软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2020,31(2):455−470 [doi: 10.13328/j.cnki.jos.005628]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

可验证的云存储安全数据删重方法*

咸鹤群[1,2], 刘红燕[1,2], 张曙光[1], 侯瑞涛[1]

[1](青岛大学 计算机科学技术学院,山东 青岛 266071)

[2](综合业务网理论及关键技术国家重点实验室(西安电子科技大学),陕西 西安 710071)

通讯作者: 咸鹤群, E-mail: xianhq@126.com

摘 要: 数据删重技术在云存储系统中得到了广泛的应用.如何在保证数据隐私的前提下,在半可信的云存储系统中实现高效的数据删重,是云计算安全领域的研究热点问题.现有方案在数据标识管理和用户数量统计方面普遍依赖于在线的可信第三方,执行效率有待提高,且容易造成系统瓶颈.提出了一种可验证的数据删重方法,无需可信第三方在线参与.基于双线性映射构造双文件标识方案进行流行度查询,确保标识不泄露数据的任何明文信息.采用改进的群签名方案,使用户可验证服务器返回的流行度标识,有效地防止云服务器伪造数据流行度的查询结果.设计了多层加密方案,可以根据数据的流行度,采用不同的加密方式.分析并证明了方案的安全性和正确性.通过仿真实验,验证了方案的可行性和高效性.

关键词: 数据删重;双线性映射;群签名;广播加密

中图法分类号: TP309

中文引用格式: 咸鹤群,刘红燕,张曙光,侯瑞涛.可验证的云存储安全数据删重方法.软件学报,2020,31(2):455−470. http://www.jos.org.cn/1000-9825/5628.htm

英文引用格式: Xian HQ, Liu HY, Zhang SG, Hou RT. Verifiable secure data deduplication method in cloud storage. Ruan Jian Xue Bao/Journal of Software, 2020,31(2):455−470 (in Chinese). http://www.jos.org.cn/1000-9825/5628.htm

Verifiable Secure Data Deduplication Method in Cloud Storage

XIAN He-Qun[1,2], LIU Hong-Yan[1,2], ZHANG Shu-Guang[1], HOU Rui-Tao[1]

[1](College of Computer Science and Technology, Qingdao University, Qingdao 266071, China)

[2](State Key Laboratory of Integrated Services Networks (Xidian University), Xi'an 710071, China)

Abstract: Data deduplication technology has been widely applied in cloud storage systems. Under the premise of ensuring data privacy, how to effectively perform deduplication in semi-trusted cloud storage environments becomes one of the primary issues in cloud computing security. Current schemes rely heavily on online trusted third parties to manage data labels and to keep track of the number of users. The trusted third party plays such a vital role in those schemes that it is indispensable even at the cost of unsatisfying efficiency and potential bottleneck. A verifiable secure data deduplication scheme in cloud storage is proposed, which doesnot require any online trusted third party. The dual-tag scheme based on bilinear mapping is adopted to conduct popularity check. The tag is used to retrieve files without leaking any exploitable information. A modified group signature scheme is designed to prevent the cloud server from forging popularity query results. Users can verify the authenticity of query results from the cloud server. The multi-layered cryptosystem is adopted in the proposed scheme, in which different encryption strategies are applied according to the popularity of specific data. The correctness and security of the proposed scheme are analyzed and proved. Simulation results show that the proposed scheme is secure and efficient.

* 基金项目: 国家自然科学基金(61303197); 综合业务网理论及关键技术国家重点实验室开放课题(ISN19-14); 赛尔网络下一代互联网创新项目(NGII20170414)

Foundation item: National Natural Science Foundation of China (61303197); Open Project of the State Key Laboratory of Integrated Services Networks (ISN19-14); CERNET Innovation Project (NGII20170414)

收稿时间: 2018-03-03; 修改时间: 2018-07-11; 采用时间: 2018-08-06

Key words: data deduplication; bilinear mapping; group signature; broadcast encryption

随着用户数据规模的快速增长,跨用户云存储成为数据存储的主流应用形态.从简单的备份系统到云存储系统,用户可以使用低成本、可扩展的在线服务.用户将数据外包至云服务器,由云服务器执行数据存储和管理.这种应用形态从根本上改变了资源部署和服务提供的方式,避免了用户对本地硬件维护成本的大量投入.用户数据量的增长,使云服务器产生了大量的冗余数据.云服务提供商(cloud service provider,简称 CSP)普遍采用数据删重技术提高存储空间利用率,从而降低成本[1].数据删重技术也称作重复数据删除技术(data deduplication),是指多个用户上传同一数据时,云服务器仅保留一份数据拷贝,为其他合法用户创建该数据拷贝的链接,由此节省存储空间和网络带宽[2].现有的商用云存储系统普遍应用了数据删重技术,如 Dropbox、Wuala、Mozy、Geogle Drive 等.研究表明,在备份系统中,数据删重技术可以有效地降低高达 90%~95%的存储需求.在普通应用系统中,该技术可降低 60%以上的存储需求[3,4].

数据删重技术具有广阔的商业发展前景,但同时也带来了新的安全问题.有效的数据删重方案要在保证数据安全的前提下实现重复数据的删除.为了保护数据隐私,用户通常先将数据加密,再将其外包至云服务器[5,6].然而,加密会对数据删重操作的执行效率产生巨大的影响.即使相同的数据,在不同的加密密钥和加密方式的作用下,所得的密文也不相同,因而云服务器难以根据密文直接判断数据是否来自同一明文.数据删重技术和数据加密技术的兼容性问题是目前云存储安全领域的研究热点之一.

根据删重操作发生的位置,可将数据删重技术分为服务器端数据删重和客户端数据删重.客户端数据删重应用较为普遍,可节省网络上传带宽,但易遭侧信道攻击和在线穷举攻击[7].服务器端数据删重要求所有数据都上传至云端,云服务器进行重复判断并执行删除操作.该方式能够有效抵御侧信道攻击和在线穷举攻击,但会浪费大量的网络带宽.根据处理数据的粒度,可将数据删重技术分为文件级数据删重和块级数据删重[8].与文件级数据删重相比,块级数据删重具有更小的数据粒度,执行效率更高,但为了实现数据的解密操作,需要在每个数据存储块中记录其所属的文件标识.以上方案各有优点和局限性,通常根据应用场景和安全需求进行选择.

目前已有的云存储数据删重方案主要关注的安全问题是如何在删重过程中防止云服务器直接或间接地解密用户数据,均未考虑在没有在线可信第三方的情况下,云服务器可能进行的欺骗行为.比如,很多方案根据拥有数据的用户数量对数据进行分类,并针对不同数据采取不同的保护措施.当云服务器回答用户对某数据的类型查询时,通过伪造查询结果,可以诱使用户使用安全性更低的加密方式.这种欺骗行为可以为 CSP 节约成本,也降低了窃取用户数据的难度.

本文提出了一种可验证的客户端数据删重方案(verifiable secure data deduplication,简称 VSDD),解决了在不借助在线第三方服务器的前提下,对云端操作的正确性进行判断和验证的问题.本文的贡献可归纳如下.

1) 利用双线性映射的性质,构造了双文件标识方案,实现数据重复判断以及进行流行度的查询.该过程不借助任何第三方服务器,并且不泄露任何数据明文信息.
2) 提出了基于群签名的数据签名方案,在无可信第三方的情况下,实现了用户对云服务器流行度标识返回结果的验证,可有效防止云服务器伪造数据的流行状态.
3) 设计了多层加密模式,利用广播加密技术进行密钥的发放,实现了客户端的数据删重,进一步降低了计算开销和通信开销.

1 相关工作

普通加密算法与数据删重系统之间存在兼容性问题,Douceur 等人首次提出了实现数据保密性和删重效率性的平衡方案——收敛加密(convergent encryption,简称 CE)[9−11].该方案使用数据散列值作为数据加密密钥,在加密算法确定的情况下,相同的数据可加密得到相同的密文.尽管 CE 简单且高效,但当数据的信息熵较低时,易遭受离线穷举攻击[9−11].此外,CE 不满足标识一致性原则,无法实现语义安全[12].

为了规范化收敛加密的安全定义,Bellare 等人提出了数据加密原语(message-locked encryption,简称 MLE),

但仍未从根本上实现数据的语义安全[12,13].

Stanek 等人提出根据数据的流行度,采取不同的保密措施[14].若数据内容为少数人所知,则视为隐私数据,即非流行数据;反之,为流行数据.采用新的阈值加密系统对数据进行多层加密,实现客户端数据删重.为了实现用户的身份管理和数据的安全删重,该方案引入了第三方服务器和索引服务器IS,以防止女巫攻击.但IS易遭受穷举攻击,额外的服务器增加了系统的复杂性,实用性不强.Puzio 等人提出借助第三方服务器协助云服务器进行数据删重的 perfectDedup 方案[15].该方案通过完美散列函数计算数据标识,且允许对流行数据块进行重复数据删除.对于非流行数据块,云服务器将保存多个数据拷贝.该方案仍无法防止来自云服务器的离线穷举攻击.

Liu 等人首次在不借助可信第三方服务器的前提下,提出了服务器端数据删重方案[1].拥有相同数据拷贝的用户,调用 Password Authenticated Key Exchange(PAKE)协议进行密钥交换[16,17].因此,拥有相同数据的用户使用相同的密钥对数据加密,得到相同的密文.与之前的方案相比,此方案摆脱了第三方,采用同态加密,安全性得到了一定的提高[18].但每个用户在上传数据之前都需要与其他用户执行 PAKE 协议,以交换加密密钥.这带来了额外的计算开销和通信代价,显著降低了数据删重的执行效率,对用户在线的要求也降低了该方案的实用性.

当数据被外包至云服务器,为了防止非授权用户对数据的访问,云服务器需要考虑数据的所有权问题,并且实现数据的动态所有权管理[19].Hur 等人提出了基于随机收敛加密(randomized convergent encryption,简称RCE)和组密钥管理机制的解决方案[20,21],将默克尔哈希树用于组密钥的加密和解密,实现所有权的动态管理.在该方案中,使用数据的双层哈希值作为标识,仍易遭受来自CSP的穷举攻击.

为了提高数据删重的效率,Cui 等人提出了一种基于密文策略属性加密的数据删重方案[22,23].该方案采用混合云结构,公有云负责数据的存储,私有云负责数据重复检测.这种混合云结构,增大了云服务器的管理开销.所有数据都采取属性加密策略,执行效率较低.文献[24]通过从实际智慧医疗系统中分析电子病历的内部特征,提出了一种加密电子病历删重系统,用户依赖于智慧手机的TEE管理密钥.该方案主要针对电子病历的数据,无法与通用方案进行比较.Singh 等人提出了基于密钥共享机制、将密钥分布到多个密钥管理服务器的数据删重方案[25].该方案基于 Permutation ordered binary(POB)数字系统实现数据的隐私保护.文献[26]提出了多媒体数据的模糊去重方案,方案采用模糊重复性检测,适用于非数值型数据.

为了保证数据标识不泄露数据明文的信息,上述方案都借助第三方服务器或混合云结构进行数据标识的管理,实用性较差.而如果没有可信第三方,用户无法对云服务器的流行度标识返回结果进行验证.云服务器可能伪造数据的流行状态,欺骗用户使用安全性低的加密方式,达到降低成本或者获取用户数据的目的.针对此问题,本文提出了一种可验证的云存储安全数据删重方法,无需可信第三方进行数据标识的管理和流行度的查询,实现了安全、高效的数据删重.

2 预备知识

2.1 双线性映射

双线性映射是构造密码体制的重要工具[27].设 n 是一个正整数,$(G_1,+)$和$(G_2,+)$是两个 n 阶的加法循环群,其零元分别为 0_1 和 0_2.设$(G_0,\cdot)$是一个 n 阶的乘法循环群,其单位元设为 1.并假设在群$(G_1,+),(G_2,+),(G_0,\cdot)$上计算离散对数是困难的.如果存在一个二元函数 $e:G_1\times G_2\to G_0$,满足以下性质,则称该二元函数 e 为双线性映射[28,29].

- 双线性
  1) $\forall P_1,P_2\in G_1,Q\in G_2$,满足 $e(P_1+P_2,Q)=e(P_1,Q)\cdot e(P_2,Q)$;
  2) $\forall P\in G_1,Q_1,Q_2\in G_2$,满足 $e(P,Q_1+Q_2)=e(P,Q_1)\cdot e(P,Q_2)$;
  3) 对任意的 $P\in G_1,Q\in G_2$ 和任意的 $a,b\in Z$,满足 $e(aP,bQ)=e(P,Q)^{ab}$.
- 非退化性
  1) $\forall P\in G_1,P\neq 0_1$,存在 $Q\in G_2$,使得 $e(P,Q)\neq 1$;
  2) $\forall Q\in G_2,Q\neq 0_2$,存在 $P\in G_1$,使得 $e(P,Q)\neq 1$.

2.2 随机收敛加密

对数据 M 进行收敛加密,具体操作为:加密密钥 $K=H(M)$,密文 $C=E(K,M)$,其中,哈希函数 $H:\{0,1\}^*\rightarrow\{0,1\}^*$,$E$ 为对称加密操作.云服务器可根据 CE 的密文判断是否来自同一明文,并由此进行数据删重.但收敛加密不满足标识一致性原则.假设 Alice 拥有数据 M',Bob 拥有数据 M,且 $M'\neq M$.Alice 恶意上传数据$\langle C_A,T(C_A)\rangle$,其中,$C_A$ 是 M'的密文,$C_A=E(H(M),M')$,$T(C_A)$是 M'的数据标识,$T(C_A)=H(E(H(M),M))$.当 Bob 此后计算出密文 $C_B=E(H(M),M)$、标识 $T(C_B)=H(E(H(M),M))$,并将$\langle C_B,T(C_B)\rangle$上传至云服务器.云服务器发现存在匹配的数据标识,$T(C_A)=T(C_B)$,将会执行数据删重,删除 C_B,仅保留 C_A.经过上述操作后,Bob 下载并解密数据,得到的将会是 M',这意味着标识的一致性遭到破坏.

为了解决 CE 可能出现的标识不一致问题,研究者提出了 RCE 方案[13].初始上传者随机选择加密密钥 L,加密数据 M 得 $C_1=E(L,M)$.计算 $C_2=L\oplus K$,其中,$K=H(M)$.用户计算 $C=C_1\|C_2$,并将 C 和标识 $T(M)=H(H(M))$发送至云服务器,仅保存 K.在数据下载阶段,用户可将解密后的数据按标识生成算法计算出 T',比较 T 和 T'.若二者相等,则认为数据无误.RCE 易遭受离线穷举攻击,因此仍无法达到语义安全,当数据隐私程度较高时,不宜采用 RCE.虽然在安全性上存在一定的局限,但由于其高效性,RCE 依然被广泛地应用在数据删重方案中.

2.3 广播加密

广播加密指广播中心(BC)可使发送者选取任意用户集合广播消息,使授权用户能获得解密密钥,从而对广播的消息进行正确解密[30].非授权用户无法获取解密密钥.广播加密方案一般包含权威的 BC、私钥产生器(PKG)两个实体.在基于身份的分层广播加密(HIBE)方案中,用户 ID 可以是任意能够代表用户身份的信息[31].PKG 将这些信息转换成用户的公/私钥对.BC 用公钥对信息加密,用户利用自己的私钥进行解密.HIBE 将对用户进行分层,上层用户可为下层用户生成私钥,这样既可减轻 BC 的计算负担,也可降低对 BC 的安全要求.HIBE 的密钥分配过程如图 1 所示.

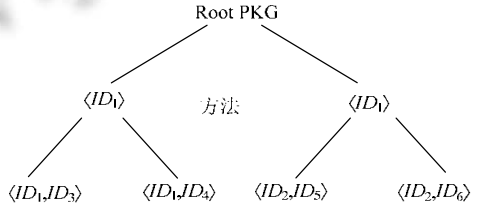


Fig.1 Key distribution procedure of HIBE

图 1 HIBE 的密钥分配过程

设λ为安全参数,$|S|$为接收用户集所能容纳的最大用户数量,HIBE 方案主要分为 5 个步骤.

1) *Setup*:初始化函数,$(mpk,msk)\leftarrow Setup(\lambda,|S|)$,输入$\lambda$和$|S|$,PKG 运行参数发生器产生系统主公/私钥对$(mpk,msk)$.其中,$mpk$ 公开,msk 仅 PKG 掌握.
2) *Extract*:私钥提取函数,$usk_{id}\leftarrow Extract(msk,id)$.输入 msk 和用户 id,输出用户私钥.
3) *UserKey*:用户私钥计算函数,$usk_{id}\leftarrow UserKey(id,usk_{id|l-1})$.输入用户 id 和该用户的父结点 $id|l-1$ 私钥,l 为该用户所在的层数,输出该用户私钥.
4) *KeyEncrypt*:广播加密可分为以下两个阶段:$(Hdr,K)\leftarrow Encrypt(S,mpk)$,$C\leftarrow Encrypt(K,M)$.输入接收用户集 $S=(ID_1,ID_2,\ldots,ID_i)$,其中,$1\leqslant i\leqslant|S|$.BC 利用 mpk 生成报头信息 Hdr 和会话密钥 K.设 M 为 BC 将要广播的信息,用 K 对 M 加密可得密文 C,广播(Hdr,C).
5) *KeyDecrypt*:用户解密可分为两个阶段: $K\leftarrow Decrypt(S,ID_i,usk_{ID_i},Hdr,mpk)$ 以及 $M\leftarrow Decrypt(K,C)$.用户输入自己的 ID_i、私钥和广播的数据.如果 $ID_i\in S$,用户可用自己的私钥获得解密密钥 K,进而通过 K 解密 C[32].

2.4 计算Diffie-Hellman问题(CDH问题)

设 G 表示阶为大素数 P 的乘法循环群,群 G 的 CDH 问题可描述为:对于给定的 $P^a,P^b\in G$,其中,$a,b\in Z_n^*$ 是未知的整数,计算 $Q=P^{ab}\in G$.

3 方 案

为了提高数据删重效率,我们将数据分为流行数据和非流行数据.设 t 表示数据的流行度阈值.对于数据 M,$count_M$ 表示云服务器中对该数据拥有访问权限的用户数量,则非流行数据是指当前的 $count_M<t$,具有较高隐私性的数据,需要采取语义安全的加密算法进行保护.随着拥有访问权限的用户数量不断增长,数据的隐私程度逐渐降低.当 $count_M\geqslant t$ 时,数据状态由非流行状态转换为流行状态.

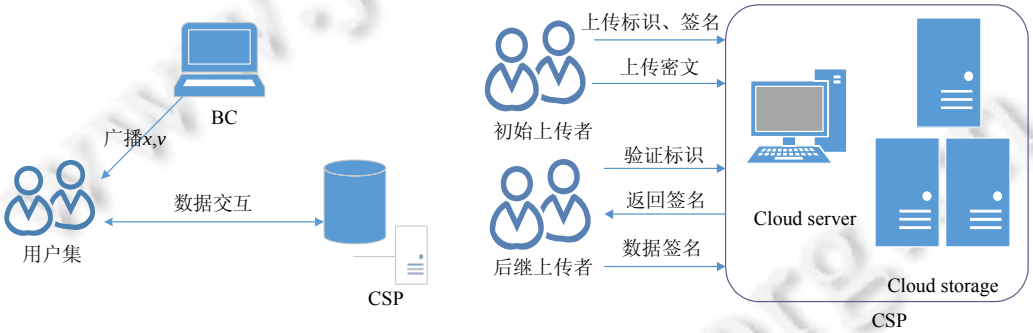
为了安全、高效地进行客户端的数据删重,设计的方案应该具有以下几个性质.

1) 数据安全性:保证用户外包至云服务器的数据安全.
2) 数据完整性:保证数据的完整性且其完整性用户可验证.
3) 数据访问控制:确保仅拥有访问权限的用户可访问数据.
4) 流行度的可验证:确保流行度判断的正确性和流行度查询结果的可验证性.

3.1 系统模型与敌手模型

3.1.1 系统模型

如图 2 所示,VSDD 方案的系统模型包含 3 个实体:数据所有者、BC 和 CSP.在系统建立时,BC 为用户群体中每个用户广播密钥 x 和 v.CSP 为每个数据所有者提供数据的存储和共享服务,可实现数据的访问控制.用户集通过与云服务器进行交互,获取数据的所有权.



(a) 系统架构 (b) 数据交互

Fig.2 System model

图 2 系统模型

数据所有者是指拥有数据,且将数据外包至云服务器存储的用户.若用户上传的数据在云端不存在,则认为该用户是初始上传者;否则,称该用户为后继上传者.每个用户都有其唯一的 ID 号,与用户一一对应.当用户提出下载请求时,云服务器先判断该用户是否具有该数据的访问权限.为了防止云服务器伪造流行度查询结果,每个用户都将数据签名上传至云服务器,以便对数据流行度状态进行验证.

本方案采用多层加密系统,所有数据可分为流行数据和非流行数据.根据数据不同的流行度,采用不同的加密方式.对非流行数据,可采用双层加密模式,内层采用 RCE,外层基于广播加密的密钥 v 进行对称加密.对流行数据仅采用 RCE 加密.

3.1.2 敌手模型

在本文实现的数据删重方案中,我们将主要考虑以下两类攻击者.

1) 外部攻击者:该类攻击者可获得部分数据,也可对网络中传输的数据进行监听,从而对用户数据进行穷举攻击.
2) 内部攻击者:主要指云服务器端.我们认为云服务器是诚实但好奇的实体,可在用户不知晓的情况下,对云中存储的数据进行任意访问.

3.2 符号说明

本文定义 $x\xleftarrow{R}Z$ 为从有限集合 Z 中随机选取元素 x.设$(G_1,+),(G_0,\cdot)$分别是阶为大素数 P 的加法循环群和乘法循环群,且 g 为群 G_1 的一个生成元.定义双线性映射 e:$G_1\times G_1\to G_2$.H 为一个映射函数,其特性为$\{0,1\}^*\to G_1$.$Hash$ 表示哈希函数,其特性为$\{0,1\}^*\to\{0,1\}^*$.对于数据 M,$K=Hash(M)$为其对应的哈希值.$E(K,M)\to C$ 表示使用密钥 k 对 M 进行对称加密,$D(k,C)\to M$ 表示使用密钥 k 对数据 C 进行解密.Sig 表示数据的签名.

每个用户都有对应的身份标识 ID.设用户集 $US=\{U_1,U_2,U_3,\ldots,U_m\}$,对应的身份集合为$\{ID_1,ID_2,ID_3,\ldots,ID_m\}$.所有权集 G_{ID} 是指对同一数据拥有访问权限的用户集合.设数据 M 对应的所有权集为 G_{ID},若 $ID_u\in G_{ID}$,则认为用户 U_u 对 M 拥有访问权限.

3.3 可验证的云存储安全数据删重方法

在 VSDD 方案中,云服务器负责维护云中数据的所有权列表 *File_list*,其结构见表 1.在 *File_list* 结构中,数据标识与 CSP 端存储的数据之间建立了一一对应关系,云服务器可通过检索数据标识来查找数据.为了防止云服务器伪造流行度查询结果,用户可对数据进行签名,并通过检验签名来验证流行度查询结果的正确性.只有当数据处于非流行状态时,数据签名才会被更新.流行度标识是数据是否达到流行度阈值的标志,如果是流行数据,则流行度标识为 1;反之为 0.

Table 1 List of *File_list*

表 1 *File_list* 列表

序号	标识	密文	签名 *Sig*	G_{ID}	流行度状态
R_1	U_1,B_1	C_1	$\langle\theta_0,\varepsilon_0,num_0\rangle,\langle\theta_1,\varepsilon_1,num_1\rangle,\ldots,\langle\theta_t,\varepsilon_t,num_t\rangle$	$ID_1,\ldots,ID_t$	1
R_2	U_2,B_2	C_2	$\langle\theta_0,\varepsilon_0,num_0\rangle,\ldots,\langle\theta_i,\varepsilon_i,num_i\rangle(i\prec t)$	$ID_1,\ldots,ID_i$	0
…	…	…	…	…	…

3.3.1 数据标识的检查

本文采用基于双线性映射的双文件标识作为文件查找的关键词.用户将数据标识上传至云服务器,云服务器通过检索云中的 *File_list* 列表,判断该数据是否已存储在云端.双文件标识构造和检查操作如下.

1) 对于数据 M,用户随机选择参数 $u(u\in Z_p)$,并计算出 $K=Hash(M)$,令 $U=g^{Hash(M)*u}$,$B=g^u$;
2) 用户将$\langle U,B\rangle$作为数据标识,构造上传请求 $upload\|\langle U,B\rangle\|ID$,发送至云服务器;
3) 云服务器收到上传请求后,遍历 *File_list* 列表,查询是否存在$\langle U',B'\rangle$,使得 $e(U,B')=e(U',B)$.

通过比较双线性映射结果,即可判断该数据是否已存储在云端.采用$\langle U,B\rangle$作为数据标识,含 u,$Hash(M)$两个未知参数,CSP 无法根据已知的参数 U,B 进行穷举攻击,从而推断出用户存储的明文数据.

3.3.2 数据签名

当用户发出上传请求,如果数据已存储在云端,CSP 会返回该数据的流行度标识.一旦 CSP 伪造流行度查询结果,这将会导致非流行数据泄露的风险增大.本文通过对群签名方案进行改进,设计出新的签名方案,摆脱了第三方服务器.用户验证签名即可实现对流行度查询结果的验证,消除了 CSP 伪造流行度查询结果的可能.

用户比较数据所对应的用户数量 $count_M$ 和流行度阈值 t.若 $count_M<t$,则该数据为非流行数据,用户需要上传对该数据的签名,并更新数据签名链表;若 $count_M\geqslant t$,表示用户数量已经达到了流行度阈值,则 CSP 无需更新该数据的签名.

数据签名以链表的形式进行存储.数据签名结构为$\langle\theta,\varepsilon,num\rangle$,其中,$\langle\theta,\varepsilon\rangle$是签名,$num$ 是对链表的顺序编码.数据签名链表的结点之间存在关联关系,上一个结点影响下一个结点的值.数据签名方案具体如下.

初始变量的设置:θ_0=0,λ_0=0 以及 num_0=0;

1) 随机选择参数 $x \in Z_p^*$,并进行广播加密,将 x 发送给所有用户;

2) 用户 U_i 接收到 CSP 返回的数据签名$\langle \theta_{i-1},\lambda_{i-1},num_{i-1}\rangle$,并对数据 M 进行签名.假设 U_i 的私钥为 $y_i \in Z_p^*$,其公钥为 $Y_i=y_ig$;

3) 用户 U_i 随机选择 $z_i \in Z_p^*, Z_i = z_ig$,计算 $\begin{cases}\sigma_i = (x+y_i+z_i)H(M)-\lambda_{i-1}H(M)\\ \lambda_i = y_i+z_i\\ num_i = num_{i-1}+1\end{cases}$;

4) 用户将计算出的元组$\langle \sigma_i,\varepsilon_i,num_i\rangle$发送给 CSP;

5) CSP 收到该元组后,在签名链表中新增一个结点,结点数值为$\langle \theta_i,\lambda_i,num_i\rangle$,其中,

$$\theta_i=\theta_{i-1}+\sigma_i=(num_i*x+y_i+z_i)H(M);$$

6) U_i 对签名进行验证,判断公式(1)是否成立:

$$e(g,\theta_i)=e(num_i*x*g+\lambda_i*g,H(M)) \quad (1)$$

若公式(1)成立,则认为云服务器对数据流行度的判断是正确的.

用户通过对云服务器返回的数据签名进行验证,可判断流行度查询结果的正确性.数据签名中的数据项 num 与阈值相比较,用户即可判断该数据的流行度.

3.3.3 数据上传

根据数据的流行度不同,可将数据上传分为流行数据的上传和非流行数据的上传.用户提出数据上传请求时,云服务器将会返回数据签名链表中最后一个结点数据,用户可通过第 3.3.2 节的验证方式进行验证,判断出数据类型,并执行相应操作.数据 M 的上传过程具体如图 3 所示.

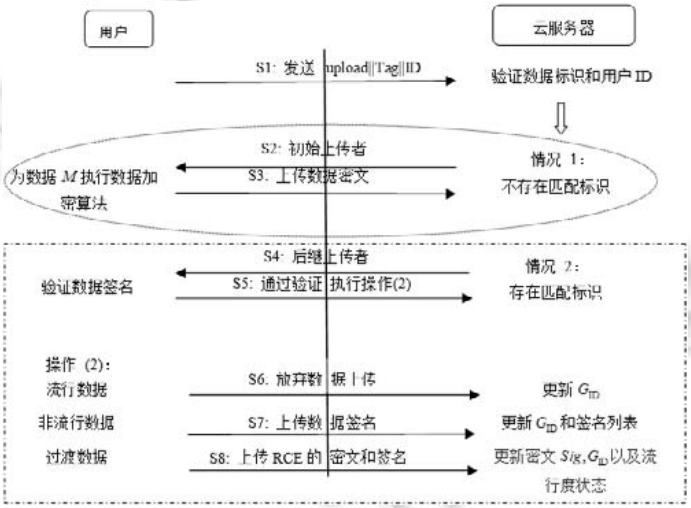


Fig.3 Data upload procedure

图 3 数据上传过程

用户运行标识生成算法,并将数据标识和用户 ID 上传至云服务器.云服务器根据数据标识进行检索,判断 M 是否已存储在云端.根据检测结果,可分为情况 1 和情况 2.情况 1 表示未检测到匹配标识,则将该用户视为 M 的初始上传者.情况 2 表示 M 已存储在云端,则将该用户视为后继上传者,且该用户无需继续上传数据密文,仅上传签名即可.用户可验证云服务器返回的数据签名,若数据签名验证成功,则执行操作(2).根据验证结果,用户可将数据 M 分为流行数据、过渡数据和非流行数据,并执行相应的操作,其中,过渡数据是指 $count_M=t$ 的数据.

由于流行数据的隐私程度较低,因此仅采用 RCE 进行加密,且可进行客户端删重.对非流行数据,VSDD 方

案采取多层加密.外层加密密钥可通过以下操作获得.BC 运行 *Setup* 算法,获得系统的主公/私钥对(*mpk*,*msk*),仅公开 *mpk*.通过分层调用 *Extract* 算法和 *UserKey* 算法,用户可获得私钥 usk_{id}.BC 通过 *KeyEncrypt* 算法对加密密钥 v 进行加密,输出(Hdr,S,C_v).授权用户可调用 *KeyDecrypt* 算法,解密 C_v,从而获得加密密钥 v.对于数据 M,其加密过程具体如下.

- 若 M 是非流行数据
  1) 使用 v 作为加密密钥,这一参数云服务器不可知.用户使用 v 进行外层加密.
  2) 采用 RCE 加密,用户随机选择参数 L.
  3) 内层加密:$C_1=E(L,M)$,$C_2=L\oplus K$,$C_3=C_1\|C_2$.
  4) 外层加密:$C=E(v,C_3)$.用户将 C 发送至云服务器,仅保存 K 值.
- 若 M 是流行数据
  1) 采用 RCE 加密,用户随机选择参数 L.
  2) 内层加密:用户计算 $C_1=E(L,M)$,$C_2=L\oplus K$,$C_3=C_1\|C_2$.

3.3.4 数据下载

用户需要先将数据标识、用户 ID 发送给云服务器.上传用户 ID 可有效防止用户之间合谋以及非法用户对数据的访问.数据下载的具体过程如图 4 所示.

Fig.4 Data download procedure

图 4 数据下载过程

云服务器接收到用户的数据下载请求时,先根据数据标识,判断该数据是否已存储在云端;其次,判断当前用户是否具有该数据的访问权限.若满足前两个条件,云服务器返回该数据签名链表中最后一个签名、流行度标识和该数据的密文.用户验证返回的流行度标识是否正确,并根据流行度标识采取适当的解密算法进行解密.若云服务器中不存在该数据或当前用户不具有访问权限,则下载请求无效.

流行数据和非流行数据采取不同的加密方式,因此对应于不同的解密措施.用户保存数据的哈希值 K 以及广播密钥 v.具体解密措施如下.

- 非流行数据的解密
  1) 对密文进行外层解密:$C_3=D(v,C)$.
  2) 对解密出的 C_3 进行内层解密:$L=C_2\oplus K$,$M=D(L,C_1)$.
- 流行数据的解密

  对下载的密文进行 RCE 算法的解密:$L=C_2\oplus K$,$M=D(L,C_1)$.

4 安全性证明与分析

本节从 3 个方面详细分析 VSDD 方案的安全性.

4.1 数据标识的安全性证明

用户需提交数据标识至云服务器,作为数据查找的关键词.VSDD 方案基于双文件标识进行数据的查找和流行度的检测,数据标识的安全性建立在特殊的散列函数 *Hash* 和双线性映射 *e* 之上.

双线性映射定义为 $e:G_1\times G_1\to G_2$,其中,$G_1=\langle g\rangle$,g 为群 G_1 的生成元.由散列函数的安全假设,可得出引理 1.

引理 1. 对于安全的散列函数 $Hash$,$\forall M_1,M_2\in\{0,1\}^*$且 $M\neq M'$,$Hash(M_1)=Hash(M_2)$的概率是可忽略的,可设为θ.即 $\Pr[Hash(M_1)=Hash(M_2)|M\neq M']\prec\theta$.

定理 1(数据标识的唯一性). 设数据 M 的初始上传者 U_i 已选择随机数 u,并将 M 的标识$\langle U,B\rangle$保存在云服务器,其中,$U=g^{Hash(M)*u}$,$B=g^u$.当用户 U_j 上传数据 M'时,随机选择 u',计算 $U'=g^{Hash(M')*u'}$,$B'=g^{u'}$.则当且仅当 $M=M'$时,$e(U,B')=e(U',B)$成立.即若 $M\neq M'$,则$\exists(e(U,B')=e(U',B))$的概率是可忽略的:

$$\Pr[e(U,B')=e(U',B)|M\neq M']\prec\theta.$$

证明:采用反证法进行证明.假设存在 $M\neq M'$,使得 $e(U,B')=e(U',B)$.即

$$\begin{aligned}e(U,B')=e(U',B)&\Leftrightarrow e(g^{Hash(M)*u},g^{u'})=e(g^{Hash(M')*u'},g^{u})\\&\Leftrightarrow e(g,g)^{Hash(M)*u*u'}=e(g,g)^{Hash(M')*u'*u}\\&\Leftrightarrow e(g,g)^{Hash(M)}=e(g,g)^{Hash(M')}.\end{aligned}$$

因为 u 和 u'为两个用户随机选择的参数,极大概率下,$u\neq u'$,即 $\Pr[u=u']\prec\theta$.

正如引理 1 所示,若 $M\neq M'$,则 $\Pr[Hash(M)=Hash(M')]\prec\theta$.

不失一般性,可得出 $Hash(M)\neq Hash(M')$,从而 $e(g,g)^{Hash(M)}\neq e(g,g)^{Hash(M')}$.

这与假设相矛盾,所以假设不成立.即当且仅当为同一数据时,才可能满足该双线性映射等式.由此可证数据标识具有唯一性. □

定理 2(数据标识的正确性). 数据 M 的初始上传者 U_i 选择随机数 u,并将 M 的标识$\langle U,B\rangle$保存在云服务器中,其中,$U=g^{Hash(M)*u}$,$B=g^u$.CSP 将$\langle U,B\rangle$保存为 M 的数据标识.当用户 U_j 上传数据 M'时,随机选择 u',计算出 $U'=g^{Hash(M')*u'}$,$B'=g^{u'}$.数据标识是正确的,即存在 $e(U,B')=e(U',B)$,但 $M\neq M'$的概率是可忽略的:

$$\Pr[M\neq M'|e(U,B')=e(U',B)]\prec\theta.$$

证明:根据双线性映射 $e(U,B')=e(U',B)$,又因为 $e(U,B')=e(g^{Hash(M)*u},g^{u'})=e(g,g)^{Hash(M')*u*u'}$,可得 $Hash(M)=Hash(M')$.正如引理 1 所示,可得出 $M=M'$.由此可证,数据标识满足正确性原则.只要满足该双线性映射等式,即可视为同一数据. □

4.2 数据签名的安全性证明

4.2.1 数据签名的正确性

定理 3(数据签名的正确性). 用户仅验证数据签名链的最后一个签名即可判断出所有数据签名的正确性.

证明:采用反证法证明.假设数据 M 的签名链中第 j+1 个结点$(\theta_j,\lambda_j,num_j)$是正确的,但第 i+1 个结点$(\theta_i,\lambda_i,num_i)$是错误的,其中 $0\leqslant i\leqslant j$.数据签名链可表示为$(\theta_0,\lambda_0,num_0)\to\ldots\to(\theta_i,\lambda_i,num_i)\to\ldots\to(\theta_j,\lambda_j,num_j)$.

因为第 j+1 个签名是正确的,故满足等式 $e(g,\theta_j)=e(num_j*x*g+\lambda_j*g,H(M))$.

根据签名的构造方法,可得$\langle\theta_j,\lambda_j,num_j\rangle$.

其中,$\begin{cases}\sigma_j=(x+y_j+z_j)H(M)-\lambda_{j-1}H(M)\\ \lambda_j=y_j+z_j\\ num_j=num_{j-1}+1\\ \theta_j=\theta_{j-1}+\sigma_j=(num_j*x+y_j+z_j)H(M)\end{cases}$.

可得$\begin{cases}\theta_{j-1}=\theta_j-\sigma_j=(num_{j-1}*x+y_{j-1}+z_{j-1})H(M)\\ \lambda_{j-1}=y_{j-1}+z_{j-1}\\ num_{j-1}=num_j-1\end{cases}$.

$\Rightarrow e(g,\theta_{j-1})=e(num_{j-1}*x*g+\lambda_{j-1}*g,H(M))$等式成立,即第 j 个签名$(\theta_{j-1},\lambda_{j-1},num_{j-1})$是正确的.

同理,依此逆向递推,我们可证明出等式 $e(g,\theta_i)=e(num_i*x*g+\lambda_i*g,H(M))$成立,即第 $i+1$ 个签名$(\theta_i,\lambda_i,num_i)$是正确的.这与假设相矛盾,因此假设不成立.

综上所述,用户可通过最后一个签名的验证,完成对整个数据签名链表的验证. □

4.2.2 数据签名的安全分析

上文中数据签名正确性的证明,说明了签名具有不可伪造性.每当用户收到云端返回的流行度查询结果时,用户都需对云服务器返回的最后一个签名进行验证.一旦云服务器伪造了数据签名,则该签名将无法通过验证.

在数据签名的构造过程中,云服务器已知的参数包括上一个结点数值$(\theta_{i-1},\lambda_{i-1},num_{i-1})$和当前用户上传的数据$(\sigma_i,\lambda_i,num_i)$.因为$\sigma_i=(x+y_i+z_i)H(M)-\lambda_{i-1}H(M)$中的 $H(M)$和 x 对于云服务器而言是未知参数,云服务器可对 M 进行穷举攻击,进而对 $H(M)$进行猜测.但因为云服务器不是授权用户,无法获得密钥 x,因此云服务器成功伪造正确的数据签名$(\theta_i,\lambda_i,num_i)$的可能性极低.

定理 4(数据签名的匿名性). 如果群 G_1 上的 CDH 问题是困难的,则本方案具有匿名性.即敌手不能根据签名,判断出具体的签名者.

证明:在随机预言机模型下,我们证明改进的群签名的匿名性,并为敌手提供最大可能的攻击性[33].采用类似文献[33]的游戏进行证明.

游戏 1. 假设敌手知道每个群成员的私钥 y 以及 BC 发送的密钥 x.设$\mathcal{A}$是方案匿名性的攻击算法.我们可进行游戏,$\mathcal{A}$能够对任意消息产生签名.经过上述过程的多次重复后,$\mathcal{A}$停止,输出消息 M 以及两个诚实的用户 U_{i_0},U_{i_1}.假设在猜测阶段用户 U_{i_b} 生成签名 $Sig_{i_b}=\langle\theta_{i_b},\lambda_{i_b},num_{i_b}\rangle$,其中,$b\in\{0,1\}$.$\mathcal{A}$需要猜测出用户 U_{i_0} 和 U_{i_1} 具体哪个是真正的签名者.一旦$\mathcal{A}$猜测正确,则证明该签名方案不具有匿名性.

假设$\mathcal{A}$能够正确地猜测出签名者是 U_{i_0}.因为$\mathcal{A}$已知参数 x 以及 U_{i_0} 的私钥 y_{i_0},所以,$\mathcal{A}$可以计算出 $v_{i_0}=(num_i*x)H(M)$, $\mu_{i_0}=y_{i_0}H(M)$, $w_{i_0}=\theta_{i_0}-v_{i_0}-\mu_{i_0}$ 以及 $Z_{i_0}=z_{i_0}g=\lambda_{i_0}H(M)-\mu_{i_0}$.

因为 $Z_{i_0},H(M)\in G_1$,所以存在 $a,b\in Z_n^*$,使得 $Z_{i_0}=ag,H(M)=bg$,则

$$\begin{aligned}e(g,w_{i_0})&=e(g,\theta_{i_0}-v_{i_0}-\mu_{i_0})\\&=e(g,\theta_{i_0})e(g,v_{i_0})^{-1}e(g,\mu_{i_0})^{-1}\\&=e(g,(num_{i_0}*x+y_{i_0}+z_{i_0})H(M))\cdot e(g,num_{i_0}*xH(M))^{-1}\cdot e(g,y_{i_0}H(M))^{-1}\\&=e(g,z_{i_0}H(M))\\&=e(Z_{i_0},H(M))=e(ag,bg)=e(g,abg).\end{aligned}$$

根据双线性映射的非退化性,可知 $w_{i_0}=abg$.这意味着$\mathcal{A}$解决了群 G_1 中的一例 CDH 问题.因为群 G_1 的 CDH 问题是困难的,所以我们的签名方案具有匿名性. □

4.3 抵御来自CSP的攻击

对数据 M,云服务器存储初始上传者上传的数据标识$\langle U,B\rangle$,其中,$U=g^{Hash(M)*u}$,$B=g^u$.在双文件标识$\langle U,B\rangle$中,云服务器已知的参数仅有 g.参数 $Hash(M)$是关于数据本身的信息,因为哈希算法具有公开性,CSP 可对 M 进行穷举攻击.但由于参数 u 是由用户随机选择的,因此 CSP 无法获得 u.另一方面,为了进行穷举攻击,CSP 需要从已知参数 U 和 B 中得到 $Hash(M)$.从 U 中计算得到 $Hash(M)*u$,是一个离散对数困难问题.为了恢复出 $Hash(M)$,还需由 B 计算参数 u,这又是一个离散对数问题,所以 CSP 难以对 M 的内容进行穷举攻击.因此,CSP 无法从数据标识$\langle U,B\rangle$中推测出数据 M 的任何明文信息,即 VSDD 方案可抵御来自 CSP 的攻击.

5 仿真与实验分析

本方案采用 C++语言,利用 CMP[34]、PBC[35]、OPENSSL、PBC_bce 函数库进行方案实现.运行平台是具有 4GB 运行内存、4 核 CPU、1Mbps 带宽的腾讯虚拟云服务器,运行的操作系统是 Linux.在实验中,我们设置 512 比特的基域,其中每个元素 $element$ 的大小为 160 比特,且 $element\in Z_P^*$.

为了模拟真实的实验环境,我们对云端的 *File_list* 列表进行随机化设置,在列表中包含超过 2 000 个不同的数据项,且非流行数据和流行数据的比例接近 1:1.数据的流行度阈值设置为 7.

本节从通信开销和计算代价两个方面,与 perfectDedup 方案进行比较,分析 VSDD 方案的性能优势.由于广播加密仅在系统初始化时执行 1 次,因此其时间开销不计入实验结果.

我们将不同体积的数据上传至云服务器,分以下 3 个场景进行模拟实验.

- 场景 1(非流行数据):该数据在本次上传完成后,仍为非流行数据.
- 场景 2(流行度转换):该数据在本次上传完成后,由非流行数据转换为流行数据.
- 场景 3(流行数据):该数据在本次上传之前,已为流行数据.

5.1 通信开销

我们从理论上对 CE、RCE 以及提出的方案进行性能比较,比较结果见表 2.表 2 主要对 VSDD 方案的通信开销和存储开销进行了分析.在通信开销方面,上传的数据规模包括数据外包请求、数据上传过程中所需传输的数据量;下载的数据规模指数据下载、数据标识检查等所需传输的数据量.存储开销是指用户需要存储的信息,如数据加密密钥等.对数据 M,符号 C_M,C_T,C_{Sig},C_C,C_E 分别表示数据规模、生成的数据标识规模、数据签名规模、基于收敛加密所得的密文规模、基于对称加密所得的密文规模.C_K 表示数据加密过程中密钥的规模.C_{ID} 表示用户身份标识的规模.C_x,C_v 表示广播加密所得参数 x 和 v 的规模.

在 RCE 方案中,密文 C_C 包括两个部分:数据加密所得的密文和密钥加密所得的密文.与 CE 和 RCE 方案相比,VSDD 方案在数据上传和下载的通信开销方面,仅额外传输了签名信息;在存储开销上,仅增加了密钥参数信息,数据体积较小,都没有增加过多的额外开销.数据签名的验证用于保证数据的安全性,防止 CSP 伪造流行度查询结果.此外,VSDD 方案根据数据流行状态采取不同的加密措施,可实现更好的数据保护.

Table 2 Comparison of performance

表 2 性能比较

方案	通信开销		存储开销	
	上传的数据规模	下载的数据规模	加密密钥的规模	数据标识规模
CE	$C_C+C_T+C_{ID}$	C_C	C_K	C_T
RCE	$C_C+C_T+C_{ID}$	C_C+C_T	C_K	C_T
VSDD 流行数据	$C_C+C_T+C_{ID}+C_{Sig}$	$C_T+C_{Sig}+C_C$	C_K	C_T
VSDD 非流行数据	$C_T+C_{ID}+C_{Sig}+C_E$	$C_T+C_{Sig}+C_E$	C_K+C_v	C_T

5.2 计算代价

为了直观地分析 VSDD 方案的优势,我们记录并分析了 3 种场景下数据加密和解密所用的时间.实验使用 MD5 作为加密哈希函数,用于密钥生成、数据标识以及签名的产生.实验采用 AES 作为对称加密的函数.具体的实验结果见表 3,其中,T_{Hash} 表示运行哈希函数所用时间,Enc 和 Dec 分别表示数据的加密和解密所用时间.

Table 3 Comparison of CE, RCE and the proposed scheme in terms of execution time

表 3 CE、RCE 以及提出方案的执行时间比较

具体操作	时间(s)	CE		RCE		VSDD 方案			
		上传	下载	上传	下载	上传		下载	
						流行数据	非流行数据	流行数据	非流行数据
T_{Hash}	0.025	2	–	2	2	3	3	2	2
数据加密	Enc	1	–	1	–	1	2	–	–
数据解密	Dec	–	1	–	1	–	–	1	2
总体计算开销(s)		Enc +0.05	Dec	Enc +0.05	Dec +0.05	Enc +0.075	2Enc +0.075	Dec +0.05	2Dec +0.05

为了讨论不同场景下,数据上传过程中服务器和客户端的计算时间开销各自所占的比例,我们进行了 3 组模拟实验,实验结果如表 4 和图 5 所示.将不同体积的数据设置为非流行数据、流行度转换数据、流行数据,并

分别执行数据的上传操作.如果数据在云服务器中已存在,我们将执行客户端删重,这样可以节省大量的网络带宽.通过比较实验结果可以看出,除流行数据的上传操作外,其他场景下客户端所消耗的时间都远小于服务器端.这是因为非流行数据需要上传签名,流行度转换数据需上传 RCE 加密的密文.与服务器端的计算开销相比,客户端的计算开销非常小,大部分计算任务由云服务器承担.

Table 4 Time span used by server and client side in different scenarios (ms)

表 4 不同场景下服务器端和客户端所消耗的时间 (ms)

	8MB		32MB		128MB		512MB	
	服务器端	客户端	服务器端	客户端	服务器端	客户端	服务器端	客户端
场景 1	3 264	1 565	11 254	2 643	31 551	6 573	160 992	18 532
场景 2	3 141	1 160	9 964	2 053	25 810	4 934	135 650	14 257
场景 3	1 298	925	2 605	1 024	4 675	2 805	9 684	5 663

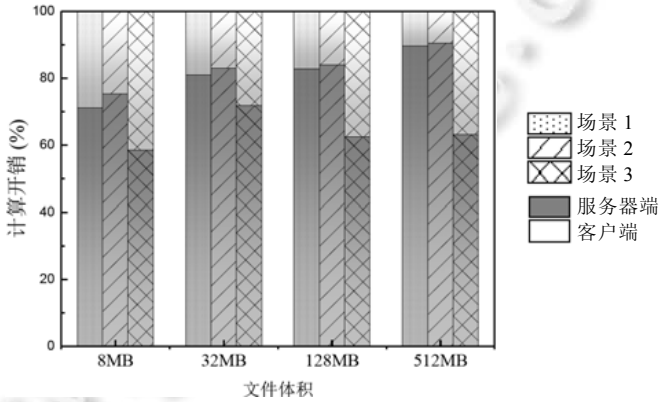


Fig.5 Computational cost of client and server in defferent scenarios

图 5 不同场景下客户端与服务器的计算开销

我们比较了 VSDD 与其他代表性方案的计算性能,共进行了 3 组模拟实验,比较了不同场景下,上传 10MB 数据过程中各个阶段所需的时间开销,实验结果如图 6~图 8 所示.图中,PoW 表示数据所有权证明操作.

由实验结果可见,VSDD 方案的时间开销明显小于 perfectDedup 方案.这是因为 perfectDedup 方案需要根据密文,判断数据是否重复以及进行数据的查找,所耗费的时间较长.此外,在 perfectDedup 方案中,用户需要与实时在线的第三方进行交互,以确定数据流行度.而在 VSDD 方案中,后继上传者根据云端返回的数据签名,即可确定数据流行状态,并完成对流行度查询结果的验证.文献[20]和文献[25]中的方案不区分数据流行度,所有数据均视为隐私数据,都使用较为复杂的加密算法进行保护.VSDD 方案根据数据的流行度采用不同的数据加密方式,可显著降低所需的时间开销,提高系统的执行效率.

我们与 perfectDedup 方案、文献[20]中的方案以及文献[25]中的方案进行了总体时间开销的比较.在不同场景下,将不同体积的数据上传至云服务器,统计其所用的平均时间开销,实验结果如图 9~图 11 所示.

当数据体积较小且为非流行数据时,4 个方案所需的总体时间开销相近.随着数据体积的增大,VSDD 方案所需的时间开销明显小于其他方案.这是因为 VSDD 方案对数据标识和数据签名进行了优化设计.而文献[20]中的方案需要由云服务器对 RCE 加密的密文进行外层加密,文献[25]中的方案需要对数据和密钥进行分块处理,并与多个服务器进行交互.此外,在 VSDD 方案中,当数据流行度状态为流行数据时,用户仅需将数据标识和签名上传即可.实验结果表明,尽管本方案因为引入数据签名而产生了一定的时间开销,但整体性能仍优于其他的方案.

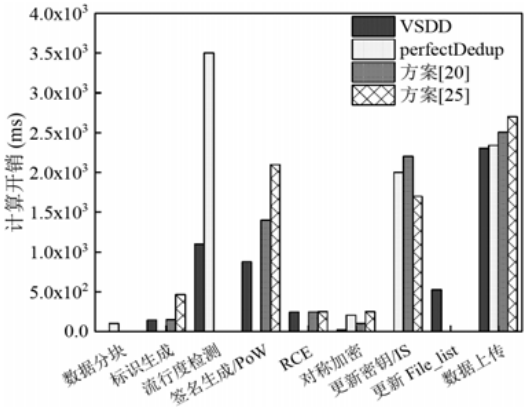


Fig.6 Comparison of computational cost (10MB, unpopular data)

图 6 计算开销对比(10MB,非流行数据)

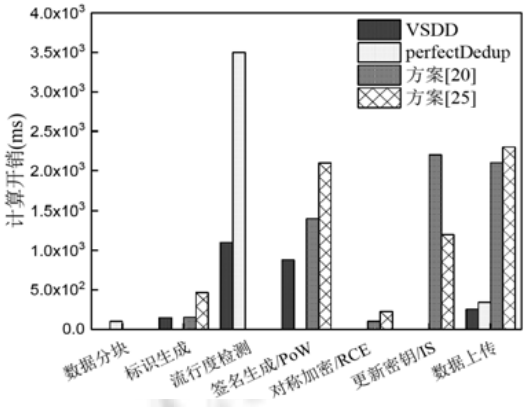


Fig.7 Comparison of computational cost (10MB, popular data)

图 7 计算开销对比(10MB,流行数据)

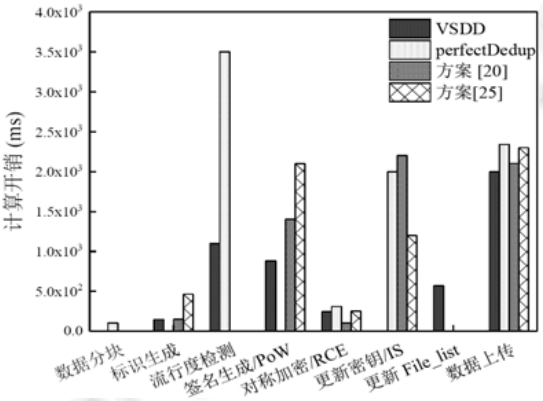


Fig.8 Comparison of computational cost (10MB, popularity transition data)

图 8 计算开销对比(10MB,流行度转换数据)

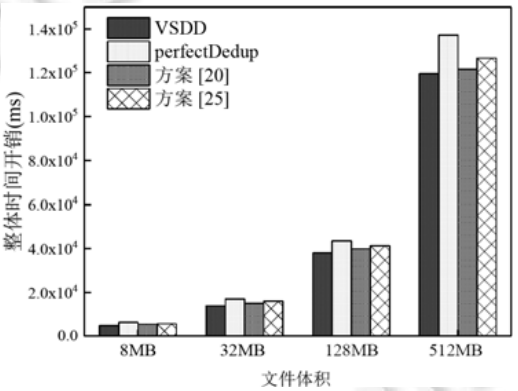


Fig.9 Comparison of overall time span (unpopular data)

图 9 整体时间开销比较(非流行数据)

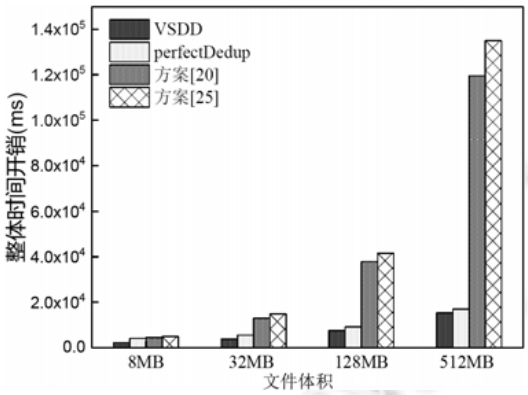


Fig.10 Comparison of overall time span (popular data)

图 10 整体时间开销比较(流行数据)

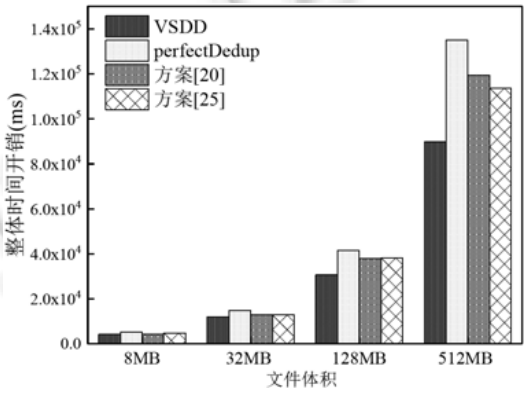


Fig.11 Comparison of overall time span (popularity transition data)

图 11 整体时间开销比较(流行度转换数据)

5.3 方案特点比较

我们将 VSDD 方案与其他代表性方案进行了比较,结果见表 5.

Table 5 Scheme charateristics comparison

表 5 方案特点比较

方案	VSDD	perfectDedup	文献[1]	文献[13]	文献[14]	文献[20]	文献[25]
无实时在线第三方	√	×	√	×	×	√	×
划分数据流行度	√	√	×	×	√	×	×
对非流行数据进行删重	√	×	×	×	√	×	×
所有权证明	√	×	×	×	×	√	√

6 总结与展望

本文设计了可验证的文件级客户端数据删重方案.VSDD 方案基于双线性映射性质,利用双文件标识进行数据的查找和流行度的查询.采用改进的群签名方案实现对数据的签名,防止云服务器伪造流行度查询结果.方案利用广播加密技术实现加密密钥的安全存储和传递,基于多层加密系统对数据进行加密,将数据分为流行数据与非流行数据,确保了非流行数据的安全,能够抵御来自 CSP 的穷举攻击.与同类方案相比,VSDD 方案不借助第三方服务器,实现了客户端的数据删重以及数据签名的可验证,节约了大量的网络带宽,用户可对云端操作的正确进行判断和验证.文中给出了详细的安全性分析,证明了 VSDD 方案的安全性.仿真实验数据表明,VSDD 方案可实现在半信任的云服务器上安全、高效的数据存储.

如何摆脱广播中心,实现加密密钥的分发以及对流行度划分方法的细化,是我们下一步研究的重点.

References:

[1] Liu J, Asokan N, Pinkas B. Secure deduplication of encrypted data without additional independent servers. In: Proc. of the 22nd ACM SIGSAC Conf. on Computer and Communications Security. New York: ACM, 2015. 874−885.

[2] Bellare M, Keelveedhi S, Ristenpart T. DupLESS: Server-aided encryption for deduplicated storage. In: Proc. of the Usenix Conf. on Security. USENIX Association, 2013. 179−194.

[3] Zhang SG, Xian HQ, Wang YZ, Liu HY, Hou RT. Secure encrypted data deduplication method based on offline key distribution. Ruan Jian Xue Bao/Jornal of Software, 2018,29(7):1909−1921 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5359.htm [doi: 10.13328/j.cnki.jos.005359]

[4] Meyer DT, Bolosky WJ. A study of practical deduplication. ACM Trans. on Storage (TOS), 2012,7(4):1−20.

[5] Baracaldo N, Androulaki E, Glider J, Sorniotti A. Reconciling end-to-end confidentiality and data reduction in cloud storage. In: Proc. of the 6th Edition of the ACM Workshop on Cloud Computing Security. New York: ACM, 2014. 21−32.

[6] Storer MW, Greenan K, Long DDE, Miller EL. Secure data deduplication. In: Proc. of the 4th ACM Int'l Workshop on Storage Security and Survivability. New York: ACM, 2008. 1−10.

[7] Koo D, Hur J. Privacy-preserving deduplication of encrypted data with dynamic ownership management in fog computing. Future Generation Computer Systems, 2018,78(2):739−752.

[8] Liu JF, Wang JF, Tao XL, Jian S. Secure similarity-based cloud data deduplication in Ubiquitous city. Pervasive and Mobile Computing, 2017,41:231−242.

[9] Fu YX, Luo SM, Shu JW. Survey of secure cloud storage system and key technologies. Journal of Computer Research and Development, 2013,50(1):136−145 (in Chinese with English abstract).

[10] Xu J, Chang EC, Zhou JY. Weak leakage-resilient client-side deduplication of encrypted data in cloud storage. In: Proc. of the 8th ACM SIGSAC Symp. on Information, Computer and Communications Security. New Yrok: ACM, 2013. 195−206.

[11] Yan Z, Ding WX, Yu XX, Zhu HQ, Deng RH. Deduplication on encrypted big data in cloud. IEEE Trans. on Big Data, 2016,2(2): 138−150.

[12] Srinivasan K, Bisson T, Goodson GR, Voruganti K. iDedup: Latency-aware, inline data deduplication for primary storage. In: Proc. of the 10th USENIX Conf. on File and Storage Technologies. USENIX Association, 2012. 1−14.

[13] Bellare M, Keelveedhi S, Ristenpart T. Message-locked encryption and secure deduplication. In: Proc. of the Annual Int'l Conf. on the Theory and Applications of Cryptographic Techniques. Berlin, Heidelberg: Springer-Verlag, 2013. 296−312.

[14] Stanek J, Sorniotti A, Androulaki E, Kencl L. A secure data deduplication scheme for cloud storage. In: Proc. of the Int'l Conf. on Financial Cryptography and Data Security. Berlin, Heidelberg: Springer-Verlag, 2014. 99−118.

[15] Puzio P, Molva R, Önen M, Loureiro S. PerfectDedup: Secure data deduplication. In: Proc. of the Int'l Workshop on Data Privacy Management. Springer Int'l Publishing, 2015. 150−166.

[16] Lou DC, Huang HF. Efficient three-party password-based key exchange scheme. Int'l Journal of Communication Systems, 2011, 24(4):504−512.

[17] Hu XX, Zhang ZF, Liu WF. Universal composable password authenticated key exchange protocol in the standard model. Ruan Jian Xue Bao/Jornal of Software, 2011,22(11):2820−2832 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3910.htm [doi: 10.3724/SP.J.1001.2011.03910]

[18] Zhang P, Huang P, He X, Wang H, Zhou K. Resemblance and mergence based indexing for high performance data deduplication. Journal of Systems and Software, 2017,100(128):11−24.

[19] Halevi S, Harnik D, Pinkas B, Shulman-Peleg A. Proofs of ownership in remote storage systems. In: Proc. of the 18th ACM Conf. on Computer and Communications Security. New York: ACM, 2011. 491−500.

[20] Hur J, Koo D, Shin Y, Kang K. Secure data deduplication with dynamic ownership management in cloud storage. IEEE Trans. on Knowledge and Data Engineering, 2016,28(11):3113−3125.

[21] Rafaeli S, Hutchison D. A survey of key management for secure group communication. ACM Computing Surveys (CSUR), 2003, 35(3):309−329.

[22] Cui H, Deng RH, Li YJ, Wu GW. Attribute-based storage supporting secure deduplication of encrypted data in cloud. IEEE Trans. on Big Data, 2017, Early-Access. [doi: 10.1109/TBDATA.2017.2656120]

[23] Cheng SJ, Zhang CH, Pan SQ. Design on data access control scheme for cloud storage based on CP-ABE algorithm. Netinfo Security, 2016,16(2):1−6 (in Chinese with English abstract).

[24] Zhang Y, Xu CX, Li HW, Yang K, Zhou JY, Lin XD. HealthDep: An efficient and secure deduplication scheme for cloud-assisted eHealth systems. IEEE Trans. on Industrial Informatics, 2018,14(9):4101−4112.

[25] Singh P, Agarwal N, Raman B. Secure data deduplication using secret sharing schemes over cloud. Future Generation Computer Systems, 2018,88:156−167.

[26] Bini SP, Abirami S. Proof of retrieval and ownership for secure fuzzy deduplication of multimedia data. In: Proc. of the Progress in Computing, Analytics and Networking. Singapore: Springer-Verlag, 2018. 245−255.

[27] Miller VS. The Weil pairing, and its efficient calculation. Journal of Cryptology, 2004,17(4):235−261.

[28] Boneh D, Boyen X. Short signatures without random oracles and the SDH assumption in bilinear groups. Journal of Cryptology, 2008,21(2):149−177.

[29] Xie WJ, Zhang Z. Efficient and provably secure certificateless signcryption from bilinear maps. In: Proc. of the 2010 IEEE Int'l Conf. on Wireless Communications, Networking and Information Security (WCNIS). IEEE, 2010. 558−562.

[30] Sakai R, Furukawa J. Identity-based broadcast encryption. IACR Cryptology ePrint Archive, 2007. https://eprint.iacr.org/2007/217

[31] Delerablée C. Identity-based broadcast encryption with constant size ciphertexts and private keys. In: Proc. of the Int'l Conf. on the Theory and Application of Cryptology and Information Security. Berlin, Heidelberg: Springer-Verlag, 2007. 200−215.

[32] Pang LJ, Li HX, Jiao LC, Wang YM. Design and analysis of a provable secure multi-recipient public key encryption scheme. Ruan Jian Xue Bao/Journal of Software, 2009,20(10):2907−2914 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3552.htm [doi: 10.3724/SP.J.1001.2009.03552]

[33] Hwang JY, Lee S, Chung BH, Cho HS, Nyang D. Short group signatures with controllable linkability. In: Proc. of the 2011 Workshop on Lightweight Security & Privacy: Devices, Protocols and Applications (LightSec). IEEE, 2011. 44−52.

[34] Loukides MK, Oram A. Programming with GNU software. O'Reilly Media, Inc., 1997.

[35] De Caro A, Iovino V. jPBC: Java pairing based cryptography. In: Proc. of the 2011 IEEE Symp. on Computers and Communications (ISCC). IEEE, 2011. 850−855.

附中文参考文献:

[3] 张曙光,咸鹤群,王雅哲,刘红燕,侯瑞涛.基于离线密钥分发的加密数据重复删除方法.软件学报,2018,29(7):1909−1921. http://www.jos.org.cn/1000-9825/5359.htm [doi: 10.13328/j.cnki.jos.005359]

[9] 傅颖勋,罗圣美,舒继武,安全云存储系统与关键技术综述.计算机研究与发展,2013,50(1):136−145.

[17] 胡学先,张振峰,刘文芬,标准模型下通用可组合的口令认证密钥交换协议.软件学报,2011,22(11):2820−2832. http://www.jos.org.cn/1000-9825/3910.htm [doi: 10.3724/SP.J.1001.2011.03910]

[23] 程思嘉,张昌宏,潘帅卿.基于 CP-ABE 算法的云存储数据访问控制方案设计.信息网络安全,2016,16(2):1−6.

[32] 庞辽军,李慧贤,焦李成,王育民.可证明安全的多接收者公钥加密方案设计与分析.软件学报,2009,20(10):2907−2914. http://www.jos.org.cn/1000-9825/3552.htm [doi: 10.3724/SP.J.1001.2009.03552]

咸鹤群(1979－),男,山东青岛人,博士,副教授,CCF 高级会员,主要研究领域为云存储安全,区块链,隐私保护,密码学.

张曙光(1991－),男,硕士,主要研究领域为云存储安全,区块链,隐私保护.

刘红燕(1994－),女,硕士,主要研究领域为云存储安全,隐私保护.

侯瑞涛(1993－),男,硕士,主要研究领域为云计算安全,隐私保护.