2021−02−01；**修回日期**：2021−03−15
基金项目：国家自然科学基金资助项目（No.61941113, No. 61971033）
Foundation Items: The National Natural Science Foundation of China (No. 61941113, No. 61971033)

1　引言

云计算服务在大数据时代为人们带来了极大的方便，其计算能力强、存储空间大的优势吸引了无数用户。用户将数据上传到云端后，数据的所有权与其使用权处于分离状态，容易造成个人隐私信息泄露的问题。确定性删除作为云数据安全存储领域的核心技术之一，旨在保障用户数据隐私，使超过用户授权期限的数据在 CSP（cloud service provider，云服务提供商）处失效且不可恢复。实现云端数据的确定性删除，是用户实现对其参与共享内容的有效管控和保护自身隐私的重要途径。

根据调研，现有研究主要实现了 3 个层面的确定性删除[1]。

- 第一层意义上的确定性删除是指将云端数据删除问题转换为用户对密钥的安全删除问题。其研究成果可以分为基于密钥集中管理和密钥分散管理两大类。集中式密钥管理的代表方案为 Nair 等[2]提出的基于 Ephemerizer 系统的确定性删除方案，通过第三方密钥管理服务器生成公钥加密数据密钥，当密钥管理服务器删除私钥之后数据密钥和密文将无法解密访问，从而实现数据的确定性删除。Tang 等[3]对上述方案进行了扩展，提出基于策略的确定性删除方案（FADE），其基本思想是将数据用 DK（data key，数据密钥）加密并与访问策略的布尔组合进行关联，每条访问策略与一个 CK（control key，控制密钥）关联，如果 CK 被密钥管理者删除则原数据便不可恢复。分散式密钥管理的代表性方案是 Geambasu 等[4]提出的 Vanish 方案。其主要思想是：用户将数据用随机生成的密钥 K 进行加密，并将此密钥进行门限密码处理后随机分发到 DHT[5]（distributed hash table，分布式哈希表）网络中，DHT 网络的动态更新特性会将密钥分片删除，此时密钥 K 将不能进行重构，达到用户数据确定性删除的目的。为了解决 Vanish 方案易受 Sybil 攻击[6]的缺陷，Zeng 等[7]提出一种解决方案 SafeVanish，通过增加密文分量长度并利用 RSA 加密对称密钥的方式抵抗 Sybil 攻击。
- 第二层意义上的确定性删除指云内或跨云的数据副本删除问题。针对云环境下数据多副本难以管理的问题，熊金波等[8]提出一种多副本安全共享与关联删除方案，方案利用 RAO（replication associated object，副本关联对象）封装用户数据副本，CSP 维护副本目录记录和追踪用户数据所有副本信息。不同服务器之间的同步机制可以监测 RAO 产生或删除副本的操作行为，保证了跨云副本的精确管理。
- 第三层意义上的确定性删除是指实现云端数据的删除反馈机制。Zhang 等[9]基于数据完整性检查提出可证明可追溯的确定性删除方案，云端对数据确定性删除后需要按照用户指定的模式对原始存储位置进行覆写，并利用区块链公开透明的特性进行覆写结果的公众验证。杜瑞忠等[10]构建基于脏数据块覆写的可搜索路径散列二叉树（DSMHT），对要删除的数据进行覆写后再进行正确性验证。余海波[11]提出基于区块链的数据确定性删除协议，利用区块链保存服务器生成的数据删除证据和经证据构造得到的 Merkle 树的根值。用户通过区块链上的证据信息，对收到的删除结果进行验证，确保服务器的行为诚实和数据不可恢复。

由上述调研可知，现有确定性删除方法聚焦的重点是确保用户可以通过删除数据密钥等手段使其共享数据不再可以访问，并使删除结果更具有说服力；但现有方法都忽略的是除了数据本身，用户的共享和删除行为也是攻击者关注的对象。

这是因为通过将用户行为与其身份进行关联，CSP 或攻击者可以监视用户的一举一动，甚至对用户进行肖像描绘、兴趣预测，并在用户不知情的情况下将分析结果随意传播以换取经济利益，这无疑会给用户隐私带来威胁。所以，如何实现在不透露用户身份的情况下在云环境下共享数据、删除（撤销）数据是解决数据共享参与用户隐私保护的另一重要途径。针对上述问题，本文提出的确定性删除方法实现了用户数据匿名上云，并在保持用户匿名情况下实现数据的确定性删除，能有效避免攻击者对用户行为的追踪分析。本文的主要贡献如下。

- 改进了可链接环签名方案，可同时兼顾用户匿名性和用户自主性，即用户可在匿名情况下自由决定签名中的链接标记。这样，用户可以将加密数据与环签名信息上传到CSP，在确定性删除阶段用户生成带有同样链接标记的签名，CSP 即可通过链接数据上传阶段和数据确定性删除阶段的签名信息实现匿名确权从而执行确定性删除。
- 构建了不同类型的区块链交易以实现密钥信息分散管理和确定性删除，避免了集中密钥管理的单点失效问题，也不需要引入可信第三方，简化了系统结构；同时得益于区块链难以篡改和可追溯的特性，使确定性删除操作被永久记录在区块链并且不可抵赖，更具有“确定性”。

2 理论基础

（1）Shamir 门限秘密共享方案[12]

其基本思想是，将秘密 k 以某种方式分成 n 份 $k_1,k_2,\cdots,k_n$，并满足：

- 任意 t 或 t 个以上 k_i 能够计算出 k；
- 任意少于或等于 t–1 个 k_i 无法计算出 k。

（2） 可链接环签名[13]

又称为 LSAG（linkable spontaneous anonymous group，可链接自发匿名群组）签名。可链接环签名有以下特性。

- 匿名性：若签名者所在环中有 n 个成员的公钥，攻击者无法确定签名由环中哪个成员生成。
- 自发性：没有群组秘密，不需要群组管理员。
- 不可伪造性：环中其他成员不能伪造真实签名者对消息 m 的签名，攻击者在获得某个有效签名的基础上，也不能为消息伪造一个签名。
- 可链接性：若诚实签名者在同一环中对消息 m 和 m'签名得到两个可链接环签名，那么这两个环签名拥有相同的链接标记。根据链接标记，环中成员能在不知签名者身份的情况下验证两个签名是否由同一签名者生成。

本文对可链接环签名方案进行了改进，使其能在同一签名群体中生成带有不同链接标记的签名。

（3）区块链[14-15]

区块链实际上是一种分布式数据库或公共分类账本，由一系列按时间顺序排列的区块组成，其中记录着经过验证的交易或者数字事件。最重要的是，区块链每个区块都包含时间戳和前一个区块的哈希值（如图 1 所示），这使得区块链的数据不可更改并且可追踪。分布式网络中的节点达成共识后，新的有效区块才会被添加到区块链。此外，分布式网络中的每个节点都保留相同的区块链副本，这使得区块链能有效防止单点失效问题。

文中所用符号缩写及含义见表 1。

3 基于区块链的确定性删除

3.1 系统组成

系统模型如图 2 所示，基于区块链的确定性删除模型由 4 个实体组成：数据拥有者（data owner，DO）、数据使用者（data user，DU）、区块链网络和 CSP（数据拥有者和数据使用者均为

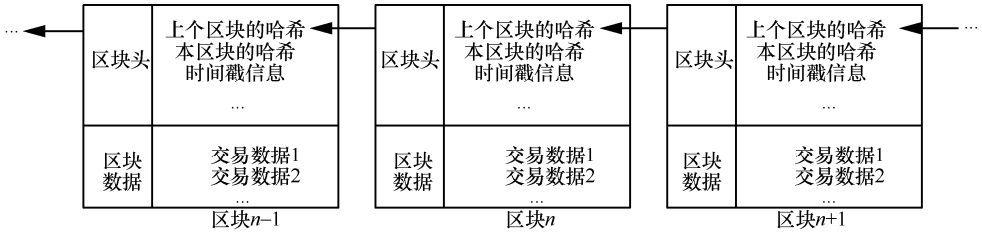


图 1　区块链的结构

表 1　符号缩写

缩写	含义
SDO	secret data object，秘密数据封装体
DVO	data visit object，数据访问封装体
GF(p)	由 p 确定的有限域
$\mathbb{Z}_q$	q 阶整数域
$H(\cdot)$	消息摘要函数
$H_1(\cdot)$, $H_2(\cdot)$	统计独立的单向散列函数，其中 $H_2(0)=0$
$H_3(\cdot)$	消息映射函数，将消息映射到整数域，以便进行运算。$H_3^{-1}(\cdot)$ 是其逆运算
$data_1 \| data_2$	将 $data_1$ 与 $data_2$ 进行拼接，‖为字符串拼接符号

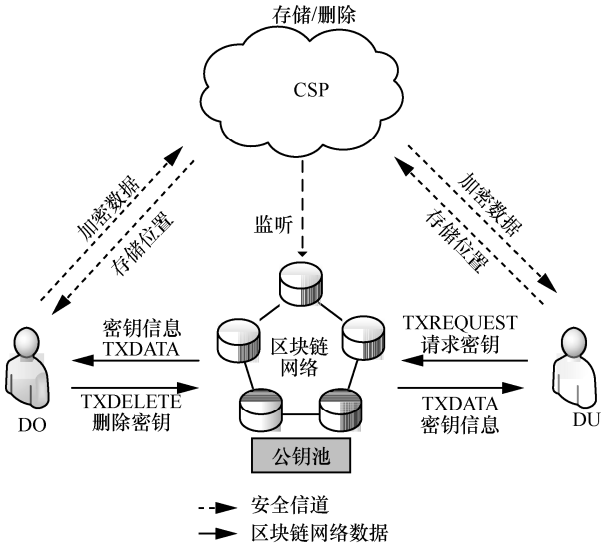


图 2　系统模型

区块链中的节点，为方便叙述将其分开表示）。其中，CSP 负责存储、返回用户的加密数据，监听区块链交易信息并执行云数据的删除；数据拥有者将密钥信息通过交易分散到区块链网络中并将加密数据上传到 CSP；数据使用者通过交易请求从区块链中恢复密钥信息并解密从 CSP 下载的加密数据；区块链网络负责密钥信息的分散管理，同时验证交易信息的合法性。本文参考 Li 等[16]的方案为区块链网络构建了 3 种类型的区块链交易，分别为 TXDATA、TXREQUEST 和 TXDELETE，分别用于数据上传、数据请求和数据确定性删除。每个区块链节点还拥有一个链下数据库，用以存储密钥信息经 Shamir 门限秘密拆分得到的子秘密。此外，区块链节点共同维护一个公钥池，其中存储所有节点的环签名公钥和 RSA 公钥，所有节点都能获取其中存储的公钥列表。

3.2　模型假设

对于上述系统模型，本文定义如下假设条件。

（1）云服务提供商（CSP）：诚实地存储用户数据，但是对用户数据好奇，可能不诚实删除用户数据。

（2）数据使用者：数据使用者是可信的，不会将恢复的明文信息透露给其他人。实际上，一旦数据使用者获得数据访问权限，诸如截屏、拍照等较高层级的操作是无法被禁止的[17]，所以本文认为假设合理。

（3）区块链网络：当收到外部交易时，每一个区块链网络中的节点会根据交易类型诚实执行相应命令，即每个节点会自动执行数据存储和数据删除的操作。假设合理性基于区块链网络的安全性，即某个节点想要改变当前区块链网络状态，则其需要利用自己想要的执行结果说服其他节点或者发动 51%攻击，无论哪一种方式都需要付出

极大的代价。

4 基于区块链的匿名确定性删除方法

4.1 设计思想

数据拥有者将数据加密后上传到 CSP，将密钥信息和存储位置利用 Shamir 秘密分享策略进行拆分，并通过区块链交易广播到不同区块链节点中存储。这样不仅避免了使用密钥管理服务器容易发生的单点失效问题，同时也使密钥信息不容易被攻击者窃取。

数据拥有者和数据使用者是对等的区块链节点。为实现匿名确定性删除，所有区块链节点发送交易时不携带身份信息，即匿名发送交易。但是匿名情况下确保交易发送者身份合法性是需要解决的问题，而可链接环签名可以实现匿名情况下的身份认证，故本文利用可链接环签名对交易进行签名和验证。在数据上传阶段，数据拥有者为 TXDATA 交易添加特定链接标记，当数据拥有者想要结束共享时，则发送带有相同链接标记的 TXDELETE 交易通知区块链节点和 CSP 执行删除操作。由于可链接环签名的不可伪造性，攻击者无法伪造带有相同链接标记的合法签名，也就是说只有数据拥有者才能删除自己共享的数据，保证了数据安全性。

本文构建了 3 种不同类型的交易，设计了区块链中交易发送算法 SendTX 和交易处理算法 HandleTX，算法会根据交易类型的不同控制区块链节点执行不同的操作；由于本方法的匿名要求，将交易的签名算法替换为改进的可链接环签名算法，实现交易的匿名验证。如果数据拥有者将密钥信息直接发送到区块链，区块链的公开性会使用户数据直接暴露给所有人，这是不能接受的。针对这一问题，本文采用 RSA 加密算法将交易信息加密后发送。当数据拥有者准备将密钥信息通过交易信息发送到区块链时，其从公钥池中随机选择 RSA 公钥对交易进行加密，这样确保了只有该公钥的持有者（即数据拥有者所选择的交易接收者）可以解密并存储密钥信息。

4.2 方案设计

方案可以分为 3 个阶段描述：数据上传阶段、数据请求阶段和数据确定性删除阶段。假设区块链由 n 个节点组成，节点 i 可以用五元组（Node_i，PK_i、SK_i，y_i，x_i）表示，其中 Node_i 为节点名称，PK_i、SK_i 为 RSA 加密公钥和私钥，y_i、x_i 为可链接环签名公钥和私钥。特别地，用 π（$1\leqslant\pi\leqslant n$）表示作为数据拥有者的节点，用 μ（$1\leqslant\mu\leqslant n$）表示数据使用者节点。在数据上传阶段之前，所有节点生成 RSA 密钥对和可链接环签名密钥对，并将公钥发送到公钥池中得到公钥列表 $L_1=\{\text{PK}_1, \text{PK}_2,\cdots,\text{PK}_n\}$ 和 $L_2=\{y_1,y_2,\cdots,y_n\}$。

4.2.1 数据上传阶段

数据拥有者在数据上传阶段将加密数据上传到 CSP，同时将密钥信息通过 TXDATA 类型的交易拆分并分散到不同区块链节点存储。这样不仅避免了单独使用密钥管理服务器的单点失效问题，而且当数据拥有者想要结束共享时，可以随时发送 TXDELETE 类型的交易撤销密钥信息进行数据的确定性删除。数据上传阶段如图 3 所示，共分为 6 个步骤。

步骤 1 假设数据拥有者待上传的加密数据为 E，其对称加密密钥为 KEY。在数据上传之前，数据拥有者向云服务提供商发送存储请求。

步骤 2 云服务提供商为数据拥有者预先分配并返回存储地址 ADDR，数据拥有者将 KEY 和 ADDR 封装成秘密数据封装体 SDO。

步骤 3 数据拥有者运行 Shamir 门限秘密共享算法将 SDO 拆分成多个子秘密。具体计算过程如下。

假设区块链网络共有 n 个节点，数据拥有者选择秘密恢复门限值 $t>n/2$（防止 51%攻击），选择一个大素数 p，在 GF(p)中随机选择系数 a_{t-1}，$a_{t-2},\cdots,a_1$，且令 $a_0=H_3(\text{SDO})$，其中

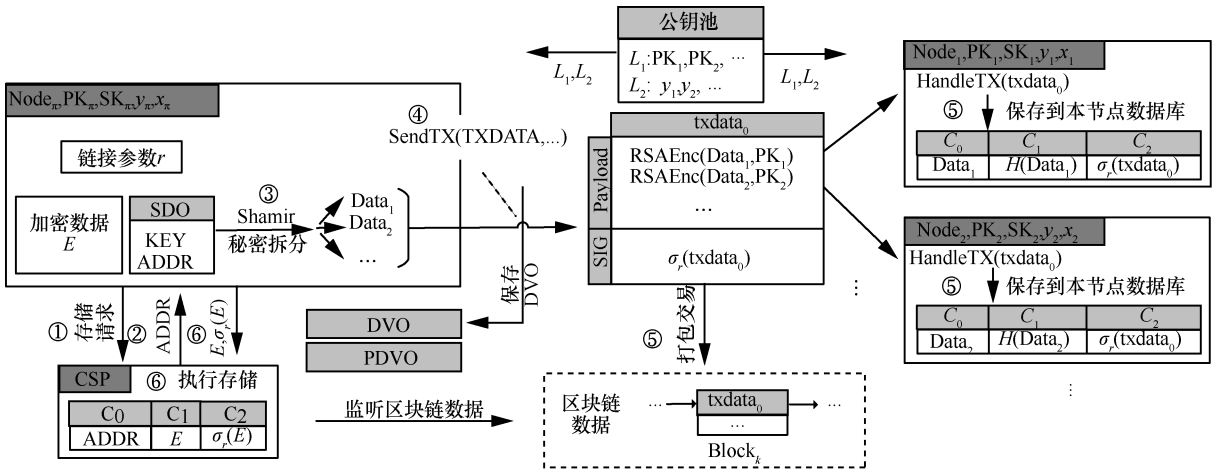


图 3　数据上传阶段

$H_3:\{0,1\}^*\to\mathbb{Z}_p$。构造 GF($p$)上的 $t-1$ 次多项式：

$$f(x)=a_{t-1}x^{t-1}+a_{t-2}x^{t-2}+\cdots+a_1x+a_0 \bmod p \quad (1)$$

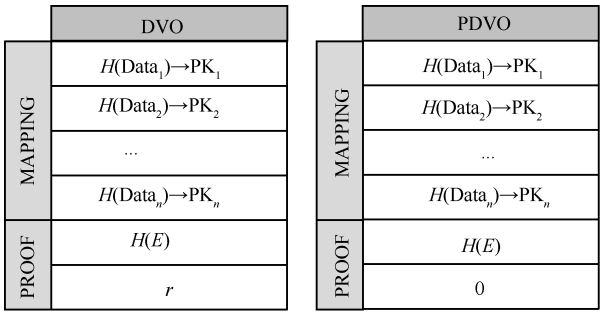
在 GF(p)中选取 n 个非零、互不相同的 $x_1,x_2,\cdots,x_n$，令 i=1,2,⋯,n，计算：

$$\text{Data}_i=y_i=f(x_i)\bmod p \quad (2)$$

并将 $x_1,x_2,\cdots,x_n$ 和 p 公开，将 $\text{Data}_i(i=1,2,\cdots,n)$ 输出。

步骤 4　数据拥有者运行交易发送算法 SendTX(TXDATA,[Data$_1$,Data$_2$,⋯,Data$_n$],None,r)→{txdata$_0$}，参数一表示发送 TXDATA 类型的交易，参数二表示要发送的数据，参数三表示交易接收者公钥，数据上传阶段接收者公钥从公钥列表随机选择，所以设置为 None（空）。参数四表示可链接环签名的链接参数。数据拥有者会将 Data$_i$ 用接收者公钥进行 RSA 加密填充到 Payload 字段，对交易进行可链接环签名后填充到 SIG 字段，然后将交易广播到区块链网络中，最后将算法运行过程所用参数保存到 DVO（如图 4 所示）以便在数据请求阶段使用。

步骤 5　区块链节点收到交易广播运行此交易处理算法 HandleTX(txdata$_0$)。算法判断交易为 TXDATA 类型后，验证交易的可链接环签名是否合法，若合法则将交易打包到区块链。打包交易后，区块链节点尝试解密 txdata$_0$ 交易的 Payload



(a) 数据访问封装体DVO　　(b) 部分数据访问封装体PDVO

图 4　两种数据访问封装体

字段，若能解密则将数据保存到数据库。以图中 3 中 Node$_1$ 节点为例，Node$_1$ 可以解密 Payload 字段的第一行，得到 Data$_1$ 后将 Data$_1$ 及其哈希值和 txdata$_0$ 的 SIG 字段保存到数据库，其他节点同理。由图 3 可知，数据拥有者 Node$_\pi$ 已经将 SDO 子秘密 Data$_1$,Data$_2$,⋯,Data$_n$ 分散到 Node$_1$,Node$_2$,⋯,Node$_n$。

步骤 6　数据拥有者运行可链接环签名的签名算法 LSAGSig(E,r)→{$\sigma_r(E)$}对加密数据 E 进行可链接环签名，链接参数为 r，并将签名结果与加密数据 E 一同尝试发送给 CSP。CSP 运行可链接环签名的链接算法 LSAGLink($\sigma_r(E)$, σ_r(txdata$_0$))→True or False，如果算法输出 True，表示当前 E 的发送者与 txdata$_0$ 是同一用户，其身份已经由区块链网络验证，CSP 即可从数据拥有者接受加密数据 E。

数据上传阶段结束后，CSP 中已经存储了加密数据 E，数据拥有者将步骤 4 中所用 Data_i 的哈希值 $H(\text{Data}_i)$和其对应 RSA 公钥封装到 DVO（data visit object，数据访问封装体）的 MAPPING 字段，将加密数据 E 的哈希值 $H(E)$和链接参数 r 封装到 DVO 的 PROOF 字段（如图 4（a）所示）。除此之外，数据拥有者还封装一个 PDVO（partial data visit object，部分数据访问封装体）（如图 4（b）所示），PDVO 省去了 DVO 的链接参数 r，没有链接参数 r 意味着无法生成合法的 TXDELETE 交易，所以 PDVO 不具有确定性删除权限。数据拥有者可以将 PDVO 交给其他数据使用者而不必担心数据被恶意删除。

4.2.2　数据请求阶段

数据使用者利用部分数据访问封装体（PDVO）访问数据拥有者共享的数据，共分为 5 个步骤，如图 5 所示。

步骤 1　数据使用者提取部分数据访问体中 MAPPING 字段（t 行，t 为 Shamir 秘密分享策略设置的门限值）的 $H(\text{Data}_i)$与其对应的 RSA 公钥 PK_i，运行发送交易算法 SendTX(TXREQUEST, $\{\text{Data}_1,\text{Data}_2,\cdots,\text{Data}_t\},\{\text{PK}_1,\text{PK}_2,\cdots,\text{PK}_t\},r)\rightarrow$ {txrequest}。

步骤 2　区块链所有节点运行接收交易算法 HandleTX(txrequest)，成功验证交易签名信息验证后将交易打包到区块链，并解密 Payload 得到数据请求者公钥 PK_μ 和请求数据的摘要 $H(\text{Data}_i)$，区块链节点根据数据摘要查找数据库 C_1 字段，若查找成功则向数据请求者返回一个 TXDATA 交易，即再次运行算法 SendTX(TXDATA,Data_i,PK_μ,0) → {txdata_i}。数据请求阶段只需要验证交易合法性即可，故链接参数设置为 0。在图 5 中，Node_1，$\text{Node}_2,\cdots,\text{Node}_t$ 节点分别返回 TXDATA 类型的交易 $\text{txdata}_1,\text{txdata}_2,\cdots,\text{txdata}_t$。

步骤 3　数据使用者运行交易处理算法 HandleTX(txdata_i)，验证 txdata_i 的签名后打包交易到区块链，并解密 Payload 字段可以得到秘密封装体 SDO 子秘密 $\text{Data}_1,\text{Data}_2,\cdots,\text{Data}_t$。

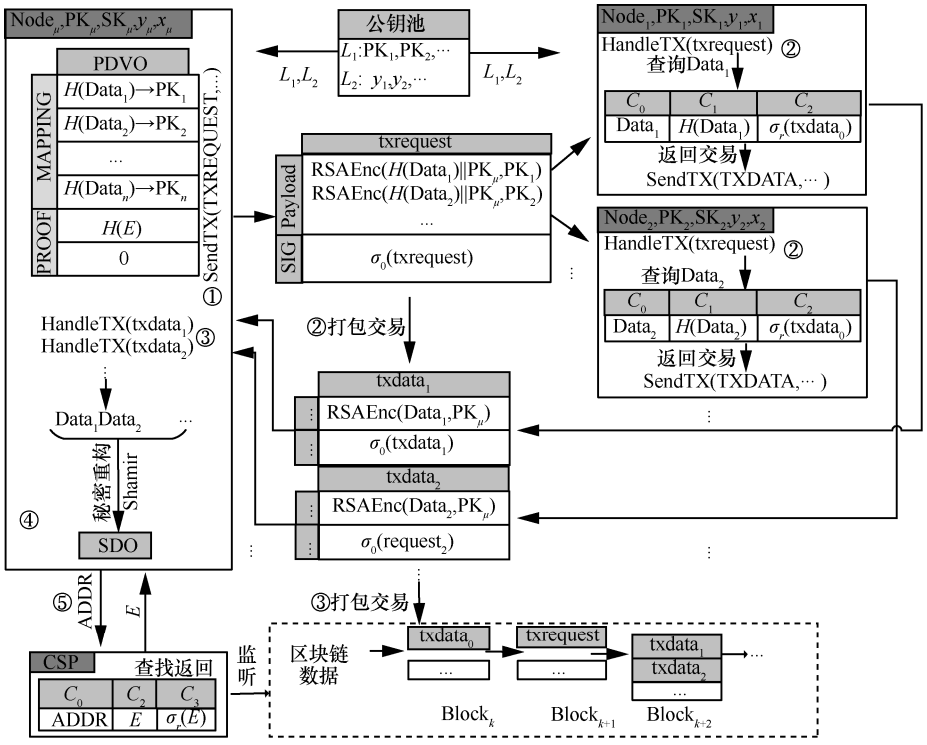


图 5　数据请求阶段

步骤 4　数据使用者对 SDO 子秘密进行 Shamir 秘密重构，将 t 组(x_i,Data$_i$)代入拉格朗日插值计算式重构 $f(x)$：

$$f(x)=\sum_{i=1}^{t}\left(\text{Data}_i\cdot\prod_{1\leqslant j\leqslant t,j\neq i}\frac{x-x_j}{x_i-x_j}\right)\bmod p \quad (3)$$

得到 $f(x)$之后计算：

$$a_0=f(0) \quad (4)$$

$$\text{SDO}=H_3^{-1}(a_0) \quad (5)$$

即可恢复秘密数据封装体 SDO。

步骤 5　数据使用者将秘密数据封装体 SDO 中的存储位置 ADDR 发送至 CSP 请求加密数据 E。CSP 查找数据库的 C_0 字段，若查找成功则向数据使用者返回 C_1 字段即加密数据 E。数据使用者首先验证 E 的摘要值是否与 PDVO 中存储的 $H(E)$一致，若一致则用密钥 KEY 对加密数据 E 进行解密即可得到数据拥有者共享的数据。

经过数据请求阶段，数据使用者通过利用 PDVO 向区块链请求到数据拥有者的 SDO 信息，并利用 SDO 向 CSP 请求得到数据拥有者的加密数据。

4.2.3　数据确定性删除阶段

数据拥有者利用 DVO 进行确定性删除操作，以撤销其在数据上传阶段共享的数据，如图 6 所示，确定性删除阶段分为 3 个步骤。

步骤 1　数据拥有者提取 DVO 的 MAPPING 字段 SDO 子秘密的哈希值 H(Data$_i$)与其对应的 PK$_i$，执行算法 SendTX(TXDELETE, H(Data$_i$), PK$_i$,r)→{txdelete}。

步骤 2　区块链中所有节点执行交易接收算法 HandleTX(txdelete)，此算法验证交易签名、将交易打包到区块链后，对交易 Payload 字段进行解密得到 H(Data$_i$)，并在数据库中匹配 H(Data$_i$)对应的数据条目。若匹配成功则执行算法 LSAGLink(σ_r(txdelete), σ_r(txdata$_i$))→True or False，如果算法输出 True，表示此 txdelete 的发送者与 txdata$_0$ 是同一用户，完成了确定性删除的匿名身份认证。确定了删除发起者的身份后，区块链节点将 H(Data$_i$)对应的整行数据条目删除。

步骤 3　区块链节点删除了 SDO 子秘密后，加密数据 E 的密钥 KEY 与存储位置 ADDR 已经被永久销毁，CSP 中还存储着加密数据 E，需要将其删除以释放空间。具体做法是：CSP 监听到步骤 1 中 txdelete 交易信息，根据 σ_r(txdelete)在数据库中进行匹配得到 C_2 字段的可链接环签名信息 $\sigma_r(E)$，执行 LSAGLink(σ_r(txdelete),$\sigma_r(E)$)→True or

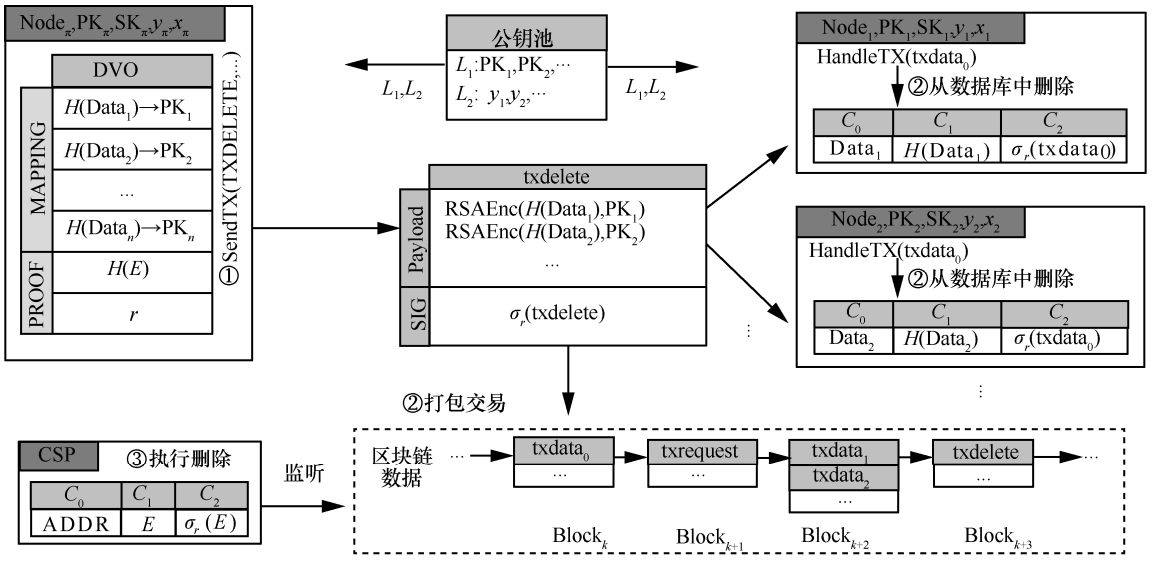


图 6　确定性删除阶段

False。若算法输出 True，表示此 txdelete 的发送者与 $txdata_0$ 是同一用户，完成确定性删除的匿名身份认证，将 $\sigma_r(E)$对应条目从本地数据库删除。至此，数据已完成确定性删除。

4.3 关键算法设计

4.3.1 区块链交易发送与交易处理算法

本方案的主要逻辑功能依靠区块链交易发送与交易处理算法实现。数据拥有者或数据使用者通过交易发送算法发送不同类型的区块链交易，区块链节点根据交易处理算法通过收到交易的类型执行相应操作。

算法 1 交易发送算法

输入 交易类型 Type，将被加密发送的数据列表 DatasOrHashs，目的节点 RSA 公钥列表 PKs，可链接环签名链接参数 r。

输出 类型为 Type 的交易 TX。

（1）function SendTX(Type, DatasOrHashs,PKs,r):
（2）begin
（3） if Type=TXDATA or TXDELETE then
（4） PK_{form}←None，LinkSym←r
（5） else if Type=TXREQUEST
（6） PK_{form}←PK_{π}，LinkSym←0
（7） end if
（8） if PK = None
（9） PK_{dest}←RandomChoose(L_1)
（10） else
（11） PK_{dest}←PKs
（12） end if
（13） for i=0 to PKs.length
（14） Payload[i]←RSAEnc(DatasOr Hashs[i]|| PK_{form}, PK_{dest}[i])
（15） end for
（16）σ_r(Payload)←LSAGSig(L_2, Payload, LinkSym, y_{π}, x_{π})
（17）return TX(Type, Payload, σ_r(Payload))
（18）end

算法 1 的第（3）~（7）步判断要发送的交易类型，如果是 TXDATA 或 TXDELETE 类型的交易，说明不需要区块链返回交易，故 PK_{form} 可以设置为空值 None；但这两种类型的交易需要发送者添加链接标记 r，将 r 保存到变量 LinkSym 以便后续调用。如果是 TXREQUEST 类型的交易，区块链会向自己返回 TXDATA 交易，所以需要将自己的 RSA 公钥赋值给 PK_{form}。算法第（8）~（12）步指定交易接收者 RSA 公钥（确保交易只能由特定接收者解密），如果 PKs 参数未设定（None），则从公钥池的 RSA 公钥列表中随机选择接收者公钥（RandomChoose）；如果 PKs 不为空，则说明这是数据使用者请求的交易，PKs 即数据使用者的公钥。第（13）~（15）步是加密过程，保证交易信息只有接收者能解密。第（16）步对交易进行可链接环签名，实现匿名情况下交易合法性的验证。算法最后返回交易信息 TX。

算法 2 交易处理算法

输入 接收到的交易 TX

输出 Type 为 TXDATA 的交易或者空

（1）function HandleTX(TX)
（2）begin
（3） if LSAGVer(L_2,TX.Payload,TX.σ_r (Payload)) =Reject
（4） exit
（5） end if
（6）try
（7）{DatasOrHashs || PK_{from}}← RSADec(TX.Payload, SK_{π})
（8）except 解密失败 exit
（9） if TX.Type=TXDATA
（10） $C0$,$C1$,$C2$ ← DatasOrHashs, H(DatasOrHashs), TX.σ_r(Payload)
（11） 将 $C0$,$C1$,$C2$ 分别保存到本节点数据库 C_0,C_1,C_2 列
（12） else if TX.Type=TXREQUEST

（13）*C*0,*C*1,*C*2←尝试查找本节点数据库中 C_1 列为 DatasOrHashs 的数据

（14）查找成功则执行 SendTX(TXDATA, *C0*, L_1, L_2, 0)，否则 exit

（15）　else if TX.Type is TXDELETE

（16）　*C*0,*C*1,*C*2←查找本节点数据库中 C_1 列为 DatasOrHashs 的数据

（17）　若查找失败，exit，否则继续

（18）　　if LSAGLink(*C2*,TX. σ_r(Payload)) =True

（19）　　删除本节点数据库中 C_1 列为 Data 的数据

（20）　　end if

（21）　end if

（22）end

算法 2 第（3）~（8）步首先验证交易，即验证交易的可链接环签名字段是否合法，若是非法交易则结束算法；若交易合法则会被打包到区块链，所有节点会尝试解密该交易，如果解密失败，说明此交易接收者不是自己，结束程序；若解密成功，则说明此交易是发给自己的，执行后续步骤。第（9）~（19）步判断交易类型并执行相应操作，如果判断交易类型为 TXDATA，则将交易数据保存到本地数据库。如果交易类型为 TXREQUEST，说明有数据使用者正在请求数据，尝试从本地数据库查找数据，如果找到则通过 TXDATA 交易返回该数据，否则退出算法。如果交易类型为 TXDELETE，首先查找本地数据库，确定要删除的数据是否存在。如果存在则将 TXDELETE 的可链接环签名字段与对本地查找到的签名信息进行环签名的链接操作，若链接成功则说明 TXDELETE 交易的发送者是数据库中该条数据的拥有者，执行删除操作。

4.3.2　改进的可链接环签名算法

原始可链接环签名算法在相同公钥列表下只能生成带有相同链接标记的签名结果，无法应用在本方案中。主要有两点原因：首先，如果同一用户生成的签名结果只能有一种链接标记，那么攻击者可以根据此链接标记定位用户所有签名，从而分析用户行为，这样会大大降低用户匿名性。其次，改进的可链接环签名可以很容易匿名实现文件所有权的鉴权。原理是：针对某一次共享过程，用户对其进行签名并附加一个独特的链接标记，在确定性删除阶段发送带有同样链接标记的 TXDATA 交易，区块链节点和 CSP 即可确定用户对该数据的所有权并执行删除操作。如果只能生成一种链接标记，就无法指定对哪一次共享进行确定性删除。具体改进算法，包括签名算法、签名验证算法和签名链接算法。

令 $G(q)=<g>$ 为 q 阶循环群且满足离散对数安全假设。令 $H_1:\{0,1\}^*\to\mathbb{Z}_q$ 且 $H_2:\{0,1\}^*\to G$ 为统计独立的加密散列函数。随机选择 $x_\pi\in\mathbb{Z}_q$ 作为环签名私钥，$y_\pi=g^{x_\pi}$ 作为环签名公钥。所有用户将自己的公钥发送到公钥池中得到环签名公钥列表 $L_2=\{y_1,\cdots,y_n\}$。

- 签名算法 $\mathrm{LSAGSig}(L_2,m,r,y_{\pi,},x_\pi)\to\sigma_r m$：算法输入为消息 $m\in\{0,1\}^*$，环签名公钥列表 $L_2=\{y_1,\cdots,y_n\}$，y_n 对应的私钥$(1\leqslant\pi\leqslant n)$和链接参数 ，根据链接参数计算 $h=H_2(r)$ 和 $\tilde{y}=h^{x_\pi}$，得到链接标记。接下来选择 $u\in_R\mathbb{Z}_q$，并计 $c_{\pi+1}=H_1L_2,\tilde{y},m,g^u,h^u$，为使签名闭合成环，令 $i=\pi+1,\cdots,n,1,\cdots,\pi-1$，选择 $s_i\in_R\mathbb{Z}_q$ 并计算 $c_{i+1}=H_1(L_2,\tilde{y},m,g^{s_i}\tilde{y}_i^{c_i}$，$h^{s_i}\tilde{y}^{c_i})$。最后计算 $s_\pi=u-x_\pi c_\pi \bmod q$. 得到并输出可链接环签名结果 $\sigma_r m=(c_1,s_1,\cdots,s_n,\tilde{y},h)$。
- 签名验证算法 $\mathrm{LSAGVer}(L_2,m,\sigma_r m)\to$ Accept or Reject：算法输入为消息 m，环签名公钥列表 L_2 以及消息 m 在 L_2 上的环签名 $\sigma_r(m)$。令 $i=1,\cdots,n$，首先计算中间变量 $z_i'=g^{s_i}y_i^{c_i}$，$z_i''=h^{s_i}\tilde{y}^{c_i}$，根据中间变量计算 $c_{i+1}=H_1(L_2,\tilde{y},m,z_i',z_i'')$，$(i\neq n)$。最后验

证 $c_1 = H_1(L_2, \tilde{y}, m, z'_n, z''_n)$ 是否成立，若成立则输出 Accept，否则输出 Reject。

- 签名链接算法 LSAGLink$(\sigma'_r(m)', \sigma''_r(m)'')$ → True or False 算法输入为两个合法的环签名，设为 $\sigma'_r m' = (c'_1, s'_1, \cdots, s'_n, \tilde{y})$ 和 $\sigma''_r(m'') = (c''_1, s''_1, \cdots, s''_n, \tilde{y}'')$，一个公众验证者检查链接标记 $\tilde{y}' = \tilde{y}''$ 是否成立，若成立则说明两个签名是同一签名者针对同一链接参数 r 生成，算法输出 True；否则，两个签名是不同签名者生成，或者同一签名者用不同链接参数 r 生成，算法输出 False。

5 安全性分析

5.1 改进的可链接环签名安全性分析

定理 1 改进的可连接环签名不改变原始方案的匿名性。

证明 假设 L 为可链接环签名公钥列表，m 为待签名消息，r 为改进方案中的链接参数。原方案中，签名算法第一步计算 $h = H_2(L)$ 和 $\tilde{y} = h^{x_\pi}$，最终生成签名 $\sigma_r(m) = (c_1, s_1, \cdots, s_n, \tilde{y})$。改进方案中，签名算法第一步计算 $h = H_2(r)$ 和 $\tilde{y} = h^{x_\pi}$，最终生成签名 $\sigma_r(m) = (c_1, s_1, \cdots, s_n, \tilde{y}, h)$。对比原方案与改进方案可知，原方案中 L 为公钥列表，这是所有环内成员已知条件，即 h 为环内成员已知条件，所以原方案生成签名等价于 $\sigma_r m = (c_1, s_1, \cdots, s_n, \tilde{y}, h)$，与改进方案相同。所以，改进方案中将 $h = H_2(r)$ 加入签名结果使环内成员可见，相比较原方案没有透露更多信息。综上所述，改进的可链接环签名方案不改变原方案匿名性。

定理 2 除了签名者，其他人不能伪造带有相同链接标记的签名。

证明 要生成链接标记，首先通过单向哈希函数计算 h，再计算 $\tilde{y} = h^{x_\pi}$。所以要想伪造相同链接标记，必须解决离散对数问题，计算出签名者私钥，才能伪造消息 m 的签名。而目前没有有效的算法在多项式时间内解决离散对数问题，所以只有签名者可以生成同样链接标记的环签名。

5.2 方案匿名性分析

（1）区块链交易信息对匿名性影响

区块链节点和 CSP 通过验证交易信息的签名字段验证交易发出者身份的合法性，不涉及交易发出者的真实身份，切断了用户数据与其身份的关联性。由于可链接环签名的匿名性可知，区块链节点之间是相互匿名的，节点对于 CSP 来说也是匿名的。假设攻击者想要跟踪某一用户的共享和删除行为，需要破解区块链上记录的环签名信息（第 5.1 节已经证明签名方案的安全性）。攻击者还可以根据链接标记追踪用户行为，但是用户的每次共享行为都会随机选择链接参数生成不同链接标记，除非攻击者窃取到签名者私钥，否则系统是安全的，能有效防止攻击者的追踪分析。

（2）安全信道对匿名性影响

方案中数据使用者和数据拥有者与 CSP 之间会使用安全信道进行数据传输，安全信道的建立可能会涉及用户 IP 地址，但是用户 IP 地址不能代表用户身份。首先，用户 IP 地址通常由运营商进行分配，所以会经常变动。其次，在大多数情况下一个公网 IP 地址对应的是一个用户群体而不是单独一个用户。在安全信道进行数据传输之前，通信双方可以先进行密钥协商，利用协商的密钥加密传输的数据，这个过程可以不涉及用户身份。综上所述，安全信道不会降低用户匿名性。

5.3 不可抵赖性分析

在所提方法中，区块链网络不仅负责密钥信息的分散存储，还起到验证操作者身份的作用，用户的所有操作都会经验证后记录在区块链中并且不会被篡改。不论是数据上传、数据请求还是数据确定性删除都需要操作者向区块链发送相应类型的交易，CSP 监听到交易信息才会对用户数据进行相应操作。如果某节点试图篡改区块链数据，则其需要从该区块开始计算哈希值直到最新区块使区块之间通过哈希值连接，这需要非常强

大的算力，通常需要全网一半以上的算力才能做到如此（51%攻击）。

如上所述，区块链的难以篡改性，使链上数据具有权威性，任何一方都不能抵赖。如果 CSP 没有诚实删除用户数据，则用户可以将环签名链接参数 r 和签名私钥 x_π 交给第三方仲裁机构，仲裁机构根据链接参数计算 $h=H_2(r)$ 和 $\tilde{y}=h^{x_\pi}$，将得到的链接标记 $\tilde{y}$ 与出现纠纷交易的签名进行对比，如果链接标记相同，说明用户确实执行过删除操作，CSP 无法抵赖。

6　实验仿真

6.1　实验环境

本文采用阿里云服务器模拟云服务提供商（CSP），用本地计算机运行区块链节点（包括数据拥有者和数据使用者），其中阿里云服务器 CPU 为 2 核心，内存为 4 GB，操作系统为 CentOS 8.0 64 位，网络带宽为 1 Mbit/s；本地计算机配置为 Intel Core(TM) i5-10210U CPU @ 1.60 GHz 四核，设备内存为 12 GB，操作系统为 Windows 1 064 位，网络带宽为 100 Mbit/s。方案中使用的 AES 算法、RSA 算法和数据摘要算法采用 Python 加密库 PyCryptodome 提供的 API 函数。区块链采用基于 RPC[18]（remote procedure call，远程过程调用）实现的开源项目[19]。数据拥有者或数据使用者与 CSP 之间的数据传输采用 FTP 进行模拟，CSP 对区块链数据的监听通过调用区块链的接口实现。

6.2　实验及结果分析

实验主要对方案的 3 个阶段耗时情况进行了实验评估。实验选择 Shamir 秘密分享门限 t 为 $0.6n$，选取 AES 密钥长度为 128 bit，RSA 模数长度为 1 024 bit。

（1）数据上传阶段和数据确定性删除阶段性能分析

数据上传阶段中，数据拥有者和数据使用者与 CSP 之间的数据传输速率取决于双方带宽的最小值，可直接计算得出，所以本实验没有统计文件上传耗时，即实验统计的时间是从数据拥有者发出存储请求到数据拥有者开始向 CSP 上传文件。系统主要时间消耗为 Shamir 秘密拆分算法、交易发送算法 SendTX 和交易处理算法 HandleTX，SendTX 算法主要包含 RSA 加密和签名算法，HandleTX 算法主要包含 RSA 解密算法和签名验证算法。表 2 为数据上传阶段各个算法耗时占比情况（“其他”指代数据拥有者向 CSP 发送请求、数据库存储和区块链网络通信等开销），可见数据上传阶段主要时间开销来源是 RSA 算法和签名（改进的可链接环签名）算法。同时由图 7 可见，随着节点数量的增加，系统耗时呈现线性增长的趋势，这是因为签名的生成和验证是与环内成员（区块链节点）数量线性相关的，RSA 加解密次数也是与节点数量呈线性相关的。

表 2　数据上传阶段耗时占比详情

上传阶段	节点数量								平均
	20	30	40	50	60	70	80	90	
秘密拆分	0.46%	0.30%	0.23%	0.19%	0.15%	0.13%	0.11%	0.10%	0.21%
RSA 加密	9.25%	9.42%	9.33%	9.67%	9.61%	9.30%	17.33%	17.33%	11.41%
RSA 解密	9.52%	9.53%	9.57%	9.24%	9.25%	9.27%	8.77%	8.77%	9.24%
签名	50.88%	50.41%	49.46%	49.73%	49.10%	49.04%	43.02%	42.42%	48.01%
验证签名	27.86%	27.23%	26.97%	26.07%	25.70%	24.97%	22.38%	22.31%	25.44%
其他	2.02%	3.11%	4.44%	5.10%	6.19%	7.30%	8.38%	9.06%	5.70%

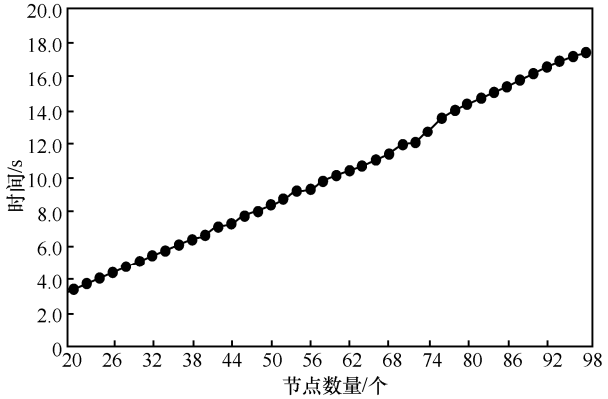


图 7　数据上传阶段耗时曲线

（2） 数据请求阶段性能分析

实验统计数据请求阶段时间是从数据使用者发送 TXREQUEST 交易到接收到 CSP 返回的密文结束。期间各个算法耗时占比见表 3。可见 RSA 解密耗时占比非常大，其原因是，当区块链接收数据使用者发送的 TXREQUEST 交易后还需要返回 t 个 TXDATA 交易，而针对这 t 个交易数据使用者要逐个解密这些交易的 Payload 字段。从图 8 也可以看出，由于 RSA 解密阶段的影响，整个数据请求阶段耗时远远高于数据上传阶段。这个问题是可以改善的，由于此解密过程在数据使用者本地计算机进行，可以用计算机编程的多线程思想解决，即对 t 个秘密并行解密，优化后与数据上传阶段耗时是相近的。

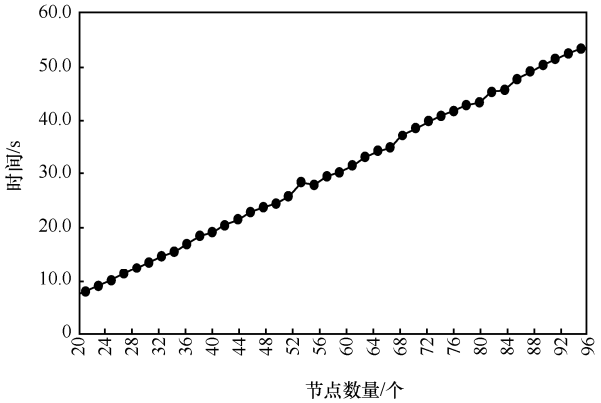


图 8　数据请求阶段耗时曲线

（3）数据确定性删除阶段性能分析

实验统计数据确定性删除阶段时间是从数据拥有者发送 TXDELETE 交易到云端从数据库中

表 3　数据请求阶段耗时占比

请求阶段	节点数量								平均
	20	30	40	50	60	70	80	90	
RSA 加密	3.63%	2.69%	2.09%	1.84%	1.55%	1.31%	1.28%	1.17%	1.94%
RSA 解密	72.32%	79.23%	83.20%	84.48%	86.14%	87.40%	88.13%	88.79%	83.71%
签名	10.26%	7.51%	5.88%	5.06%	4.32%	3.80%	3.21%	2.87%	5.36%
验证签名	11.18%	8.04%	6.33%	5.24%	4.45%	3.81%	3.27%	2.95%	5.66%
秘密重构	0.18%	0.09%	0.05%	0.03%	0.02%	0.02%	0.01%	0.01%	0.05%
其他	2.43%	2.44%	2.44%	3.34%	3.52%	3.66%	4.10%	4.21%	3.27%

表 4　数据确定性删除阶段耗时占比

确定性删除阶段	节点数量								平均
	20	30	40	50	60	70	80	90	
RSA 加密	12.61%	12.83%	12.70%	13.21%	13.14%	12.78%	13.97%	13.95%	13.15%
RSA 解密	12.97%	12.98%	13.02%	12.63%	12.65%	12.73%	11.63%	11.62%	12.53%
签名	35.63%	35.82%	35.77%	36.43%	36.53%	37.19%	36.90%	36.78%	36.38%
验证签名	34.55%	34.11%	34.11%	33.06%	33.00%	32.60%	32.80%	32.92%	33.39%
其他	4.25%	4.26%	4.40%	4.66%	4.68%	4.70%	4.70%	4.72%	4.55%

删除加密数据结束。此阶段与数据上传阶段类似，主要时间开销为 RSA 算法和签名算法（见表 4），并且随着节点数量增加，时间消耗呈线性增长（如图 9 所示）。

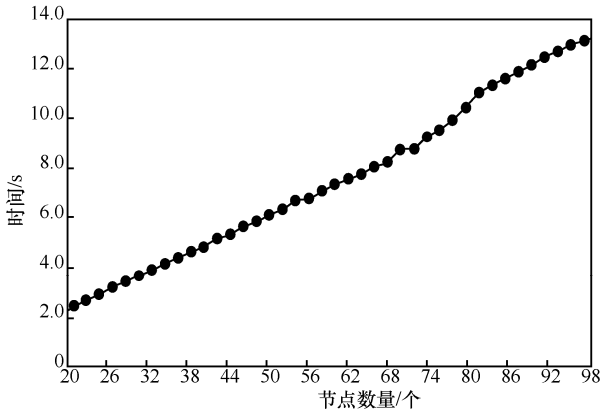


图 9　数据确定性删除阶段耗时曲线

7　结束语

本文在现有确定性删除方法的基础上做了进一步的探索，提出切断用户数据与用户身份关联的确定性删除理念，并提出了基于区块链的匿名确定性删除方法。方法在区块链中构建不同类型的交易实现用户对数据密钥的隐藏、请求和删除，并应用改进的可链接环签名技术实现了云数据的匿名确定性删除，进一步提高了用户使用云计算服务时的隐私性。最后对方案性能进行了评估，结果表明系统能实现较高效率的数据上传、数据请求和数据确定性删除。在未来研究中，会在方案中加入节点动态加入和退出功能，并进一步提高系统效率。

参考文献：

[1] 熊金波, 李凤华, 王彦超, 等. 基于密码学的云数据确定性删除研究进展[J]. 通信学报, 2016, 37(8): 167-184.
XIONG J B, LI F H, WANG Y C, et al. Research progress on cloud data assured deletion based on cryptography[J]. Journal on Communications, 2016, 37(8): 167-184.

[2] NAIR S K, DASHTI M T, CRISPO B, et al. A hybrid PKI-IBC based ephemerizer system[C]//Proceedings of the International Information Security Conference. [S.l.:s.n.], c2007: 241-252.

[3] TANG Y, LEE P P, LUI J C, et al. FADE: secure overlay cloud storage with file assured deletion[C]//Proceedings of the Security and Privacy in Communication Networks (SecureComm). [S.l.:s.n.], c2010: 380-397.

[4] GEAMBASU R, KOHNO T, LEVY A, et al. Vanish: increasing data privacy with self-destructing data[C]//Proceedings of the 18th USENIX Security Symposium. [S.l.:s.n.], c2009: 299-315.

[5] DABEK F. A distributed hash table[D]. Massachusetts: Massachusetts Institute of Technology, 2005.

[6] WOLCHOK S, HOFMANN O S, HENINGER N, et al. Defeating vanish with lowcost sybil attacks against large DHTs[C]//Proceedings of the 17th Annual Network & Distributed System Security Conference (ISOC NDSS). [S.l.:s.n.], c2010: 1-15.

[7] ZENG L, SHI Z, XU S, et al. SafeVanish: an improved data self-destruction for protecting data privacy[C]//Proceedings of the IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom). Piscataway: IEEE Press, 2010: 521-528.

[8] 熊金波, 沈薇薇, 黄阳群, 等. 云环境下的数据多副本安全共享与关联删除方案[J]. 通信学报, 2015, 36(S1): 136-140.
XIONG J B, SHEN W W, HUANG Y Q, et al. Security sharing and associated deleting scheme for multi-replica in cloud[J]. Journal on Communications, 2015, 36(S1): 136-140.

[9] ZHANG M, ZHANG H, YANG Y, et al. PTAD: provable and traceable assured deletion in cloud storage[C]//Proceedings of the 2019 IEEE Symposium on Computers and Communications (ISCC). Piscataway: IEEE Press, 2019.

[10] 杜瑞忠, 石朋亮, 何欣枫. 基于覆写验证的云数据确定性删除方案[J]. 通信学报, 2019, 40(1): 130-140.
DU R Z, SHI P L, HE X F. Cloud data assured deletion scheme based on overwrite verification[J]. Journal on Communications, 2019, 40(1): 130-140.

[11] 余海波. 基于区块链的数据分布式存储安全机制研究[D]. 上海: 华东师范大学, 2020.
YU H B. Research on security mechanism of data distributed storage based on blockchain[D]. Shanghai: East China Normal University, 2020.

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

[13] LIU J K, WEI V K, WONG D S. Linkable spontaneous anonymous group signature for Ad Hoc groups[C]// Proceedings of the Australasian Conference on Information Security and Privacy. Heidelberg: Springer, 2004.

[14] NAKAMOTO S. Bitcoin: a peer-to-peer electronic cash system[Z]. 2008.

[15] 李董, 魏进武. 区块链技术原理、应用领域及挑战[J]. 电信科学, 2016, 32(12): 20-25.

LI D, WEI J W. Theory, application fields and challenge of the blockchain technology[J]. Telecommunications Science, 2016, 32(12): 20-25.

[16] LI G, SATO H. A privacy-preserving and fully decentralized storage and sharing system on blockchain[C]//Proceedings of the 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC). Piscataway: IEEE Press, 2019.

[17] 曹景源, 李立新, 李全良, 等. 云存储环境下生命周期可控的数据销毁模型[J]. 计算机应用, 2017, 37(5): 1335-1340.
CAO J Y, LI L X, LI Q L, et al. Data destruction model for cloud storage based on lifecycle control[J]. Journal of Computer Applications, 2017, 37(5): 1335-1340.

[18] 周亮, 王震, 王冠. 远程过程调用技术在分布式软件系统中的应用[J]. 航空电子技术, 2020, 51(4): 47-52.
ZHOU L, WANG Z, WANG G. Application of remote procedure calling technology in distributed software system[J]. Avionics Technology, 2020, 51(4): 47-52.

[19] Carlos-Zen, blockchain-python[EB]. 2018.

[作者简介]

王双星（1996− ），男，北京科技大学计算机与通信工程学院硕士生，主要研究方向为区块链技术与隐私保护。

罗劲瑭（1989− ），男，博士，国网四川省电力公司经济技术研究院研究员，主要研究方向为数据中心网络中的拥塞控制和信息安全。

帅莉莎（1995− ），女，北京科技大学计算机与通信工程学院博士生，主要研究方向为移动人群感知和互联网安全。

张佳敏（1992− ），女，北京科技大学计算机与通信工程学院博士生，主要研究方向为网络和信息安全、隐私保护。

张敏（1972− ），女，北京科技大学计算机与通信工程学院副教授，主要研究方向为新一代互联网技术、内容分发网络、智能电网 CPS 安全等。

阳小龙（1970− ），男，博士，北京科技大学计算机与通信工程学院教授、博士生导师，主要研究方向为 IP 路由与交换、云计算数据/内容中心网络、网络安全等。