面向隐私保护的数据块调整机制

史玉良 陈 玉 孙世彬 崔立真

(山东大学计算机科学与技术学院 济南 250101)

摘 要 在云计算环境下,通过分块混淆的隐私保护机制,将租户的数据分成多个数据块,并且存储到不同的数据节点上,以此实现数据的隐私保护.虽然该方法可以实现在明文状态下保护租户数据的隐私安全,但在实际环境中,由于租户的隐私需求、数据需求是可变的,导致云端底层的数据块结构和存储位置发生变化,因此在这种隐私保护机制下依然存在隐私泄露的风险.所以该文基于分块混淆隐私保护方法,提出一种面向隐私保护的数据块调整机制.该机制首先根据租户更新后的隐私约束,基于少动性原则,对原始的隐私保护策略中违背隐私约束的数据块进行分割;然后再结合隐私约束,重组数据块,并生成隐私保护调整策略;由于数据块分割结果的多样性,导致最终生成的可行隐私保护策略并不唯一,所以该文最后综合隐私需求、性能需求、负载需求和不对等均衡,提出了一种基于全局最优的隐私保护策略选择算法,实现从多种可行策略中筛选出满足所有要求的最优调整策略.实验结果表明,该文提出的数据块调整机制,可以找到一种最优的隐私保护调整策略,并且满足系统的性能和负载要求,增强租户数据的隐私保护效果.

关键词 云计算;数据块;隐私保护;数据调整;负载能力;最优调整策略

中图法分类号 TP311 **DOI 号** 10.11897/SP.J.1016.2017.02719

Data Chunks Adjustment Mechanism for Privacy Protection

SHI Yu-Liang CHEN Yu SUN Shi-Bin CUI Li-Zhen

(*School of Computer Science and Technology, Shandong University, Jinan* 250101)

Abstract In the cloud computing environment, the data of the tenant are divided into several data chunks and these data chunks are stored on different data nodes by means of the privacy protection mechanism based on chunk-confusion, in order to realize the privacy protection of data. Although this method can realize the privacy protection of the tenant's data in the plaintext state, due to tenant's privacy needs and data demands are variable, the underlying data chunks structure and storage location in the cloud will change, which makes there still exist a risk of leakage of privacy under the privacy protection mechanism based on chunk-confusion. For example, the store mode of data chunk is {post-code, age, disease}, which means that these three properties can be placed together and it will not reveal the privacy of the tenant' data. When the tenant suddenly adds a new attribute (the worker number) in the store mode of data chunk, if an attacker gets information about this chunk and happens to know an patient's the worker number, then this attacker may know the patient's condition, which is what tenant do not want to see. Therefore, based on chunk-confusion based the mechanism of privacy protection, the paper proposed a kind of mechanism of data chunks adjustment for privacy protection. In the first place, according to the privacy constraints updated by the tenant, the mechanism splits data chunks that go against

收稿日期:2016-06-28;在线出版日期:2017-05-23.本课题得到山东省泰山产业领军人才工程专项经费(tscy20150305)、山东省重点研发计划(2016GGX101008,2016ZDJS01A09)、山东省自然科学基金重大基础研究项目(ZR2017ZB0419)资助.**史玉良**,男,1978 年生,博士,教授,中国计算机学会(CCF)会员,主要研究领域为云计算、数据库、隐私保护. E-mail: shiyuliang@sdu.edu.cn. **陈 玉**,男,1990 年生,硕士研究生,主要研究方向为云计算、隐私保护. **孙世彬**,男,1989 年生,硕士研究生,主要研究方向为云计算、隐私保护. **崔立真**,男,1976 年生,博士,教授,中国计算机学会(CCF)会员,主要研究领域为数据科学与工程、服务计算、智能协同计算.

the privacy constraints in the original privacy protection strategy. In the process of splitting data chunks, the data storage structure of data chunks is kept unchanged as far as possible based on the less dynamic principle to reduce the cost of data transfer and the adjustment cost of privacy protection strategy. In the second place, combined with the privacy constraints, data chunks that do not violate the privacy constraints are reassembled into the new data chunks to enhance the processing efficiency of data and to generate the privacy protection adjusting strategy. In addition, because of the diversity of the data chunks segmentation results, the resulting feasible privacy strategies are not unique. Therefore, considering the privacy requirements, performance requirements, load demands and unequal equalization, the paper proposed an optimal algorithm to select the optimal adjustment strategy to meet all requirements, realizing when the data chunks are adjusted, it can not only reduce the performance and load capacity of the system, but also enhance the effect of privacy protection. Finally, the adjustment mechanism completes the data migration and placement according to the optimized adjustment strategy of privacy protection. In order to verify the effectiveness and performance of the adjustment mechanism of data chunks based on privacy protection proposed in this paper, we designed the simulation experiment from the point of view of the accuracy and privacy of optimal privacy protection strategy, system performance and load, etc. Experimental results prove that the privacy protection adjustment mechanism proposed in this paper can not only find an optimal strategy for privacy protection, but also meet the system's performance and load requirements, and enhances the privacy protection effect of the tenant's data.

Keywords cloud computing; data chunk; privacy protection; data adjustment; load capacity; optimal adjustment strategy

1 引 言

随着云计算技术的快速发展、云产品的推广和普及,云计算安全问题逐渐成为人们关注的焦点.云计算技术给我们的生活、生产带来很多便利,但同样带来很多问题,如果处理不当,将会造成巨大损失.近几年,云安全事故频频发生.2014 年①,12306 铁路客户服务中心网站爆发用户账号串号问题,大量用户身份证等信息遭泄露;同年,韩国发生史上最大规模的信用卡个人信息泄露事件,KB 国民卡和 NH 农协卡等公司的一亿多条用户信息被泄露.2015 年②,Anthem 数据泄露事件,超过 8000 万条病患与员工记录信息遭到曝光;同年 5 月,美国人事管理办公室遭遇数据泄露,超过 2150 名联邦政府雇员的私人信息包括其社保号码、犯罪及金融记录等内容被泄露.2016 年③,土耳其发生 5000 万公民信息泄露事件;同月,美国最大的电信运营商 Verizon 的 150 万客户记录遭泄露.数据信息的泄露,不仅给当事人和公司造成了诸多问题和损失,也时刻警惕着人们——数据安全很重要.

针对云计算中的数据隐私安全问题,研究者们已经提出了很多解决方案.文献[1]在 DaaS(Database as a Service)模式下提出一种基于隐私保护的桶划分算法,该算法将信息泄露的隐私指标体系与查询效率结合,通过基于遗传算法的桶划分算法对隐私与效率的模型进行优化,从而降低密文查询中隐私泄露的概率.然而此方法只是解决了查询过程中的隐私泄露问题.差分隐私保护技术是数据分布中常用的隐私保护方法,可以有效地保护非交互式模式下的数据隐私.但是如果有两个组织同时拥有同一个人的多个不完全相同属性的信息(例如银行和保险公司同时拥有客户信息),则一旦两家公司进行数据整合,一些用不到的信息也会发布出去,客户的隐私就可能泄露.文献[2-3]对差分隐私保护模型进行了调整,提出了两段垂直划分的隐私保护方法,该方法将发布的信息分成两块,一块不包含敏感属性,一块包含敏感属性.在交互机制下,各个组织只需要提交第一块属性和数据即可,这种方法对于交

① http://www.freebuf.com/articles/database/54631.html
② http://cloud.idcquan.com/yaq/82751.shtml
③ http://www.leagsoft.com/news/p/1443

互机制下的隐私保护很有效，但是 SaaS(Software as a Service)下，它并不能完全满足租户的隐私需求.为了高效地保护云中数据安全，文献[4]提出了分布式数据分块云存储模型(DDFM).模型中，层-层的保护策略，不仅可以提高数据安全，也可以提高数据灵敏性，降低费用，保证数据完整性；基于粒度计算的数据分块算法，将数据分成块，然后加密处理，再通过授权，最后存储到云中.DDFM 在对数据分块时，使用了加密数据技术，这将会影响数据的处理效率.

上述的分块和调整方法，虽然可以在一定条件下保护云中数据的安全，但是对于我们研究的 SaaS 环境下的隐私保护，它们并不适合.因此，本文前期工作[5-7]提出了一种基于分块混淆的隐私保护机制.该机制通过构造三方(可信第三方、云服务提供商和租户)交互模型，实现在明文状态下保护租户的数据隐私.首先，根据租户定制的隐私约束，将租户提交的数据分成多个数据块，使得在同一个隐私约束中的属性分布到不同的数据块中，即租户身份信息和敏感信息分割开来.然后隐私保护模块与第三方协作，通过乘法同态加密为同一条记录的不同数据块中的数据生成一个数据切片标识，从而混淆数据块之间的关系，并把这些关系存储到可信第三方，以便后续的数据重构；数据关系经过混淆后，攻击者很难发现不同数据块中记录的对应关系，但当数据块中的数据分布不均匀时，攻击者仍可能以较大的概率猜中某个个体的隐私信息，因此文献[6]提出了 α、β、γ 三种均衡化机制，通过添加伪造数据，保证数据隐私泄露的概率不会超过 $\max(1/n,\beta^k,\gamma^k)$($n$ 为记录数，k 为块的个数).最后，再根据各个数据节点的计算能力、存储能力和负载能力等，将数据块存储到节点上.为了减少数据连接次数，提高数据的访问效率，我们在前期工作中提出了数据块生成策略的优化算法[7]，将关联度高且不会泄露租户隐私的属性划分到同一个数据分块中.通过我们的分块混淆机制，可以实现在明文状态下保护租户数据的隐私安全.

然而，租户的需求是动态变化的，如果租户改变了隐私需求和数据需求，对应的隐私保护策略就会发生改变，以至于底层云存储中的数据模式和数据块的位置也会变化，进而可能破坏原有的数据分布均衡，破坏数据块间的关联关系，泄露租户的数据隐私.对此文献[8]构建了一种数据分块调整模型，模型中节点代表数据块，边代表操作，这样就将调整问题转化成了求无向图中原点到目的节点的最短路径问题.然而该模型只考虑了隐私需求的变化对数据隐私的影响，并没有考虑到负载和性能的需求，而且模型假设已经知道了调整策略，但是实际调整策略有很多种，无法直接确定.因此本文针对租户隐私需求和数据需求的变化，提出一种面向隐私保护的数据块调整机制.首先根据更新后的隐私约束，对原始的隐私保护策略中违背隐私约束的数据块进行分割，拆分过程中，保持数据块中原有结构不变，以减少数据迁移的开销和隐私保护策略调整的代价；然后结合隐私约束，将不违背隐私约束的数据块重新组合成新的数据块，以提升数据的处理效率，并生成隐私保护调整策略；由于数据块分割结果呈多样性，导致最终生成的可行隐私保护策略并不唯一，所以本文最后综合隐私需求、性能需求、负载需求和不对等均衡，提出一种基于全局最优的隐私保护策略选择算法，实现在数据块调整的同时不降低系统的性能和负载能力，增强隐私保护效果.

本文第 2 节介绍相关研究；第 3 节描述相关的定义和调整机制的框架模型；第 4 节介绍隐私保护调整机制的具体实现；第 5 节介绍相关实验和分析；最后对本文进行简要的总结.

2　相关工作

随着密码学的快速发展，数据加密技术已逐渐成熟.文献[9]使用阈值加密算法将数据所有者分成若干用户组，并且给每个用户组一个单独的密钥用于解密数据，用户组间的成员共享密钥的部分内容.这种方案不仅加强了数据保密性，而且还降低了密钥的数量.基于身份定时释放加密算法(ID-TRE)和分布式哈希表网络(DHT)，文献[10]提出了全生命周期的敏感数据隐私保护方案.该方案首先将敏感信息加密为密文，并使用抽取算法提取密文和封装密文；然后使用 ID-TRE 算法加密密钥，并把密钥的密文和敏感信息的密文组合起来，生成一个密文段；最后将密文段分发到 DHT 网络中，将封装好的密文存储到云服务器上.通过这种对数据和密钥双层加密的全周期敏感信息保护，可以保证数据在整个上传、处理和存储过程的隐私安全.文献[11]利用“理想格”的数学对象模型构造了隐私同态算法，使租户可以充分地操作加密状态的数据，然而同态加密方法对数据处理性能有较大影响.文献[12]使用部署在云中的密码协同处理器作为可信处理器，

在密码协同处理器内部处理租户的敏感数据和受保护程序,防止隐私泄露,但是这种隐私保护方法增大了数据处理的开销.文献[13]针对医疗数据,提出一种记录共享的云计算隐私保护方案.该方案首先基于医疗记录的属性分类,垂直分割医疗数据集,将不同的医疗数据分开,分别考虑隐私保护问题;然后通过分析,把不需要隐私保护的数据集合并起来;最后将统计学与密码学结合起来,使用混合搜索技术访问明文和密文,实现医疗数据应用和隐私保护之间的多种范式平衡.然而这种隐私保护方法并没有考虑到需求动态性的问题.文献[14]融合具有高编码效率的 Huffman 编码和具有数据存储优势的布鲁姆过滤器,并且结合现存的安全加密方法,实现了 DaaS 模式下基于隐私保护的模糊关键字查询处理.但是这种隐私保护方法只是对于查询操作的,并不适合数据存储的隐私保护.文献[15]介绍了两种面向发布的数据隐私保护技术,即 k 匿名技术和 l-多样性匿名技术,这两种方法通过泛化、抑制等数据处理,将用户的敏感性隐藏起来,实现用户的隐私保护.但是它们都是对静态、面向数据发布的隐私保护,而实际的云计算环境中,我们需要的是面向存储的隐私保护技术,并且还要适应动态变化.针对事务数据库隐私保护发布的数据安全性与效用性不足问题,文献[16]提出一种有效的满足差分隐私约束的事物数据发布策略.该策略首先基于项集 I 构建了事务数据库 D 的完整树,然后基于压缩感知技术对完整树添加满足拆分隐私约束的噪音,从而得到含噪声的树,最后再在这颗树上执行频繁项集挖掘任务.针对人体传感器网络中的敏感数据,文献[17]提出一种差分隐私保护方法.该方法首先构造一个树型存储结构,以减少数据在传输过程中发生的错误,并且提供远程查询;然后通过哈尔小波①转换方法,将直方图转为完全二叉树,进而实现敏感数据的隐私保护.上诉两种方法都是基于差分隐私保护的,因此存在着数据失真的问题.文献[18]分析了现存的不同分布式存储系统,提出使用数据分块方法提高系统的安全性,实现最小化的加密,保障数据安全的同时,提升数据处理效率.加密技术可以高效地保护数据隐私,但是对于查询来说,代价昂贵.对此,文献[19]将加密技术与数据分块技术相结合,提出了新的解决方案.该方案根据用户的保密需求,将数据分成了多个数据块,并通过加密机制保护它们的关联关系;此外使用启发式算法确定数据的分块数.数据分块技术实现了敏感数据的分离和保护,在隐私保护中起到了重要作用.

综上可知,文献[9-14]基于数据加密技术,实现隐私保护,但是加密技术在云计算环境中,效率并不理想,且资源消耗大;文献[15]介绍了两种数据发布中的匿名隐私保护技术,虽然可以保护数据隐私,但是对于 SaaS 中的数据存储隐私保护,并不完全适用;文献[16-17]基于差分隐私保护,提出了对应的隐私保护方法,然而数据失真问题并不能解决;文献[2-4,8,18-19]根据差分隐私保护和加密机制,提出了数据分块思想,对原有的隐私保护方法进行了调整,在保护数据隐私的同时,增强了数据的可用性.上诉的隐私保护和调整方法,可以在特定的环境下实现隐私保护,为本文的研究提供了很好的学术价值.本文在前期的研究基础之上,提出了面向隐私保护的数据块调整机制,可以根据租户需求的变化,动态地调整数据块和隐私保护策略,满足租户调整需求,保护租户隐私.

3 隐私保护调整模型

3.1 相关定义

定义 1. 隐私约束(Privacy Constraints,PC),对于租户属性集合 $AS=\{A_1,A_2,\cdots,A_n\}$,若存在 AS 的非空子集 $AS_{pc}=\{C_1,C_2,\cdots,C_m\}$,其中 AS_{pc} 中包含租户的隐私信息,且 AS_{pc} 所有属性的组合为可以唯一确定一条租户敏感信息的最小集合,或者可以确定隐私信息的分布规律,则称 AS_{pc} 为一条隐私约束.例如属性集合 $AS_{pc1}=\{$姓名,疾病$\}$,该集合可以唯一确定哪个病人患了哪种疾病,所以 AS_{pc1} 就是一个隐私约束.

定义 2. 隐私保护策略(Privacy Protection Strategy,PPS),根据 PC,将租户的属性集合 AS 分割成 k 个属性集合,即$\{AS_1,AS_2,\cdots,AS_k\}$,且 $AS_i\cap AS_j=\varnothing$, $\forall pc_i,pc_j\in PC\rightarrow pc_j\not\subset AS_i$ 且 $pc_i\subset AS_i$ $(0<i,j\leq k,0<k\leq n)$.

定义 3. 不对等均衡(Unequal Equalization,UE),在调整过程中,要遵循不对等均衡,即保证敏感属性所在的数据块与其他数据块之间的数据记录不存在一对一关系.

在数据调整的过程中可能会出现如表 1、表 2 所示的情况.如表 1 所示,根据 PPS 将姓名和疾病

① http://www.360doc.com/content/13/0925/12/10724725_316957631.shtml

分别存到两个数据块中，以实现病人信息的隐私保护. 为了便于表示和理解，表中数据块内数据顺序存储. 调整前租户认为姓名、年龄和身高可以放在一起，不会泄露隐私. 通过对数据的分析以及客户的反映，租户认为，年龄不能和姓名、身高放在一起. 根据租户的要求进行调整，其中一种调整方案如表 2 所示. 观察表 2 发现，如果攻击者知道 Ross 的年龄，就可以唯一确定 Ross 患了感冒，这是租户不希望的结果. 出现隐私泄露的原因是，调整前，年龄和姓名属于同一个数据块，可以存在一对一的关系，且不会泄漏隐私，但调整后，这种一对一的关系直接造成了数据关系的泄露，因此表 2 方案并不可取.

表 1　调整前两个数据块的信息

姓名	身高	年龄	性别	疾病
Ben	173	25	Male	Cold
Angelia	168	40	Female	Fever
Iris	170	40	Female	Cancer
Ross	170	30	Female	Fever
Tom	174	25	Male	Toothache

表 2　调整后两个数据块的信息

姓名	身高	性别	疾病	年龄
Ben	173	Male	Cold	25
Angelia	168	Female	Fever	40
Iris	170	Female	Cancer	40
Ross	170	Female	Fever	30
Tom	174	Male	Toothache	25

定义 4.　负载能力(Load Capacity，LC)，指单位时间内节点处理数据的能力. 对于每个节点，都会有一个阈值 TD_{lc}. 如果实际测量值超过 TD_{lc}，表示当前负载超过了节点的负载能力.

节点的负载能力与 CPU 使用率(CPU_{usage})、内存占用率(MEM_{usage})、IO 负载(IO_{load})和磁盘存储率($DISK_{\text{stt}}$)有关，本文给出 LC 的计算公式，如式(1)所示.

$$LC = c \times CPU_{\text{usage}} + m \times MEM_{\text{usage}} + io \times IO_{\text{load}} + d \times DISK_{\text{stt}} \tag{1}$$

式中，c、m、io 和 d 是 4 个参数的影响因子，它们的大小与每台节点服务器的配置有关. 4 个参数的值可以从节点的历史监控信息中获取. 本文根据实际需求，取当前时间的前 3 min 数据的平均值作为 4 个参数的计算值计算当前节点的 LC. 由于节点的负载能力是动态变化的，所以需要根据当前的节点负载记录，预测数据调整之后的节点负载能力是否满足要求. 贝叶斯预测模型[①]是运用贝叶斯统计进行的一种预测. 贝叶斯统计不同于一般的统计方法，其不仅利用模型信息和数据信息，而且充分利用先验信息，因此本文选用贝叶斯模型预测节点的负载能力. 从节点 $node$ 的历史负载数据中，抽取 Y 次不超载的数据和 N 次超载的数据. 已知：$P(Y)=P_y$，$P(N)=P_n$，当前的 4 个参数值分别为 $C_{CPU_{\text{usage}}}$，$C_{MEM_{\text{usage}}}$，$C_{IO_{\text{load}}}$ 和 $C_{DISK_{\text{stt}}}$，调整后的磁盘存储率为 $F_{DISK_{\text{sst}}}$.

对于无属性增加的情况，本文只考虑使用 $F_{DISK_{\text{sst}}}$ 预测节点的负载能力是否满足要求. 根据条件概率公式可知，当磁盘存储率为 $F_{DISK_{\text{stt}}}$ 时，节点负载满足要求的概率 $P(LC)$ 为

$$P(Y/F_{DISK_{\text{stt}}}) = \frac{P(F_{DISK_{\text{stt}}}/Y)P(Y)}{P(F_{DISK_{\text{stt}}}/Y)P(Y)+P(F_{DISK_{\text{stt}}}/N)P(N)} \tag{2}$$

对于有属性增加的情况，假设调整后 $node$ 新增了 w 个属性，它们对 CPU、内存和 IO 的影响集合为 $wset=\{(c_1\times f_1, m_1\times f_1, io_1\times f_1),(c_2\times f_2, m_2\times f_2, io_1\times f_2),\cdots,(c_w\times f_w, m_w\times f_w, io_w\times f_w)\}$. 其中 c_i 表示属性 w_i 对 CPU 使用率的贡献，m_i 表示属性 w_i 对内存占用率的贡献，io_i 表示属性 w_i 对 IO 负载的贡献，f_i 表示 w_i 所在机器与 $node$ 节点机制的性能综合比，$(0<i\leq w)$. 调整后的 CPU 使用率($F_{CPU_{\text{usage}}}$)、内存占用率($F_{MEM_{\text{usage}}}$)、IO 负载($F_{IO_{\text{load}}}$)如下所示：

$$F_{CPU_{\text{usage}}} = \sum_{i=1}^{w} c_i \times f_i + C_{CPU_{\text{usage}}} \tag{3}$$

$$F_{MEM_{\text{usage}}} = \sum_{i=1}^{w} m_i \times f_i + C_{MEM_{\text{usage}}} \tag{4}$$

$$F_{IO_{\text{load}}} = \sum_{i=1}^{w} io_i \times f_i + C_{IO_{\text{load}}} \tag{5}$$

根据条件概率公式可以计算出 $P(Y/F_{CPU_{\text{usage}}})$、$P(Y/F_{MEM_{\text{usage}}})$ 和 $P(Y/F_{IO_{\text{load}}})$ 的概率. 再根据贝叶斯定理可得如下公式：

$$\begin{aligned} P(LC) = {} & P\left(\frac{Y}{F_{CPU_{\text{usage}}}}\right)\times P\left(\frac{Y}{F_{MEM_{\text{usage}}}}\right)\times P\left(\frac{Y}{F_{IO_{\text{load}}}}\right)\times P\left(\frac{Y}{F_{DISK_{\text{stt}}}}\right)\Big/\left(P\left(\frac{Y}{F_{CPU_{\text{usage}}}}\right)\times P\left(\frac{Y}{F_{MEM_{\text{usage}}}}\right)\times P\left(\frac{Y}{F_{IO_{\text{load}}}}\right)\times P\left(\frac{Y}{F_{DISK_{\text{stt}}}}\right)+\left(1-P\left(\frac{Y}{F_{CPU_{\text{usage}}}}\right)\right)\times\left(1-P\left(\frac{Y}{F_{MEM_{\text{usage}}}}\right)\right)\times\left(1-P\left(\frac{Y}{F_{IO_{\text{load}}}}\right)\right)\times\left(1-P\left(\frac{Y}{F_{DISK_{\text{stt}}}}\right)\right)\right) \end{aligned} \tag{6}$$

根据上述的式(2)～式(6)可以计算出一次调整

① https://en.wikipedia.org/wiki/Bayes%27_theorem

后节点负载满足要求的概率.根据经验设定概率阈值为 P_{lc}.如果 $P(LC)$大于 P_{lc},表示满足负载要求,否则超载.

定义 5. 性能需求(Performance Requirements,PR),表示租户对应用准确性、响应时间等的最低要求,为便于描述和计算,本文使用响应时间 t 代表性能需求.假设租户设定的最低要求为 t_{low}($0<t\leq t_{\text{low}}$).响应时间主要与数据查询时间、数据传输时间和数据重构时间有关.数据查询时间在分块机制下,主要与数据块连接次数有关;数据传输时间主要与节点间带宽和本地数据量有关;数据重构时间主要与总的查询记录数和分块数有关.为了便于计算,本文给出了响应时间的计算公式:

$$t=(k-1)\times T_{\text{aver}}+\frac{(Size_{\text{total}}-Size_{\text{local}})}{Bandwidth_{\text{aver}}}+\frac{Num_{\text{total}}}{RT_{\text{aver}}} \tag{7}$$

式中 k 为分块数,T_{aver} 为平均一次连接时间,$Size_{\text{total}}$ 为要查询的总数据量,$Size_{\text{local}}$ 为本地数据量,$Bandwidth_{\text{aver}}$ 为节点的平均带宽,Num_{total} 为总的记录数,RT_{aver} 为平均重构一条记录所用时间.

定义 6. 少动性原则(Less Dynamic Principle,LDP),在分割数据块保护数据隐私时,尽可能地保证数据块结构的原有性.例如,原有数据块 $chunk_1=\{CA_1,CA_2,CA_3,CA_4\}$,某个隐私约束 $PC_1=\{CA_1,CA_3\}$,由此可知 $chunk_1$ 违背 PC_1,所以需要分割 $chunk_1$,分割的结果有很多种,但是根据少动性原则,最终分割的结果为$\{(CA_1),(CA_2,CA_3,CA_4)\}$和$\{(CA_3),(CA_1,CA_2,CA_4)\}$.

3.2 整体架构

在 SaaS 环境下,租户的数据放在非完全可信的服务提供商处,这就导致租户的数据安全无法得到保障.因此我们在前期工作[5-7]中提出了基于分块混淆的隐私机制,可以实现在明文状态下保护租户的数据隐私.但是当租户需求变化时,原有的隐私保护策略,已经不能满足租户的隐私需求了,对此,基于前期的研究,本文提出了面向隐私保护的数据块调整模型,模型的整体架构如图 1 所示.

左侧的最上边是应用层,租户通过它来租赁,管理和维护应用 App,同时提交数据和对应的需求.左侧的中间包括隐私保护的基础模块和可信第三方.在基础模块中,租户首先要根据基础模块与可信第三方通信,完成身份认证,并根据租户提交的隐私需求,定制隐私保护策略 PPS,根据 PPS,将租户的数据分成多个数据块;然后为了隐藏原数据间的关

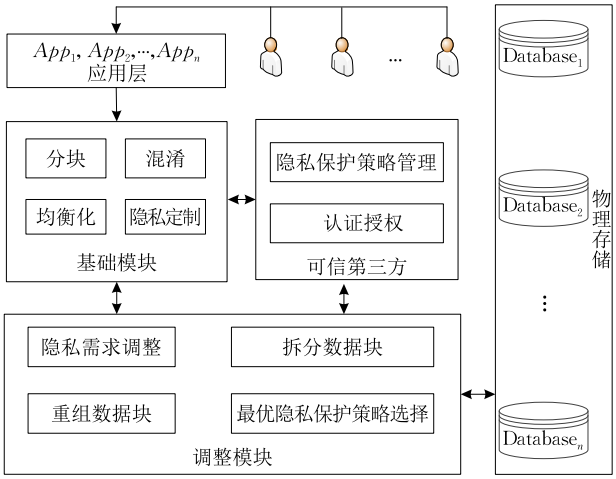

图 1 整体架构

系,基础模块为每条记录都添加了分块编号 sid,并打乱每个数据块内的顺序.在数据块中,敏感属性值的种类和比例也会影响数据的隐私保护效果,所以基础模块使用均衡化机制,通过添加伪造数据的方式确保敏感属性分布均衡;最后将隐私保护策略保存到可信第三方上(本文假设可信第三方是完全可信的,攻击者无法从这里获取到隐私保护策略).左侧最下面是调整模块,租户向基础模块提交了一些新的隐私需求,调整模块首先从基础模块获取这些需求,并从第三方获取 PPS,并对 PPS 进行调整,将违背隐私需求的数据块进行拆分;然后在不违背隐私需求的前提下,再将所有拆分后的数据块进行重新组合,生产多条可选的隐私保护策略;最后根据性能需求、隐私需求、负载需求等,从中选择出最优的一条策略作为新的隐私保护策略,保存到第三方上.模型中隐私保护处理的过程都是在内存中进行的,本文假设内存是安全的.对于隐私保护策略,攻击者无法准确地获取到完整的策略,因此不会影响调整的过程.

3.3 调整模型

前期工作[6]中,采用分块混淆隐私的保护机制,将租户的敏感数据分割成多个数据块,并把数据块存储在多个数据节点上.这种方法,在租户需求不变的时候,可以很好地保护租户的数据隐私.但是现实中,租户的需求(隐私需求,数据需求、性能需求等)都是随着应用的持续使用而发生变化.需求的变化不仅影响云端底层的数据存储结构,还可能导致租户敏感数据的隐私泄露.例如,原有的某个数据块模式为{邮政编码,年龄,病症},表示这 3 个属性放在一起,不会泄露隐私.但是租户突然在这个数据块模

式中增加了一个属性：工号. 这时，如果攻击者获取了这个数据块的信息，并且恰巧知道某个人的工号，则攻击者就知道了这个工号对应病人的患病情况，这是租户不想看到的结果. 因此本文提出了面向隐私保护的数据块调整机制. 该机制可以在满足租户调整需求的同时，完成底层数据结构和存储位置的调整，并且保护租户的敏感信息不被泄露. 隐私保护调整模型如图 2 所示. 上部表示租户需求的变化，包括隐私需求和数据需求，隐私需求表示租户希望对指定的数据属性进行隐私保护，并且要求某些属性不能放在同一个数据块中，在本文中，直接表现形式为隐私约束；数据需求在本文中特指租户对数据属性的操作需求，包括属性的增添和删除，属性名的修改操作，并不会导致隐私泄露，只需要更改隐私保护策略中对应的属性名即可，相对比较简单，所以本文不予考虑. 中间部分为隐私保护调整策略的生成过程. 下部为最优隐私保护调整的选择和数据迁移过程. 对于隐私需求变化，首先获取更新后的所有隐私约束(nPC)；然后根据 nPC，对原有隐私保护策略进行调整，调整过程中，将违背 nPC 的数据块进行分割，分割的时候，要遵循少动性原则；最后对于分割后的数据块，在满足 nPC 的条件下，重新组合数据块. 对于数据需求变化，如果当前是删除属性操作，则无须调整隐私保护策略，直接从隐私保护策略和实际物理存储中删除对应的属性和数据；如果当前是增加属性操作，检测增加的属性与各数据块中原有属性的关联关系，然后根据 nPC 进行数据重组. 由于多个数据块组合的结果并不唯一，所以最终生成的可行 PPS 呈现多样性. 为了从这些可行的 PPS 中选择出最优的调整策略，本文提出了基于全局最优的 PPS 选择算法. 该算法综合考虑隐私需求、性能需求、负载需求和不对等均衡，实现最优策略的选择. 最后模型根据选择出的最优隐私保护调整策略，完成数据的迁移和放置. 下节将介绍隐私保护调整模型的算法实现.

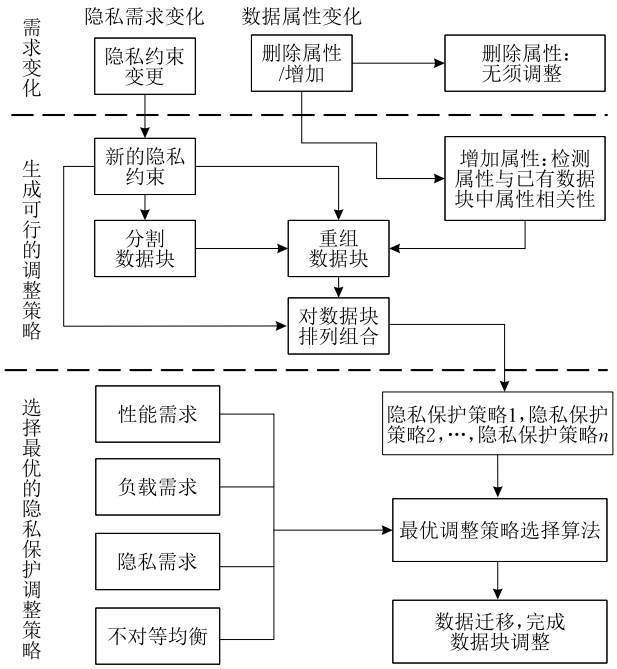

图 2　隐私保护模型

4　隐私保护调整模型的实现

4.1　基于分割重组的隐私保护策略生成算法

租户的需求变化会生成多种可行的隐私保护调整策略，如图 3 所示.

租户 T_1 原有的属性集合为 $\{A_1, A_2, \cdots, A_9\}$，定制的隐私约束 $\{(A_1, A_4, A_7), (A_3, A_8), (A_5, A_6, A_9)\}$，最终生成的隐私保护策略为 $\{(A_1, A_2, A_3), (A_4, A_5, A_6, A_8), (A_7, A_9)\}$. 根据实际需求，租户 T_1 计划了 3 种需求调整方案：第 1 种，对原有的隐私约束进行了更改，要求 A_5 和 A_6 不能放在一起. 同时，增加了一个新的隐私约束 (A_1, A_2)，即要求 A_1 和 A_2 不能放在同一个数据块中，根据新的隐私约束，对原有的隐私保护策略 PPS 进行了调整，调整的过程如图 3 所示，最终生成 16 种满足少动性原则的隐私保护策略；第 2 种，直接删除了一个隐私约束 (A_3, A_8)，因为删除一个隐私约束，不会降低当前系统的性能、负载等，不会破坏原有的隐私保护策略，不会泄露隐私，所以无需调整；第 3 种，只增加了一个隐私约束 (A_5, A_6)，根据新增的隐私约束，调整 PPS，最后生成 4 种满足少动性原则的隐私保护调整策略. 隐私保护调整策略的生成算法如算法 1～算法 3 所示，根据算法，最终可以生成多种满足隐私需求和数据需求的隐私保护调整策略.

算法 1.　数据块分割(Data Chunk Partition, DCP).

输入：$PC_n[]$；//更新的隐私约束

　　$PPS_{old}[]$；//原有隐私保护策略

输出：Chunks；//拆分后的数据块集合

　　Aa (Active attribute)；//活跃属性集合

1.　$Chunks$=null, Aa=null；

2.　FOR (int i=0; i<$PCn.length$; i++)

3.　　FOR (int j=0; j<$PPS_{old}.length$; j++)

4.　　IF $PPS_{old}[j]$ 违背 $PC_n[i]$

5.　　根据$PC_n[i]$和少动性原则计算分割方法种类way；

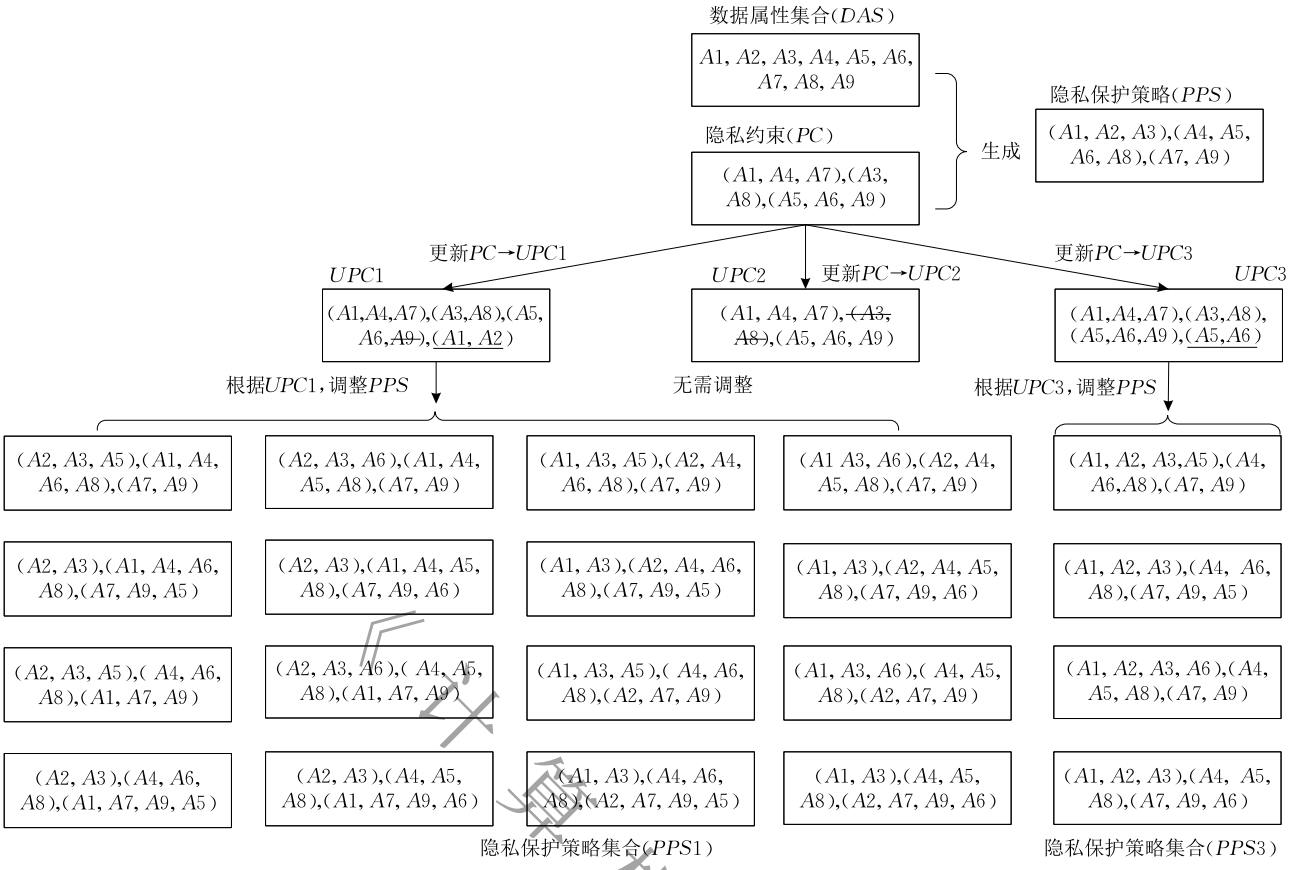

图 3 隐私保护策略生成实例图

```
6.      WHILE (way)
7.       分割 PPS_old[j];
8.       Aa=Aa∪活跃属性;
9.       Chunks=Chunks∪分割后的数据据块;
10.      way--;
11.     END WHILE
12.    ELSE
13.     Chunks=Chunks∪PPS_old[j];
14.    END IF
15. END FOR
16. END FOR
17. Return Aa and Chunks;
```

第 1 行定义了两个集合用于存储分割后的数据块和活跃属性(可以迁移的属性). 2 行和 3 行遍历更新后的隐私约束和原有的隐私保护策略. 4 行～14 行开始分割数据块,其中第 4 行判断当前数据块是否违背当前隐私约束;第 5 行,根据隐私约束和少动性原则计算对当前的数据块有几种分割策略,并将种类赋值给 way;6 行～11 行,分割数据块,根据数据块拆分种类,分别将拆分结果中的活跃属性存入 Aa 中,拆分后的数据块存入 $Chunks$ 中;第 13 行,将不违背隐私约束的数据块存入到拆分后的数据块集合中. 第 17 行,返回所有数据块的分割结果. 算法中包括两个循环,所以时间复杂度为 $O(n^2)$.

算法 1 完成了对违背隐私约束的数据块的分割,算法 2 根据算法 1 的结果,对数据块和属性进行重新组合,并生成可行的隐私保护调整策略.

算法 2. 数据块重组(Data Chunk Reorganization,DCR).

```
输入: Chunks;//拆分后的数据块集合
      Aa: Active attribute;//活跃属性集合
      PC_n[];//更新的隐私约束
输出: Chunks_new;//重新组合后的数据块集合
1.  Chunks_new=null;//存储新生成的数据块
2.  Tmp_chunk=null;//存储临时组合数据块
3.   FOR (int i=0;i<Aa.length; i++)
4.    FOR (int j=0;j<Chunks.length; j++)
5.     IF Aa[i]∉Chunks[j]
6.      Tmp_chunk=Aa[i]∪Chunks[j];
7.      Tag=false;
8.      FOR (int t=0;t<PC_n.length; t++)
9.       IF Tmp_chunk违背 PC_n[t]
10.       Tag=true;
11.       break;
12.      END IF
13.     END FOR
14.     IF !Tag
```

15.　　$Chunk_{new}=Chunk_{new}\cup Tmp_{chunk}$；
16.　　END IF
17.　　END IF
18.　　END FOR
19.　　END FOR
20. Return $Chunks_{new}$；

1 行和 2 行定义了两个变量，用于存储重组后可行的数据块和临时组合未验证的数据块. 3 行和 4 行遍历活跃属性和拆分后的数据块. 5 行～15 行开始重组数据块，第 5 行判断活跃属性是否包含在数据块中，如果不包含，第 6 行将它们合并成一个数据块，临时存入在变量 Tmp_{chunk}；7 行～13 行判断 Tmp_{chunk}是否违背隐私约束 PC_n，如果违背，将变量 Tag 设置为 true；第 14 行判断当前的组合是否违背隐私，如果不违背，则将 Tmp_{chunk}保存到 $Chunk_{new}$. 20 行返回重新组合后的数据块集合. 通过算法 2，将算法 1 生成的活跃属性和数据块，在不违背隐私约束的前提下，重新组合. 由于算法 2 需要遍历活跃属性集合、数据块集合以及隐私约束集合，所以它的效率并不是很高，时间复杂度为 $O(n^3)$. 但是，隐私保护调整机制的算法只有在租户更新需求时才调用，因此，算法 2 的效率是可以接受的.

算法 3.　隐私保护调整策略生成算法(Privacy Protection Adjustment Strategy Generation Algorithm, PPASG).

输入：$Chunks$；//拆分后的数据块集合
　　$Chunks_{new}$；//重新组合后的数据块集合
输出：PPS_{adjust}；//可行的调整策略
1.　$PPS_{adjust}=$null；$TMP_{pps}=$null；
2.　$Chunk_{total}=Chunks\cup Chunks_{new}$；
3.　FOR (int $i=0$；$i<Chunk_{total}.length$；$i++$)
4.　　$TMP_{pps}=Chunk_{total}[i]$；
5.　　FOR (int $j=0$；$j<Chunk_{total}.length$；$j++$)
6.　　IF $i\neq j$ && $Chunk_{total}[i]\cap Chunk_{total}[j]=\varnothing$
7.　　$TMP_{pps}=TMP_{pps}\cup Chunk_{total}[j]$；
8.　　END IF
9.　　END FOR
10.　$PPS_{adjust}=PPS_{adjust}\cup TMP_{pps}$；
11. END FOR
12. Return PPS_{adjust}；

算法 1 和算法 2 生成了拆分后的数据块和重新组合的数据块. 算法 3 根据这些数据块，生成可行的隐私保护调整策略. 1 行和 2 行定义了一些变量. 第 3 行开始遍历所有的数据块. 第 4 行，把当前遍历的数据块赋值给临时变量 TMP_{pps}，继续遍历除了第 4 行选中以外的数据块. 5 行～7 行，如果内外两层循环遍历的不是同一个数据块，且数据块 $Chunk_{total}[i]$和 $Chunk_{total}[j]$ 没有交集，则它们可同时出现在一个隐私保护策略中，把 $Chunk_{total}[j]$并到 TMP_{pp} 中，最后可生成一个包含所有属性的不违背隐私约束的隐私保护调整策略. 第 10 行，把 TMP_{pps} 加入到 PPS_{adjust} 中. 第 12 行返回所有可行的隐私保护调整策略. 算法 3 通过两层的循环，生成隐私保护调整策略，所以它的时间复杂度为 $O(n^2)$.

4.2　基于全局最优的隐私保护策略选择算法

通过算法 1～算法 3，生成了多种不违背隐私约束的调整策略. 但是，对于每个调整策略，它们对系统的性能影响、负载影响以及迁移代价等都是不同的，所以本文设计了算法 4(最优策略选择)，综合考虑各种因素，从调整策略集合中选择出既满足隐私约束的，又对系统影响小，迁移代价低，调整速度快的隐私保护调整策略.

算法 4.　最优策略选择(Optimal Strategy Selection, OSS).

输入：PPS_{adjust}；//隐私保护调整策略集合
　　$PPS_{old}[]$；//原有隐私保护策略
　　$LOAD_{history}$；//节点的历史负载信息
　　$INFOR_{monitor}$；//节点的资源监控信息
　　LOG_{Attr}；//属性操作日志
输出：$PPS_{optimal}$；//最优的隐私保护调整策略
1.　$PPS_{optimal}=$null，$MAX=-1$；
2.　$TMP_{value}=0$，$MAX_{value}=-1$；
3.　FOR (int $i=0$；$i<PPS_{adjust}.size()$；$i++$)
4.　根据不对等均衡，检测调整策略对应的物理数据块存储是否满足要求；
5.　IF 满足要求
6.　　比较 PPS_{old} 和 $PPS_{adjust}.get(i)$，记录下变动的属性和迁移路线；
7.　$LOAD=$false；
8.　　WHILE (每条迁移路线) //计算负载
9.　　根据 $INFOR_{monitor}$ 和式(1)，计算数据迁移前，目标节点的负载能力 $LC_{current}$；
10.　　IF $LC_{current}<TD_{lc}$ //负载表示正常
11.　　根据 $LOAD_{history}$ 和式(2)～式(6)，估算目标节点迁移数据后的负载情况 $P(LC)$；
12.　　IF $P(LC)<P_{lc}$ //预测迁移后的负载
13.　　$LOAD=$true；
14.　　break；
15.　　END IF
16.　　ELSE //当前负载不满足要求
17.　　$LOAD=$true；
18.　　break；

```
19.      END IF
20.    END WHILE//计算负载
21.    IF !LOAD//计算响应时间
22.      根据式(7)计算当前策略的响应时间 t_current;
23.      IF t_current < t_low
24.        TMP_value = (t_current * t') * (P(LC) * lc');
25.      END IF
26.      IF MAX_value < TMP_value
27.        MAX = i;
28.        MAX_value = TMP_value;
29.      ELSE IF MAX_value == TMP_value
30.        根据 LOG_Attr,计算 PPS_adjust.get(i)和 PPS_adjust.
          get(MAX)各自重新组合数据块中属性关联
          度的大小;
31.        IF PPS_adjust.get(i)的关联度大
32.          MAX = i;
33.          MAX_value = TMP_value;
34.        END IF
35.      END IF
36.    END IF
37.  END IF
38. END FOR
39. IF 找不到满足不对等均衡的数据块
40.    使用文献[6]中的伪造数据方法,实现不对等均
       衡,重新执行第 3 行;
    END IF
41. PPS_optimal = PPS_adjust.get(MAX);
42. Return PPS_optimal;
```

算法 4 中 1 行和 2 行,定义变量. 3 行～38 行选择最优的隐私保护调整策略,第 3 行遍历可行的隐私保护策略集合;4 行和 5 行判断是否满足不对等均衡,如果不满足,则继续遍历下一个调整策略,否则继续判断其他条件;第 6 行,比较当前的调整策略和原有策略,记录变动的过程;7 行～20 行判断当前调整策略是否满足负载要求;第 8 行遍历属性迁移路线.第 9 行,根据目标节点的资源监控信息和式(1),计算目标节点当前负载;16 行～18 行,如果负载已经超过节点的最大负载 TD_{lc}(定义 4 中设定的值),则放弃该策略,继续遍历下一个;10 行～15 行,如果没有超载,则根据节点的历史负载信息和式(2)～式(6),预测调整后目标节点的负载情况,如果负载满足要求的概率小于 P_{lc}(定义 4 中设定的值)则放弃该策略;21 行～38 行,如果当前策略满足负载要求,则开始计算性能需求;第 22 行根据式(7)计算当前策略的响应时间. 第 23 行,如果响应时间小于租户设定的最低要求 t_{low}(定义 5 中设定的值),计算负载和响应时间的混合值 TMP_{value},这个值越大,表示当前的调整策略总体效果越好,因为响应时间在可接受的范围内,有最大值的限制,因此 TMP_{value} 越大,表示负载满足要求的概率越大,调整策略的总体效果越好,表达式中 t' 和 lc' 响应时间和负载的影响因子,根据租户需求而定;26 行～29 行,如果当前策略的 TMP_{value} 比历史最优策略的 MAX_{value} 大,则更新 MAX_{value} 和 MAX(MAX,标记最优策略的序号);29 行～34 行,如果当前策略与历史最优策略的混合值相同,则根据属性的历史操作日志信息,分别计算两种策略中,迁移属性与目标数据块中属性的关联程度(本文特指属性的访问率),如果当前策略总的关联度大于历史最优策略,更新 MAX_{value} 和 MAX. 第 41 行,将最优的调整策略赋值给 $PPS_{optimal}$. 第 42 行,返回最优策略. 算法需要遍历所有可行的调整策略和每个策略中的迁移路线,所以时间复杂度为 $O(n^2)$.

4.3 性能分析

(1)数据重构准确性

数据分块时,我们为每一个记录添加了全局唯一的标识 W_{id},并且根据 W_{id} 和分块序号 C_{num},为记录的分片添加分片标识 sid,sid 的计算公式如下所示[6]:

$$sid = E_{key}(W_{id} \oplus a^{C_{num}}) \tag{8}$$

其中,E 为乘法同态加密函数,key 为函数 E 对应的密钥,a 为隐私策略中指定的某个循环群的生成元. 根据式(8),可以为 W_{id} 的多个数据分片生成 sid. 数据重构时,首先查询出某个数据块的某个 sid,根据式(8),对 sid 解密可以获得 W_{id},再根据 W_{id} 获取其他的 sid,这样便可重构并得到完整的记录信息. 对数据块进行调整时,虽然记录的分片数目,分块序号和 sid 会发生变化,但是这些变化随着数据分块的调整,被更新到了隐私保护策略中,所以当调整后再次重构时,由式(8)可知并不会影响重构的准确性.

(2)隐私保护强度

租户隐私需求的变更,可能会导致隐私的泄露,例如属性 A 和 B 同在数据块 $Chunk_1$ 中,在原来的隐私需求中,并没有约束 A、B 不能属于一个数据块,但是租户更新后的隐私需求中要求 A、B 不能同属于一个数据块,如果不更新隐私保护策略,则会泄露隐私. 在更新隐私保护策略中,如果 $Chunk_1$ 和 $Chunk_2$ 原本不存在对等关系,由于 A、B 存在对等关系,但同在 $Chunk_1$ 时,不会泄露隐私,现在一个属于

$Chunk_1$，一个属于 $Chunk_2$，这样就泄露了 $Chunk_1$ 和 $Chunk_2$ 之间的关系. 假设调整前有 k 个数据块，m_1 对一一对应的属性对，调整后有 k_1 个数据块，n_1 对属性被分开($n_1 \leq m_1$). 如果不考虑不对等均衡问题，则隐私泄露的情况如式(9)所示. $P(c)$ 为存在隐私泄露的数据分块占所有数据块的比率. 由式(9)可知，在不考虑不对等均衡的情况下，如果 $k_1 < n_1$，则 $P(c)$ 的最大值可以达到 100%，如果 $k_1 \geq n_1$，$P(c)$ 的最大值也达到了 $2n_1/k_1$. 由此可见，数据块调整中不对等均衡的重要性. 本文通过添加伪造数据的方式[6]，保证调整后数据块间不存在一一对应关系，所以泄露隐私的数据块数为 0，即 $P(c)$ 的值为 0，从而增强了隐私保护强度.

$$\begin{cases} \dfrac{2}{k_1} \leq P(c) \leq 1, & k_1 < n_1 \\ \dfrac{2}{k_1} \leq P(c) \leq \dfrac{2n_1}{k_1}, & k_1 \geq n_1 \end{cases} \tag{9}$$

5　实验与结果分析

5.1　实验环境

为了验证本文提出的面向隐私保护的数据块调整机制的有效性和性能，我们搭建了 5 台服务器作为数据节点，配置为 8 核 CPU Inter(R) Xeon(R) 2.40 GHz，10 GB 内存，500 GB 硬盘. 系统采用 Red Hat Enterprise Linux 6.5 版本，Apache Tomcat 作为应用服务器，测试数据库采用 5.5.25 MySQL Community Server (GPL)，编程环境选用 Eclipse-SDK-4.3-win64，编程语言为 Java 1.7.

5.2　结果分析

在本文的实验中，我们设计了五组实验，分别对本文提出的调整机制(PPAM)、模式调整图算法(SAG)[8]、优化调整策略(OAS)[20]、可调整的数据放置方法(ADP)[21]和随机调整算法(RAA)进了模拟实验，结果如下所示.

(1) 性能测试

为了验证 PPAM 的效率以及它对系统性能的影响，本文设计了两个实验. 实验 1，将 5 种算法在同等条件下进行效率对比，即在同样的环境下，调整时间的对比. 测试数据中，数据属性共 80 个，原隐私约束 10 个，数据记录数 50 万条. 模拟租户设定了 5 种需求调整，调整后的隐私约束个数为 10，13，18，22，28，调整时间已经过处理，实验结果如图 4 所示.

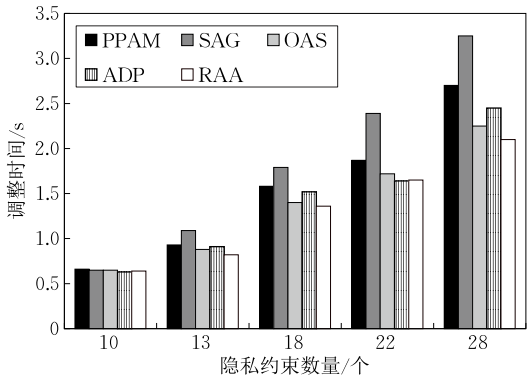

图 4　算法效率

由图 4 可知，随着隐私约束数量增加，5 种算法的调整时间也在增加. 这是因为，隐私约束个数越多，需要调整的属性和数据块数量越多，最后移动的数据量也越多，所以总的调整时间有所增加. 从图中可以看出，隐私约束个数为 10 时，5 种算法耗时彼此相近，但是随着隐私约束数量的变化，算法间的调整时间差异明显. PPAM 的耗时明显高于 RAA，OAS 和 ADP，但低于 SAG. 原因如下：在数据块调整的过程中，PPAM 不仅考虑数据调整后的隐私泄漏情况，还要保障满足性能需求和负载需求，因此它的效率比 RAA 略低；而 SAG 需要人工判断出每一个隐私约束调整的起始状态和目标状态，而且每次只能调整一个隐私约束，因此总体耗时比 PPAM 要多；OAS 主要考虑节点负载均衡，ADP 则注重数据访问性能，RAA 只考虑当前的隐私情况，因此它们的调整时间都比 PPAM 少.

实验 2，对调整后的系统性能进行测试，实验中，数据属性数量与实验 1 相同，隐私约束数量为 18 个，类型固定，数据量从 50 万～250 万单调递增，实验结果如图 5 所示，实验中设定租户最低性能要求是 2 s. 从图中可知，RAA 下，系统的响应时间总体比 PPAM 高且不稳定，存在比租户最低要求时间还高的响应时间，这是因为 RAA 算法具有随机性，因此它的响应时间忽快忽慢. PPAM 选择的是最优调整策略，所以 PPAM 的时间略低一些，总体查询时间都在租户要求的范围内，并随着数据量的增加，稳定增长. SAG 的耗时介于二者之间，因为它选择出的调整策略不是全局最优的，虽然隐私效果达到要求，但其他需求未必符合，所以效率比 PPAM 低，但又不是完全随机，因此效果好于 RAA. ADP 和 OAS 的查询时间都好于 PPAM，这是因为它们在调整时，不需要考虑隐私保护问题，所以为了提升数据访问效

率,它们可以将同类型或者同组的数据块最大化地放在一起,从而提升效率,因此查询时间比 PPAM 少.

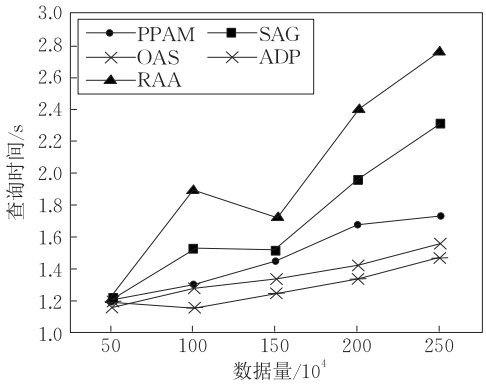

图 5 算法对系统性能的影响

(2)负载能力

为了验证 PPAM 对于系统负载的影响,本文设计了实验 3:隐私约束 18 个,类型随机变化,数据 50 万条,进行 100 次调整,并记录调整后 5 种算法下系统的负载能力情况,结果如图 6 所示.由图 6 可知,RAA 下,调整后,系统负载正常率在 50%上下波动.PPAM 下,调整后,系统负载正常率在 93%上下波动.SAG 下,调整后系统负载正常率在 85%上下波动.OAS 调整的目的就是实现节点负载均衡,因此该算法可以保证负载正常率 100%.ADP 虽然没有重点考虑负载,但为了实现优化数据访问性能,数据块放置也遵循规律,负载正常率在 80%到 90%之间浮动.由此可见,本文提出的 PPAM,负载正常率高,波动小,对于系统负载影响较小;而 SAG 和 ADP,虽然系统正常负载率与 PPAM 相近,但浮动较大,对系统性能的影响较大;RAA 下,系统负载正常率较低,难以满足系统的性能需求.

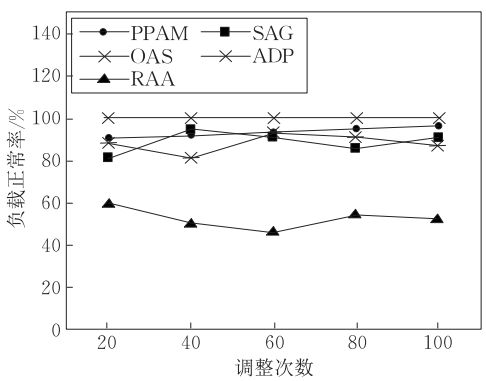

图 6 3 种算法下系统负载正常率

(3)准确性

为了验证最优策略准确性,本文对 PPAM 和 SAG 算法设计了实验 4:隐私约束 18 个,类型随机变化,数据 50 万条,进行 100 次更新调整,记录下每次调整过程中,使用 PPAM 生成的多种可行调整方案和最后选择出的最优调整策略;然后再根据每次调整生成的多种调整策略,进行实际的调整测试,并记录下调整后的系统响应时间和负载能力测试结果;最终,从 1 次~100 次调整过程中产生的可行调整策略里选择出最好的调整策略,并与 PPAM 和 SAG 直接选择出最优策略作对比:如果两个策略一样,则表示匹配成功,否则匹配失败.100 次匹配的结果如图 7 所示.

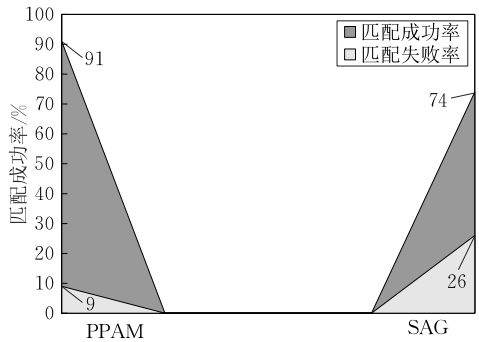

图 7 最优调整策略的准确性

从图 7 可知,实际测试选出的最优调整策略与 PPAM 生成的最优调整策略匹配成功率为 91%,与 SAG 生成的最优调整策略匹配成功率为 71%.由此可见,PPAM 直接生成的最优隐私保护调整策略准确性比 SAG 更高,效果更好.PPAM 选择的最优策略有 9%的误差,这是因为算法中使用的机器负载只是一个预测值,与真实值存在一定的差距,因此寻找的最优策略在一定范围内存在误差.对于 SAG 算法,最优的策略是用人工分析、选择出来的,所以误差较大,高达 24%.由此可见,PPAM 与 SAG 相比,具有更好的效果.

(4)隐私性

本文设计了实验 5,对于 5 种算法调整后的数据关系泄露情况进行了检测,本文研究的数据隐私泄露特指定义 3 提到的对等均衡引起的数据关系泄露.实验过程如下:实验环境与实验 4 相同,首先,对原有隐私约束进行 100 次调整;然后分别使用 5 种算法对原有隐私策略各进行 100 次调整,并检测调整后存在对等均衡的次数;最后,按照分组统计每个算法的隐私保护效果(调整总次数中,不存在隐私泄露次数所占的比率),统计结果如图 8 所示.

通过图 8 可知,PPAM 下,隐私保护效果 100%,不存在对等均衡.这是因为本文在调整过程中,着重考虑了对等均衡问题.RAA 下,隐私保护效果随着

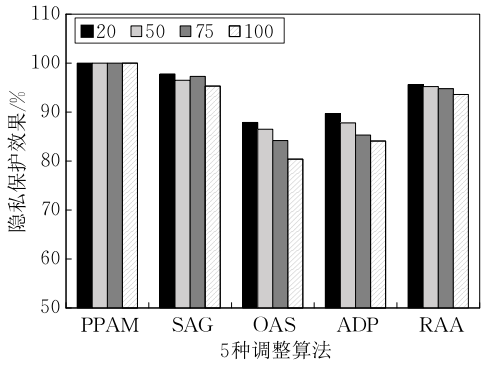

图 8　3 种算法的隐私保护效果

调整次数的增多而变差，衰减程度也逐渐趋于平稳，而且隐私保护效果维持在 93%以上. 这是因为调整次数越多，数据块彼此间交换属性的可能性就越大，所以对等均衡出现的概率也就越大；而衰减程度随着调整次数的增多，新数据块组合出现的概率也就越小，进而衰减度也就变小. 对于原有的数据，本文采用前期工作[6]中提出的分块混淆隐私保护机制，所以隐私保护效果较好，即使在数据调整时，使用了 RAA 算法，隐私泄露的概率也很低. SAG 的隐私效果比 RAA 略高，但低于 PPAM，这是因为 SAG 在调整的同时，也考虑数据分布均衡问题，所以它的效果比 RAA 好. 然而，它只考虑了块内的数据分布均衡，并没考虑到块间的对等均衡，所以隐私保护效果比 PPAM 差一些. 而 OAS 和 ADP 都没有考虑到隐私保护的问题，并且为了实现负载均衡和提升数据访问性能，把一些关系相近的数据重组到一起，增加了隐私泄露的可能性，所以隐私保护效果比 RAA 差. 由上可知，PPAM 在满足负载和性能的需求下，可以更好地保护数据隐私.

(5) 选择最优策略的影响因素

为了提高最优策略的准确性，我们研究了属性数量、隐私约束数量和数据量与最优策略准确性的关系，并设计了实验 6，结果如图 9 和图 10 所示.

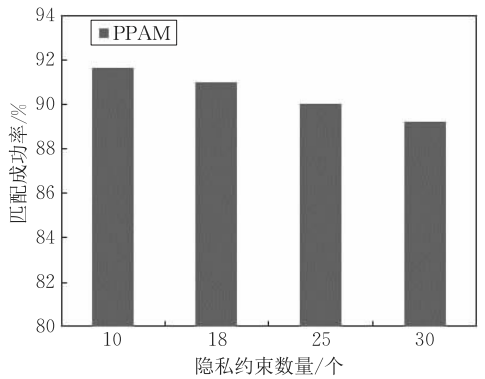

图 9　隐私约束数量对最优策略选择的影响

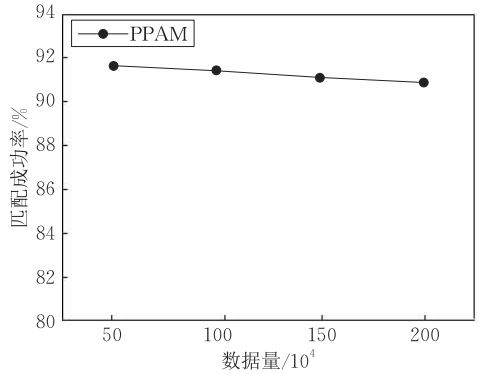

图 10　数据量对最优策略选择的影响

从图 9 可以看出，随着隐私约束数量的增加，匹配的成功率有所下降，这是因为在其他条件固定下，隐私约束数量越多，其种类也就越丰富，数据分块和重组的方案也就越复杂，数据变化的浮动也就比较大，所有系统负载预测的准确性有所下降，进而影响了最优策略的选择.

图 10 中，可以看出随着数据量的增加，匹配成功率变化不大，这是因为在隐私约束数量和类型不变的情况下，数据块分割和重组的方案基本不变，所以数据变化有规律可言，这有利于对系统负载的预测，保证了负载预测的平稳性，进而可以维持最优策略的匹配成功率.

6　总　结

租户需求的变动导致原有的隐私保护策略不再满足租户的隐私需求，且存在隐私泄露的风险. 因此本文提出一种面向隐私保护的数据块调整机制. 该机制首先根据租户提出的调整需求，对原隐私保护策略中的数据块进行拆分，将违背新隐私约束的数据块基于少动性原则，分割成多个子数据块；然后再根据新隐私约束，重新把不违背隐私约束的数据块组合起来，再从混合有原有不变的数据块和重新组合后的数据块集合中，挑选出包含所有数据属性的数据块的组合，作为隐私保护调整策略；最后通过基于全局最优的调整策略选择算法，全面考虑调整策略对系统性能、负载能力和数据隐私等的影响，筛选出最优的调整策略. 本文最后通过实验，从算法效率、对系统性能和负载的影响、生成调整策略的准确定和调整后数据的隐私性等方面验证了本文提出的隐私保护调整机制的有效性和可行性.

下一步，将重点研究数据调整对数据重构的影响，并根据调整策略，建立最快的数据重构方案.

参 考 文 献

[1] Zhang Hao, Huang Tao, Liu Sannv-Ya, Wang Li-Na. A privacy-preserving bucket partition mechanism in cloud. Chinese Journal of Computers, 2016, 39(2): 429-440 (in Chinese)
(张浩，黄涛，刘三女牙，王丽娜. 云平台下基于隐私保护的桶划分方案. 计算机学报，2016，39(2)：429-440)

[2] Kauthale K, Sunil-D-Rathod. Vertically partitioning of database for secured data release. International Journal of Computer Applications, 2015, 117(21): 1-5

[3] Mohammed N, Alhadidi D, Benjamin C-M-Fung, Debbabi M. Secure two-party differentially private data release for vertically partitioned data. IEEE Transactions on Dependable and Secure Computing, 2014, 11(1): 59-71

[4] Wang Li-Xuan, Liu Li-Fang, Liu Shen-Ling, et al. A secured distributed and data fragmentation model for cloud storage//Proceedings of the 2013 International Conference on Precision Mechanical Instruments and Measurement Technology. Shenyang, China, 2013: 2693-2699

[5] Li Lin, Li Qing-Zhong, Shi Yu-Liang, Zhang Kun. SAPS—A single attribute protection scheme for SaaS. Information, 2012, 15(1): 275-282

[6] Zhang Kun, Li Qing-Zhong, Shi Yu-Liang. Research on data combination privacy preservation mechanism for SaaS. Chinese Journal of Computers, 2010, 33(11): 2044-2054(in Chinese)
(张坤，李庆忠，史玉良. 面向 SaaS 应用的数据组合隐私保护机制研究. 计算机学报，2010，33(11)：2044-2054)

[7] Shao Ya-Li, Shi Yu-Liang. A novel cloud data fragmentation cluster-based privacy preserving mechanism. IJGDC: International Journal of Grid and Distributed Computing, 2014, 7(4): 21-32

[8] Zhang Kun, Abraham A, Shi Yu-Liang. Data combination privacy preservation adjusting mechanism for software as a service//Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics. Manchester, UK, 2013: 2007-2012

[9] Saroj S-K, Chauhan S-K, Sharma A-K, Vats S. Threshold cryptography based data security in cloud computing//Proceedings of the 2015 IEEE International Conferenceon Computational Intelligence & Communication Technology. Ghaziabad, India, 2015: 202-207

[10] Xiong Jin-Bo, Li Feng-Hua, Ma Jian-Feng, Liu Xi-Meng. A full lifecycle privacy protection scheme for sensitive data in cloud computing. Peer-to-Peer Networking and Applications, 2014, 8(6): 1-13

[11] Craig G. Fully homomorphic encryption using ideal lattices//Proceedings of the 41st ACM Symposium on Theory of Computing. Bethesda, USA, 2009: 169-178

[12] Wassim I, Ayman K, Ali C. Privacy as a service: Privacy-aware data storage and processing in cloud computing architectures//Proceedings of the 8th IEEE International Conference on Dependable, Autonomic and Secure Computing. Chendu, China, 2009: 711-716

[13] Yang Ji-Jiang, Li Jian-Qiang, Niu Yu. A hybrid solution for privacy preserving medical data sharing in the cloud environment. Future Generation Computer Systems, 2015, 43-44: 74-86

[14] Li Jin-Guo, Tian Xiu-Xia, Zhou Ao-Ying. Privacy preserving fuzzy keyword search in database as a service paradigm. Chinese Journal of Computers, 2016, 39(2): 414-428(in Chinese)
(李晋国，田秀霞，周傲英. 面向 DaaS 保护隐私的模糊关键字查询. 计算机学报，2016，39(2)：414-428)

[15] Feng Deng-Guo, Zhang Min, Li Hao. Big data security and privacy protection. Chinese Journal of Computers, 2014, 37(1): 246-258(in Chinese)
(冯登国，张敏，李昊. 大数据安全与隐私保护. 计算机学报，2014，37(1)：246-258)

[16] Ouyang Jia, Yin Jian, Liu Shao-Peng, et al. An effective differential privacy transaction data publication strategy. Journal of Computer Research and Development, 2014, 51(10): 2195-2205(in Chinese)
(欧阳佳，印鉴，刘少鹏等. 一种有效的差分隐私事务数据发布策略. 计算机研究与发展，2014，51(10)：2195-2205)

[17] Lin Chi, Wang Peng-Yu, et al. A differential privacy protection scheme for sensitive big data in body sensor networks. Annales des Telecommunications. Annals of Telecommunications, 2016: DOI: 1-11:10.1007/s12243-016-0498-7

[18] Kapusta K, Memmi G. Data protection by means of fragmentation in distributed storage systems//Proceedings of the International Conference on Protocol Engineering, ICPE 2015 and International Conference on New Technologies of Distributed Systems. Paris, France, 2015, DOI: 10.1109/NOTERE.2015.7293486

[19] Valentina C, De Capitani D V S, Sara F, et al. Combining fragmentation and encryption to protect privacy in data storage. ACM Transactions on Information and System Security, 2010, 13(3): DOI:10.1145/1805974.1805978

[20] Li Xiao-Na, Li Qing-Zhong, Zhu Wei-Yi, Li Hui. Research on optimization adjustment strategy for SaaS multi-tenant data placement. International Journal of Grid and Distributed Computing, 2015, 8(2): 319-330

[21] Sun Qian, Jin Tong, et al. Adaptive data placement for staging-based coupled scientific workflows//Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. Austin, USA, 2015: 1-12

SHI Yu-Liang, born in 1978, Ph.D., professor. His research interests include cloud computing, database and privacy preserving.

CHEN Yu, born in 1990, M.S. candidate. His research interests include cloud computing and privacy preserving.

SUN Shi-Bin, born in 1989, M.S. candidate. His current research interests include cloud computing and privacy preserving.

CUI Li-Zhen, born in 1976, Ph.D., professor. His research interests include data science and engineering, service computing and intelligent collaborative computing.

Background

The topic this paper researched belongs to the problem of privacy protection in the cloud computing environment. The popular international solution to this kind of problem is data encryption. Besides, there are also some other privacy protection methods, such as data confusion technology, anonymous privacy protection technology and differential privacy protection. However, all the methods above have only consider the privacy protection, and do not take into account the issue of privacy protection adjustment. Therefore, in our earlier works, a privacy preserving technology based on chunk-confusion is proposed. First, according to the privacy constraints (PC) customized by tenants, the data of tenants are partitioned into a lot data chunks to ensure that the attributes in PC are in different data chunks. Next, we confused the relationships among attributes and saved them on the trusted third party. Finally, data chunks are assigned to multiple data nodes according to their load capacity and computing capability. However, due to the privacy needs of tenants and the data demands of tenants are variable, the underlying data chunks structure and storage location in the cloud will change, which makes there is a risk of leakage of privacy. Hence, based on our earlier work, the paper proposed a mechanism of data chunks adjustment based on privacy protection. The mechanism can not only find an optimal strategy for privacy protection requirements and increase the privacy protection effect of the tenants' data, but also meet the system's load and performance. The previous studies of our team in the direction of privacy preserving under the cloud computing environment include: A New Privacy-Preserving Scheme DOSPA for SaaS, Data Combination Privacy Preservation Mechanism for SaaS, Policy-Based Customized Privacy Preserving Mechanism for SaaS Applications, A Novel Cloud Data Fragmentation Cluster-based Privacy Preserving Mechanism, Document-oriented Database-based Privacy Data Protection Architecture, Game-Theoretic Strategy for Personalized Privacy Protection, A Sub Chunk-confusion Based Privacy Protection Mechanism For Association Rules, Model for Hiding Data Relationships Based on Chunk-Confusion in Cloud Computing, etc.

As a part of our project, the research work in this paper is supported by the TaiShan Industrial Experts Programme of Shandong Province No. tscy20150305, and the Key Research & Development Plan of Shandong Province (Nos. 2016GGX101008, 2016ZDJS01A09), the Natural Science Foundation of Shandong Province for Major Basic Research Projects (No. ZR2017ZB0419).