文章编号:1671-6833(2024)05-0095-08

基于区块链智能合约的异构服务器安全去重方法

江 粼[1], 李嘉兴[2], 武继刚[1]

(1.广东工业大学 计算机科学与技术学院,广东 广州 510006;2.香港理工大学 人工智能设计研究所,香港 999077)

摘 要:针对大数据时代用户数据在云服务器存储中面临的可靠性提升与重复数据删除策略之间的冲突,提出了一种基于区块链智能合约的异构服务器数据安全去重方法,利用区块链的去中心化、不可篡改和公开透明等特性,以及智能合约的自动化执行能力,实现了数据存储的安全性、可靠性和隐私保护。具体而言,方法结合了秘密共享和区块链智能合约技术,设计了安全高效的云存储数据去重服务。同时,通过区块链取代集中式第三方实体的功能,消除了潜在的安全隐患,并通过智能合约脚本缓解了服务器之间的异构性。实验结果表明:研究方法在相同文件大小、不同文件块数量的情况下的平均计算开销比对比方法低 65.42%~115.77%,平均储存开销降低 7.94%~19.50%。同时,在不同异构存储服务器数量下,平均计算开销与存储开销分别降低了 67.27%~177.89%、34.01%~72.89%。研究方法在安全性、计算开销及存储开销方面优于现有的两个基于区块链的数据去重方法。

关键词:区块链; 云存储; 智能合约; 秘密共享方法; 数据去重; 安全性

中图分类号:TP309.2;TP309.3 **文献标志码**:A **doi**:10.13705/j.issn.1671-6833.2024.02.010

随着云计算技术的迅速发展与用户数据的爆发式增长,云存储系统的重要性日益凸显[1]。数据去重技术对云存储系统具有重要意义。研究表明,在云存储系统中,有超过一半的数据容量是由于重复存储引起的。有效的数据去重技术不仅能降低存储冗余,减少用户上传带宽消耗,还能节省服务器端存储空间[2-3]。然而,数据去重不仅仅是删除数据,它需要在最小存储空间内尽可能多地存储文件,并在需要时能够恢复原始数据。因此,传统的数据加密技术与云存储数据去重技术之间存在冲突,这使得在密文环境中设计高效的数据安全去重技术成为一项重要挑战[4]。

在实际应用中,基于密文的数据去重可以有效降低用户端上传数据耗费的带宽以及节省服务器端的存储空间,消除冗余的数据,同时维护用户隐私[5]。然而,当前的数据去重技术仍面临许多困难。云存储系统通常采用差异加密以保护用户数据,但现有方法未提供可靠高效的解决方法,因此,需要引入可信密钥服务器生成安全标签。然而,这带来了潜在的安全风险,一旦密钥服务器遭攻击,用户的数据和密钥可能泄露。此外,多样的云存储服务器和异构数据带来通信易遭攻击的问题。数据通常分布在多个不同服务器上,这影响了数据存储和去重方式。一个去重后的数据副本可能被不同文件引用,其丢失会对拥有关联文件的众多用户造成不利影响,威胁外包数据的可用性和可靠性。此外,备份数据的存储和安全性,尤其是在考虑自然灾害和人为损害等风险时,也是一个重要问题[6]。解决这些问题对确保云存储系统的数据安全至关重要[7-9]。

针对这些问题,本研究提出了一种基于区块链智能合约的异构服务器数据安全去重方法。该方法结合了区块链技术和确定性秘密共享方法,旨在确保数据的安全性、可恢复性和去重效率。区块链可以安全地存储数据去重过程中的重要信息,例如安全标签、纠错码和文件指纹信息。即使某服务器上的标签被伪造,其他服务器可以通过区块链信息进行识别和恢复。此外,为了去除第三方实体带来的安全隐患,基于区块链的智能合约能够保障去重操作的安全性,所有操作都将由智能合约自动执行,并

收稿日期:2023-11-06;**修订日期**:2023-12-18

基金项目:国家自然科学基金资助项目(62072118, 62302112, 62106052)

通信作者:武继刚(1963—),男,江苏徐州人,广东工业大学教授,博士,博士生导师,主要从事移动智能计算、数据科学与云计算等研究,E-mail: asjgwucn@outlook.com。

引用本文:江粼,李嘉兴,武继刚. 基于区块链智能合约的异构服务器安全去重方法[J]. 郑州大学学报(工学版),2024,45(5):95-102,142. (JIANG L,LI J X,WU J G. Secure deduplication method with blockchain-based smart contract for heterogeneous cloud servers[J]. Journal of Zhengzhou University(Engineering Science), 2024,45(5):95-102,142.)

记录在区块链上。因此,区块链智能合约技术为异构存储服务器的大数据安全去重提供了重要方向。本研究的贡献包括以下 3 个方面。

(1)提出了一种基于区块链智能合约的异构服务器数据去重方法,以实现安全、高效的云存储数据安全去重服务。

(2)利用双线性对生成秘密份额,并将确定性秘密共享方法与基于区块链的智能合约结合,确保云存储数据去重的安全性。

(3)设计区块链取代集中式第三方实体功能,从而消除其所带来的安全隐患,并通过设计智能合约脚本缓解服务器之间的异构性。

1 相关工作

1.1 传统数据去重方法

传统的数据去重方法通常涉及数据副本的存储,以确保在硬件或软件故障导致数据丢失或损坏时能够进行恢复。然而,传统的加密算法可能不再适用于云存储系统的安全去重方法,因为不同用户的数据使用不同的密钥进行加密,导致相同数据生成不同密文,从而增加了存储冗余。为解决这一问题,Douceur 等[10]提出了一种收敛加密(convergent encryption,CE)方法,该方法使用数据的哈希值来生成加密密钥。此外,Bellare 等[6]在 2013 年提出了一种支持重复数据删除的消息锁定加密(message-locked encryption,MLE)方法,其中加密和解密的密钥来自文件本身。为了提高 MLE 的效率,Chen 等[11]实现了大文件的块级重复数据删除。然而,上述方法只支持静态数据的冗余删除,无法满足动态数据去重的需求。因此,Ding 等[12]构建了一种具有同态重加密的用户可识别重复数据删除方法,而基于属性的加密是 Cui 等[13]提出的另一种在重复数据删除系统中广泛使用的数据共享方法。尽管上述研究可以提供机密性和完整性,但仍然无法为外包数据提供足够的可靠性。

1.2 基于区块链智能合约的去重方法

区块链作为密码学、共识机制、智能合约等多种可靠技术有机组合成的集成系统,在云计算系统等相关领域上具有广阔的发展前景[14-15]。如图 1 所示,区块链由区块和哈希链组成,每个区块按照时间顺序由哈希链单向串联起来,具有公开透明、不可篡改、可追溯等特点。智能合约通过在区块链上创建一个可编程的、自动化的执行环境,能够让合约脚本程序的执行过程更加安全、可靠和高效。例如,Huang 等[16]提出的 seShare 方法结合了区块链和安全重复数据删除,但受信任的第三方容易成为单点故障。Zhang 等[17]提出了一种基于区块链的安全授权去重方法,该方法利用智能合约创建防篡改账本并通过智能合约检查用户数据的完整性。Huang 等[18]将区块链技术引入到安全重复数据删除的场景中,并构建了特定的重复数据删除方法来解决恶意用户追踪问题。尽管这些方法取得了一些成果,但仍有改进的空间,特别是在如何利用区块链替代集中式第三方实体、如何处理云存储服务器之间的异构性,以及如何融合现有云存储去重技术中的密码学方法和区块链技术等方面。本文从上述改进空间的角度出发,提出了一种基于区块链智能合约的异构服务器数据去重方法,这对于大数据环境下云存储系统的性能和安全性来说具有重要的研究意义。

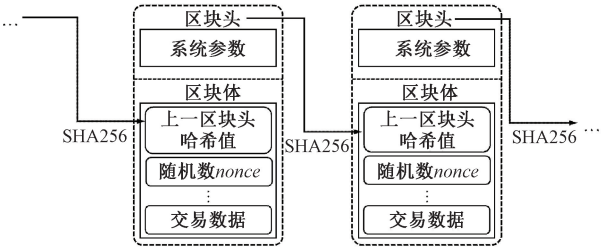

图 1 区块链结构图

Figure 1 Structure diagram of the blockchain

2 基于区块链智能合约的异构云服务器安全去重方法

2.1 初始化

首先,令 G 和 G_T 为两个以素数 p 为阶、g 为生成元的乘法循环群,选择满足条件 $e:G\times G\rightarrow G_T$ 的双线性映射 e。在区块链网络中,任一用户 u 会随机选择 $sk_u \in \mathbf{Z}_p^*$ 作为私钥,通过计算 $pk_u = sk_u g_1$ 得到公钥,其中 $g_1 \in G$,sk_u、$pk_u \in \mathbf{Z}_p$。其次,选取一个具备抗碰撞性的哈希函数 $H:\{0,1\}^* \rightarrow G$。再次,$u$ 选取一个机密的随机数 $\lambda \in \mathbf{Z}_p$,并生成一个用于区块链网络中匿名通信的私有地址 $Addr = H(pk_u \| \lambda)$。最后,网络中每个参与节点的初始化系统参数可以定义为 $Params = \{G, G_T, p, g, e, H, Addr\}$。

2.2 本文方法

2.2.1 概述

从去重细粒度区分,数据去重技术分为块级去重和文件级去重。一般而言,基于数据块级的数据去重能获得较高的压缩率,也是目前为止使用最广泛的数据去重技术。然而,文件块划分得越小,提高去重效率的同时会产生大量的密钥,密钥数目随着上传数据块的增长而呈线性相关增长。因此,合适的文件块大小对数据去重的效率具有重要意义。

然而,在对数据去重的同时需要考虑数据的重建,虽然文件的部分内容被删除,但当需要取回文件时仍然能将完整的文件内容重建出来,这就需要保留文件与唯一数据单元之间的索引信息,即元数据。一旦元数据被敌手篡改,存储的文件将不能还原。因此,元数据可以保存在区块链中,即使敌手成功攻破一个服务器,也不能成功地篡改文件的元数据,甚至被恶意删除的文件也可以根据元数据通过其他服务器中保存的副本重建数据,从而保证了数据去重的安全。此外,数据在不同服务器之间的存储方式不尽相同,基于区块链的智能合约可以通过虚拟机执行合约,缓解异构服务器之间的底层架构差异,消除异构服务器对数据去重的阻力。

如图 2 所示,为了保证云存储机密性的同时提高可靠性,本文提出了一种将秘密共享方法与基于区块链的智能合约相结合的新型重复数据删除方法。该方法是将用户数据拥有者(data owner,DO)存储在云端的数据分割成块并存储在多个异构的分布式云服务器之间,只需要一定数量的数据块就可以恢复整个文件。此外,区块链技术被用于支持可靠的去重流程记录和基于智能合约的验证。因此,不用第三方参与,本文方法也能够保障异构云存储服务器上数据去重的安全性。

现实中,CSP 部署的存储服务器一般都是性能相近的,但考虑到服务器维护等因素,每个服务器在线的时间可能相差较远。因此,时间决定出块权的共识机制,如权益证明(proof-of-stake, PoS)并不适合本文中的服务器部署环境,而算力决定出块权的工作量证明(proof-of-work, PoW)共识机制将更适合本文方法。

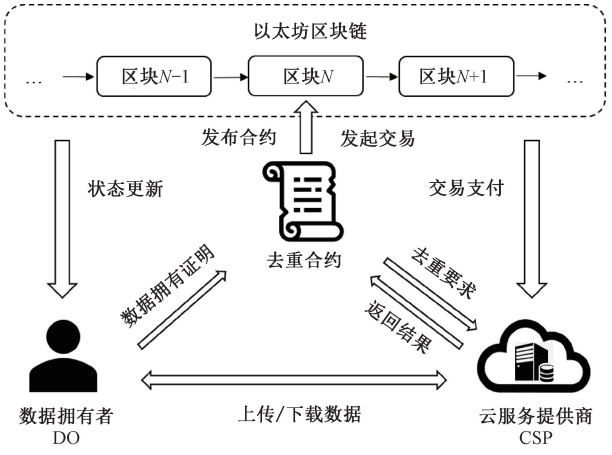

图 2　基于区块链的数据去重方法系统架构图

Figure 2　Framework of secure deduplication method with blockchain-based smart contract for heterogeneous cloud servers

2.2.2　方法具体流程

本文方法的流程图如图 3 所示。

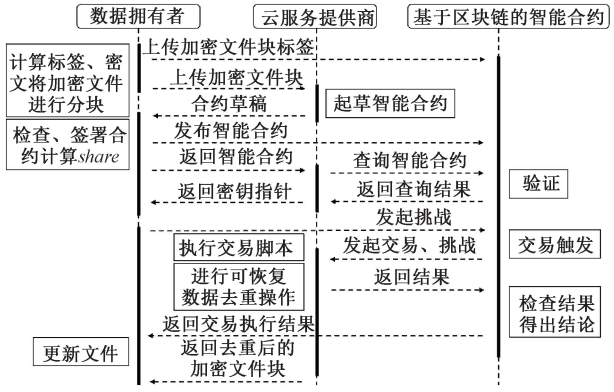

图 3　基于区块链智能合约的异构服务器安全去重流程示意图

Figure 3　Flowchart of secure deduplication method with blockchain-based smart contract for heterogeneous cloud servers

首先,用户 u 对其文件 F 生成标签 $tag(F) \leftarrow TagGen(F)$ 和收敛加密密钥 $K \leftarrow TagGen(F)$。然后 u 对文件 F 进行加密得到密文 $C \leftarrow Encrypt(F, K)$。接着,用户 u 把密文进行分块得到集合 $\{c_1, c_2, \cdots, c_l\}$ 并生成相对应的标签 $\{tag(c_1), tag(c_2), \cdots, tag(c_l)\}$。为把密文数据块存储到多个服务器中,用户 u 为每一个数据块计算 $\{c_{i_q}\} \leftarrow Share(c_i)$。DO 接收并更新本地区块链上文件 F 的标签数据集 $\{tag(F'), tag(C'), tag(c'_{i_q})\}$,同时通过 $\exists tag(F) = tag(F''), tag(F'') \in \{tag(F')\}$ 检查标签 $tag(F)$ 是否存在,以进行文件级的数据重复检测。如果文件级的去重检测未通过,则需要对 F 中所有数据块进行重复检测。如果文件级的去重检测通过了,则通过 $\exists tag(c_{i_q}) = tag(c'_{i_q}), tag(c'_{i_q}) \in \{tag(c'_{i_q})\}$ 继续对 $\{c_{i_q}\}$ 进行文件块级的重复数据检测。如果区块链上的记录包含了 c_{i_q} 的记录,则只需要通过计算式(1)并根据式(1)计算式(2)以获得所有权证明和存储服务器的身份识别 $h_{f_{c_i}}$。

$$pow_u = H(c_{i_q} \| pk_u \| nonce) sk_u。 \tag{1}$$

$$h_{f_{c_i}} = \{tag(F), tag(c_i), tag(c_{i_q}), pow_u, pk_u, Addr, nonce\}。 \tag{2}$$

此外,根据式(1),u 还需要通过式(3)为选中的服务器计算身份识别 $h'_{f_{c_i}}$。

$$h'_{f_{c_i}} = \{c_{i_q}, tag(F), tag(c_i), tag(c_{i_q}), pow_u, pk_u, nonce\}。 \tag{3}$$

如果区块链上没有 c_{i_q} 的记录,则认为该数据块被篡改或丢失了。在上传文件之前,u 使用 $\{H(c_{i_q}) \| v\}$

作为检索证明,其中,v 是随机选取的且 $v \in \mathbf{Z}_p^*$。

云存储服务器在接收到 c_{i_q} 后,通过验证式(4)是否成立来验证其所有权。

$$e(H(c_{i_q} \| pk_u \| nonce), pk_u) = e(pow_u, p)。 \quad (4)$$

如果验证失败,则 u 将被该存储服务器拒绝访问。如果验证成功,则第 j 个服务器 s_j 通过计算公式(5)

$$pow_s = H(c_{i_q} \| pk_{s_j} \| nonce) sk_{s_j}。 \quad (5)$$

返回元组 $\{pow_s, tag(c_{i_q}), pk_{s_j}, nonce\}$、一个指向文件 F 的文件指针,以及一个智能合约脚本的草稿 $Script = \{blank \| sign_{s_j}(Script) \| verify \| price\}$ 给 u。其中,$blank$ 为留给 u 数字签名的空白字段;$verify$ 是一个在区块链上实现的用于自动支付的智能合约算法。具体地,$verify$ 算法从 u 接收验证信息 $E(H(c_{i_q} \| v), sk_u)$ 和从 s_j 接收 $E(H(c'_{i_q} \| v), sk_{s_j})$,并分别使用对应的公钥 pk_u 和 pk_{s_j} 进行解密。$price$ 为 u 和 s_j 双方商定好的以太币(ETH)数量,需要双方从私人账户转移到该合约上作为押金。一旦合约执行,将从参与双方的账户中扣除,直到合约结束。

最终,$verify$ 算法通过式(6)进行判断并返回 $trigger$ 的值。

$$trigger = \begin{cases} 1, & H(c_{i_q} \| v) = H(c'_{i_q} \| v); \\ 0, & 其他。 \end{cases} \quad (6)$$

当 $trigger$ 的值为 1 时,交易将被触发,交易中的智能合约脚本也将会被自动执行。此后,u 会收到服务器发送的 pow_s,然后将检查等式(7)是否成立。

$$e(H(c_{i_q} \| pk_{s_j} \| nonce), pk_{s_j}) = e(pow_s, p)。 \quad (7)$$

如果式(7)成立,u 将使用其数字签名 $sign_u(Script)$ 填补合约脚本中的空白字段($blank$),并发布该智能合约脚本 $SC = \{sign_u(Script) \| sign_{s_j}(Script) \| verify \| price\}$,其中 $sign_u(\cdot)$ 和 $sign_{s_j}(\cdot)$ 都是基于椭圆曲线签名算法(ECDSA)的签名函数。否则,u 将认为服务器 s_j 不可信,并选择其他服务器存储数据。ECDSA 算法可防止数据在传输过程中被篡改;其次,ECDSA 在公钥系统中是加密强度较强的一种算法,其基于椭圆曲线方程的性质生成密钥的方式,具有计算量小、处理速度快、存储空间和传输带宽占用少等特点,非常适用于本文中的应用场景;最后,ECDSA 已经被应用到很多基于区块链的系统中[19],其安全性能和效率已经被学术界和工业界广泛认可。

3 安全分析

3.1 威胁模型

为分析本文方法的安全性,分别从密钥安全性和区块链网络安全性两个方面进行分析,分别对应两个游戏 $game_1$ 和 $game_2$。本章节的概率优势分析适用于暴力破解和猜测攻击等,具体过程如下。

在 $game_1$ 的威胁模型中,挑战者$\mathcal{C}$会协助敌手$\mathcal{A}$通过对 Oracle 进行查询发起攻击。为了验证本文方法关于选择明文的安全性,假设 Oracle 会提供以下 3 种查询(guess)。

(1)$Oracle_1$:给定查询的信息,Oracle 会执行密钥算法并对相应的询问进行回答。

(2)$Oracle_2$:给定公开的 B 和 rB,Oracle 会输出整数 r。

(3)$Oracle_3$:给定哈希值 H,Oracle 会输出相应的输入 r。

基于上述威胁模型,基于以下定义对其概率优势进行分析。

定义 1 给定密文和基于 ECDLP 的密钥生成算法,本文方法对于自适应选择密文攻击具有不可区分性(IND-CCA),当且仅当$\mathcal{A}$在多项式时间内得到明文信息是困难的,即 $Adv_{\mathcal{A}}^{\text{IND-CCA}}(\xi_1) \leqslant \delta_1$,其中 ξ_1 是执行时间。对任意足够小的 δ_1,$Adv_{\mathcal{A}}^{\text{IND-CCA}}(\xi_1)$ 表示$\mathcal{A}$在给定 K 和 B 的情况下经过 q 次查询得到 $r \in \mathbf{Z}_q^*$ 的概率优势。

定义 2 给定椭圆曲线上一点 $K = rB$,ECDLP 问题在多项式时间内是困难的,即 $Adv_{\mathcal{A}}^{\text{ECDLP}}(\xi_2) \leqslant \delta_2$,其中 ξ_2 是执行时间。对任意足够小的 δ_2,$Adv_{\mathcal{A}}^{\text{ECDLP}}(\xi_2)$ 表示$\mathcal{A}$在给定 K 和 B 的情况下经过 q 次查询得到 $r \in \mathbf{Z}_q^*$ 的概率优势。

定义 3 给定哈希码 $H = Hash(r)$,在多项式时间内根据输出得到单向哈希函数的输入是困难的,即 $Adv_{\mathcal{A}}^{Hash}(\xi_3) \leqslant \delta_3$,其中 ξ_3 为执行时间。对任意足够小的 δ_3,$Adv_{\mathcal{A}}^{Hash}(\xi_3)$ 表示$\mathcal{A}$在给定单向哈希函数输出获得输入 $r \in \{0,1\}^*$ 的概率优势。

在 $game_2$ 威胁模型中,由于本文方法中区块链采用的共识机制是工作量证明(proof-of-work,PoW)机制,敌手$\mathcal{A}$可以通过伪造最长分叉作为有效的区块链。具体地,因为网络中所有区块链节点只认可最长分叉作为有效的区块链,同时在基于 PoW 机制的区块链系统中矿工节点发布区块的概率与矿工节点的计算能力成正比,因此$\mathcal{A}$可以在区块链网络中聚集 50%以上计算能力发起 51%攻击。

3.2 优势分析

基于上述威胁模型,对敌手能够发起的攻击进行了概率上的优势分析,包括定理 1 中证明本文方法能够抵抗基于 ECDLP 的密钥对生成的攻击,定理 2 中证明本文方法对选择明文攻击具有

不可区分性,和对本文方法中区块链网络的安全性分析。

定理 1　假设单向哈希函数能够近似地表现为一个对 ECDLP 困难问题的随机预言机,本文方法对于抵抗$\mathcal{A}$获得用户密钥对是可证明安全的。

证明　假设在对基于 ECDLP 困难问题的密钥系统攻击 $Attc_{\mathcal{A}}^{1}$过程中,$\mathcal{A}$是一个受限于多项式时间内产生随机数用于计算用户密钥对的敌手。挑战者$\mathcal{C}$通过哈希函数和密钥生成算法计算公钥,敌手$\mathcal{A}$根据对$\mathcal{C}$的查询计算用户的密钥对,其优势可以表示为 $Succ_{\mathcal{A}}^{1}=2Pr[Exp_{\mathcal{A}}^{1}=1]-1$。假设执行时间是 ξ_2,对 Oracle_2 和 Oracle_3 的查询次数分别是 q_2 和 q_3,则有 $Adv_{\mathcal{A}}^{\text{Hash, ECDLP}}(\xi_2,q_1,q_2)=\max Succ_{\mathcal{A}}^{1}$。因此,对于任意足够小的 $\delta_2>0$,如果 $Adv_{\mathcal{A}}^{\text{Hash,ECDLP}}(\xi_2,q_2,q_3)\leqslant\delta_2$ 成立,则本文方法在随机 Oracle 模型中抵抗 $Attc_{\mathcal{A}}^{1}$ 是可证明安全的。

考虑到 $Attc_{\mathcal{A}}^{1}$在本文方法中如果能够在多项式时间内得到单向哈希函数的输入,并解决 ECDLP 困难问题,$\mathcal{A}$能够根据用户公钥成功地生成密钥对的哈希值。然而,根据定义 2 和定义 3,对于任意足够小的 δ_3 和 δ_4,$Adv_{\mathcal{A}}^{\text{ECDLP}}(\xi_2)\leqslant\delta_3$ 和 $Adv_{\mathcal{A}}^{\text{Hash}}(\xi_3)\leqslant\delta_4$ 都能够成立。因此,$Adv_{\mathcal{A}}^{\text{Hash, ECDLP}}(\xi_2,q_2,q_3)\leqslant\delta_2$ 也成立,因为 $Adv_{\mathcal{A}}^{\text{Hash,ECDLP}}(\xi_2,q_2,q_3)$ 依赖于 $Adv_{\mathcal{A}}^{\text{ECDLP}}(\xi_2)\leqslant\delta_3$ 和 $Adv_{\mathcal{A}}^{\text{Hash}}(\xi_3)\leqslant\delta_4$。此外,文献[20]也证明了在多项式时间内通过 K 和 G 计算得到 k 在概率上是不可能的,因此基于 ECDLP 困难问题的椭圆曲线算法目前仍然是安全的。综上所述,本文方法在多项式时间内对于抵抗 $\mathcal{A}$ 获得用户密钥的攻击是安全的。

证毕。

定理 2　假设单向哈希函数能够近似地表现为一个对 ECDLP 困难问题的随机预言机。

证明　假设在对选择明文攻击 $Attc_{\mathcal{A}}^{2}$过程中,$\mathcal{A}$是一个受限于多项式时间内从密文 e 中获取猜测明文 m' 的敌手。$Attc_{\mathcal{A}}^{2}$中 $\mathcal{A}$ 的成功概率可以表示为 $Succ_{\mathcal{A}}^{2}=2Pr[Exp_{\mathcal{A}}^{2}=1]-1$,其中如果 $m'=m$ 则 $Exp_{\mathcal{A}}^{2}=1$ 成立,反之亦然。假设执行时间为 ξ_1,对 Oracle_1 的查询次数为 q_1,则有 $Adv_{\mathcal{A}}^{\text{IND-CCA}}(\xi_1,q_1)=\max Succ_{\mathcal{A}}^{2}$。因此,对于任意足够小的 $\delta_1>0$,如果 $Adv_{\mathcal{A}}^{\text{IND-CCA}}(\xi_1,q_1)\leqslant\delta_2$ 成立,则本文方法对于选择明文攻击是安全的。然而,根据定义 1 和定理 1,对于任意足够小的 $\delta_2>0$,$Adv_{\mathcal{A}}^{\text{IND-CCA}}(\xi_1,q_1)\leqslant\delta_2$ 成立。因此,$Adv_{\mathcal{A}}^{\text{IND-CCA}}(\xi_1,q_1)\leqslant\delta_1$ 也成立,因为 $Adv_{\mathcal{A}}^{\text{IND-CCA}}(\xi_1,q_1)$ 依赖于 $Adv_{\mathcal{A}}^{\text{Hash, ECDLP}}(\xi_2,q_2,q_3)\leqslant\delta_2$。综上所述,本文方法对于自适应选择明文攻击具有不可区分性。

证毕。

为了验证本文方法在 game_2 中的安全性,对方法中的区块链网络进行了 51%攻击模拟,并通过概率的方式展示优势。具体地,对于区块链网络中的多个分叉,在模拟实验中假设敌手 $\mathcal{A}$ 所伪造的分叉链落后最长的分叉链 z 个区块。同时,考虑到极端情况,假设 $\mathcal{A}$ 在网络中聚集了超过 50%的计算能力。通过实验,分别分析了 $\mathcal{A}$ 在网络中占有不同比例计算能力与落后区块数量两种情况下对攻击成功概率的影响。假设 d_v 是 $\mathcal{A}$ 伪造的分叉链相较于最长区块链分叉落后 v 个区块时系统的难度值(是指通过枚举随机值 *nonce* 获得的哈希码的前置零个数),则在追赶第 v 个区块时需要的哈希计算次数为 2^{d_v},其中 $v\in\{1,2,\cdots,z\}$。因此,当 $\mathcal{A}$ 想要追赶上落后 z 个区块的最长分叉的概率可以表示为 $Adv_{\mathcal{A}}^{\text{bc}}=(2^{d_1}/2^{256})\times(2^{d_2}/2^{256})\times\cdots\times(2^{d_z}/2^{256})$。实际上,$2^{d_v}\ll 2^{256}$,因此当 z 足够大时,$Adv_{\mathcal{A}}^{\text{bc}}\to 0$。

对本文方法区块链进行 51%攻击的模拟实验结果如图 4 所示。由图 4 可知,在 $\mathcal{A}$ 占有网络中相同计算能力比例时,z 的值越大,$\mathcal{A}$ 攻击成功的概率越低。同时,在落后相同区块数量 z 的时候,$\mathcal{A}$ 占有计算能力比例越大攻击成功概率越高。例如,当 $\mathcal{A}$ 占有网络中 51%计算能力时,z 的取值大小对攻击成功的概率几乎没有影响,而在现实中,$\mathcal{A}$ 占有如此高的计算能力占比往往是不可能的,同时 z 的值也远远大于实验中的取值。综上所述,本文方法中的区块链是足够安全的。

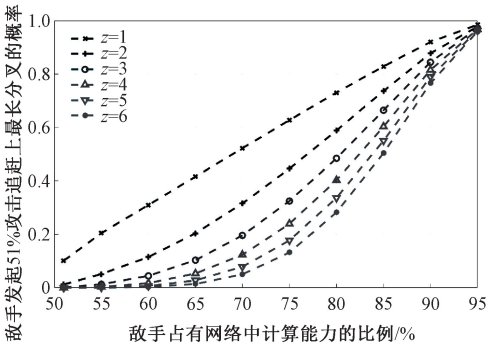

图 4　本文方法区块链中的 51%攻击概率分析

Figure 4　Probability analysis of 51% attack for blockchain in the proposed method

4　性能分析

4.1　实验环境配置

为了评估本文方法的性能,以两个广泛用于对比的相关工作为比较的基准,分别是 HRHT[17] 和

BVSD[21]方法。HRHT方法基于收敛加密算法,通过智能合约脚本创建一个能够抗篡改的账本,并通过智能合约脚本中的完整性审计协议来检测用户数据的完整性。而在BVSD方法中,用户可以通过调用智能合约脚本向云服务器提供自己的身份,并能够创建数据删除的请求作为区块链中的交易,然后云服务器对用户的数据进行删除并生成一条包含有删除证据的区块链。

本文所有的实验都在1台配备了12核24线程AMD Ryzen 9 5900X处理器和64 GB内存和Windows 10专业版操作系统的工作站上进行。本文中的异构云存储服务器的异构性主要体现在CPU体系结构、操作系统的不同。在实验中,本文对多个异构服务器的部署是通过虚拟机进行的。具体地,当异构服务器数量为1时,本文直接在工作站上进行实验;当数量超过2时,本文对Windows、Ubuntu、MacOS操作系统与X86、X64、ARM架构进行排列组合,并用虚拟机进行实验。本文方法和对比方法的原型都是通过Python编程语言实现或复现的。此外,实验中Python的版本为3.9.8,PyCharm版本为Community Edition 2021.3。

4.2 计算开销

在现有的云计算系统中,计算开销通常定义为CPU运行时间,单位为s[22]。首先对同一数据文件划分不同数量的数据块的数据去重计算开销进行实验,然后对不同异构服务器数量下的数据去重计算开销进行实验,实验结果如下。

在关于文件块数量对计算开销影响的实验中,假设每个服务器上存储了1 GB(1 024 MB)的文件数据。然后,将文件分割成不同数量的文件块,上传到云存储服务器中并进行去重操作,实验结果如图5所示。在图5中,随着文件块数量增加,所有方法的计算开销也随之增大。由于本文方法采用了轻量级的密钥管理方法,因此本文方法在不同的文件块数量上的计算开销均低于HRHT和BVSD方法。具体地,本文方法在文件块数量为500至6 000的情况下,平均计算开销比HRHT方法低65.42%,比BVSD方法低115.77%。此外,由于BVSD方法采用了基于属性的签名,相比HRHT方法需要额外的计算资源进行密钥管理。因此,HRHT方法在不同的文件块数量上的计算开销要低于BVSD方法。

在关于异构云存储服务器数量对计算开销影响的实验中,假设每个异构服务器上存储1 GB(1 024 MB)的数据。实验结果如图6所示。在图6中,随着服务器数量的增加,所有方法的计算开销也随之增大,因为多个服务器之间的通信需要耗费额外的计算资源。由于本文方法结合了智能合约和确定性秘密共享方法,计算开销均要低于HRHT和BVSD方法。同时,由于本文方法采用了轻量级的密钥管理方法,因此在面向多个异构云存储服务器的计算开销上要低于HRHT和BVSD方法。具体地,本文方法在异构服务器数量为1~10的情况下,平均计算开销比HRHT方法低67.27%,比BVSD方法低177.89%。此外,由于BVSD方法不能屏蔽多个服务器之间的异构性,因此BVSD方法的计算开销要高于HRHT方法。

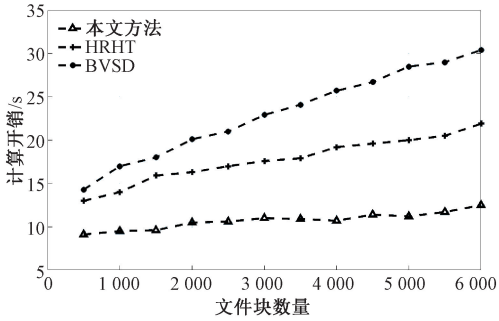

图5 相同文件大小在不同文件块数量下的计算开销

Figure 5 Computation overhead with respect to different number of data blocks in a file with same size

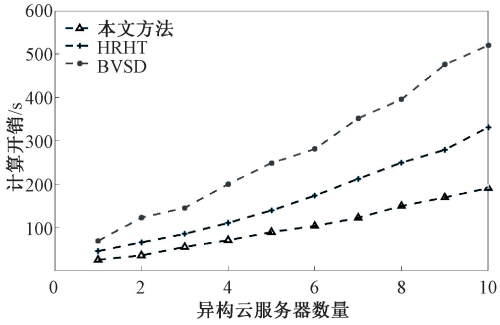

图6 不同数量异构云服务器下的计算开销

Figure 6 Computation overhead with respect to different number of the heterogeneous cloud servers

4.3 存储开销

在本节中,存储开销指的是在进行云存储服务器数据去重过程中,所有用于数据去重、恢复需要用到的存储空间大小,单位为MB。首先对同一个数据文件划分成不同数量数据块下的数据去重计算开销进行实验,然后再对不同异构服务器数量下的存储开销进行实验,具体的实验结果与分析如下。

在关于文件块数量对存储开销影响的实验中,假设每个服务器上均存储了1 GB数据。然后,将文件分割成不同数量的文件块,并上传到云存储服务器上进行去重操作,实验结果如图7所示。在图7

中,随着文件块数量的增加,所有方法的存储开销也随之增大。由于本文方法结合了基于区块链的智能合约和确定性秘密共享方法,在可恢复性去重方面的性能较为突出。因此,随着文件块数量的增加,本文方法的存储开销也不会出现大幅的增加。本文提出的去冗余方法不仅删除冗余文件,还能有效恢复原始数据,这在节省云存储系统存储空间的同时,不可避免地增加了去冗余和数据恢复的计算开销,是通过增加计算开销来提升存储能力的方法。相比之下,HRHT 方法由于采用了一种垂直的角色哈希树构建角色密钥,其存储开销要低于直接采用基于属性签名的 BVSD 方法。具体来说,本文方法在相同文件大小不同文件数量(500~6 000)情况下,平均存储开销比 HRHT 方法低 7.94%,比 BVSD 方法低 19.50%。

在异构云存储服务器数量对计算开销影响的实验中,假设每个服务器上均存储了 1 GB 数据文件。实验结果如图 8 所示,随着异构服务器数量的增加,所有方法的存储开销增大。本文方法由于采用了智能合约屏蔽了服务器之间的异构性,通过自动执行去重操作降低了云服务器的存储开销。同时,本文方法结合了秘密共享方法和智能合约策略,在可恢复去重方面的性能上要明显优于两个对比方法。因此,随着异构云存储服务器和数据量的增大,本文方法的存储开销不会出现较大的增幅。相较之下,由于 HRHT 方法由于采用了智能合约技术在运行环境层面上屏蔽了服务器的异构性,因此在存储开销的增幅上要低于 BVSD 方法。当异构服务器数量在 1~10 时,本文方法平均存储开销比 HRHT 方法低 34.01%,比 BVSD 方法低 72.89%。此结果证明了本文方法在异构云存储环境中安全去重性能的优越性。

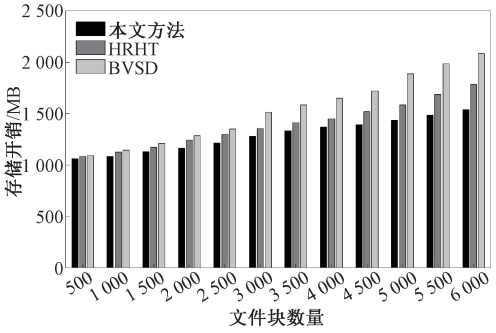

图 7　相同大小的文件在不同文件块数量情况下的存储开销

Figure 7　Storage overhead with respect to different number of data blocks in a file with same size

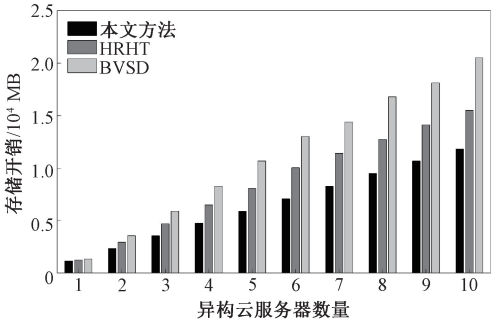

图 8　不同数量异构云服务器下的存储开销

Figure 8　Storage overhead with respect to the different number of the heterogeneous cloud servers

5　结论

针对现有云存储系统在大数据环境下存在的数据存储效率低、安全性弱以及存储服务器存在异构性等问题,本文提出了一种基于区块链智能合约的异构云服务器数据安全去重方法。具体结论如下。

(1)通过设计区块链取代集中式第三方实体的功能,消除了潜在的安全隐患,提高了数据的安全性。

(2)结合了秘密共享方法与基于区块链的智能合约技术,提高了去重的安全性和效率,并缓解了服务器之间的异构性。

(3)安全分析和实验结果表明,本文方法在安全性以及计算、存储开销上均要优于现有的两个基于区块链的去重方法,为云存储系统的数据安全去重提供了一种新的解决方法。

参考文献:

[1] LIU M Y, PAN L, LIU S J. Cost optimization for cloud storage from user perspectives: recent advances, taxonomy, and survey[J]. ACM Computing Surveys, 2023, 55(13):1-37.

[2] XIAO L, ZOU B J, ZHU C Z, et al. ESDedup: an efficient and secure deduplication scheme based on data similarity and blockchain for cloud-assisted medical storage systems[J]. The Journal of Supercomputing, 2023, 79(3): 2932-2960.

[3] YUAN H R, CHEN X F, LI J, et al. Secure cloud data deduplication with efficient re-encryption [J]. IEEE Transactions on Services Computing, 2022, 15(1): 442-456.

[4] LI J X, WU J G, CHEN L. Block-secure: blockchain based scheme for secure P2P cloud storage[J]. Information Sciences, 2018, 465: 219-231.

[5] LI J X, WU J G, CHEN L, et al. Blockchain-based secure key management for mobile edge computing[J]. IEEE Transactions on Mobile Computing, 2023, 22(1): 100-114.

[6] BELLARE M, KEELVEEDHI S, RISTENPART T. Message-locked encryption and secure deduplication[C]// Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2013: 296-312.

[7] QI S Y, WEI W, WANG J F, et al. Secure data deduplication with dynamic access control for mobile cloud storage[EB/OL]. (2023-04-03)[2023-06-18]. https://ieeexplore.ieee.org/abstract/document/10091139.

[8] PENG L, YAN Z, LIANG X Q, et al. SecDedup: secure data deduplication with dynamic auditing in the cloud[J]. Information Sciences, 2023, 644: 119279.

[9] YU X X, BAI H, YAN Z, et al. VeriDedup: a verifiable cloud data deduplication scheme with integrity and duplication proof[J]. IEEE Transactions on Dependable and Secure Computing, 2023, 20(1): 680-694.

[10] DOUCEUR J R, ADYA A, BOLOSKY W J, et al. Reclaiming space from duplicate files in a serverless distributed file system[C]// Proceedings 22nd International Conference on Distributed Computing Systems. Piscataway: IEEE, 2002: 617-624.

[11] CHEN R M, MU Y, YANG G M, et al. BL-MLE: block-level message-locked encryption for secure large file deduplication[J]. IEEE Transactions on Information Forensics and Security, 2015, 10(12): 2643-2652.

[12] DING W X, YAN Z, DENG R H. Secure encrypted data deduplication with ownership proof and user revocation[C]//International Conference on Algorithms and Architectures for Parallel Processing. Cham: Springer, 2017: 297-312.

[13] CUI H, DENG R H, LI Y J, et al. Attribute-based storage supporting secure deduplication of encrypted data in cloud[J]. IEEE Transactions on Big Data, 2019, 5(3): 330-342.

[14] HE K, CHEN J, DU R Y, et al. DeyPoS: deduplicatable dynamic proof of storage for multi-user environments[J]. IEEE Transactions on Computers, 2016, 65(12): 3631-3645.

[15] 王捷, 葛丽娜, 张桂芬. 区块链的激励机制权益证明共识算法改进方案[J]. 郑州大学学报(工学版), 2023, 44(5): 62-68.
WANG J, GE L N, ZHANG G F. Improvement scheme for the proof of stake consensus of blockchain incentive mechanism[J]. Journal of Zhengzhou University (Engineering Science), 2023, 44(5): 62-68.

[16] HUANG L X, ZHANG G X, YU S, et al. SeShare: secure cloud data sharing based on blockchain and public auditing[J]. Concurrency and Computation: Practice and Experience, 2019, 31(22): e4359.

[17] ZHANG G P, YANG Z G, XIE H R, et al. A secure authorized deduplication scheme for cloud data based on blockchain[J]. Information Processing & Management, 2021, 58(3): 102510.

[18] HUANG H, CHEN Q S, ZHOU Y P, et al. Blockchain-based secure cloud data deduplication with traceability[C]//International Conference on Blockchain and Trustworthy Systems. Berlin: Springer, 2020: 295-302.

[19] LIN C, HE D B, HUANG X Y, et al. BCPPA: a blockchain-based conditional privacy-preserving authentication protocol for vehicular ad hoc networks[J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 22(12): 7408-7420.

[20] ZHOU X T, LUO M, VIJAYAKUMAR P, et al. Efficient certificateless conditional privacy-preserving authentication for VANETs[J]. IEEE Transactions on Vehicular Technology, 2022, 71(7): 7863-7875.

[21] 刘忆宁, 周元健, 蓝如师, 等. 基于区块链的云数据删除验证协议[J]. 计算机研究与发展, 2018, 55(10): 2199-2207.
LIU Y N, ZHOU Y J, LAN R S, et al. Blockchain-based verification scheme for deletion operation in cloud[J]. Journal of Computer Research and Development, 2018, 55(10): 2199-2207.

[22] XIE Q Y, ZHANG C, JIA X H. Security-aware and efficient data deduplication for edge-assisted cloud storage systems[J]. IEEE Transactions on Services Computing, 2023, 16(3): 2191-2202.

(下转第142页)