基于 L-DHT 的多租户虚拟域隔离构建方法

曹利峰，卢新，高振升，杜学绘
（信息工程大学密码工程学院，河南 郑州 450001）

摘 要：针对云环境下多租户数据的安全隔离的问题，提出了一种基于 L-DHT 的多租户虚拟域隔离构建方法。首先，通过设计一种基于标签 Hash 映射的多租户隔离映射算法，构建了租户资源的均衡映射机制，实现对租户资源的分布式管理；然后，针对映射到同一存储节点上租户数据间的安全隔离与访问，基于谓词加密机制，通过安全标签和租户数据的有效绑定，给出了一种基于标签谓词加密的租户数据隔离存储算法；最后，通过设计多维度的租户数据隔离控制规则，利用对安全标签的解析与认证，层次化地构建起租户间相互独立、逻辑、安全的虚拟域。安全性分析表明，所提方法构建了相互间安全无干扰的租户虚拟域。仿真实验结果表明，映射算法能够更好地实现负载的动态平衡，并通过数据检索效率与访问安全性的对比分析，验证了租户访问数据的安全性与高效性。

关键词：租户虚拟域；域隔离器；安全标签；多租户映射；数据隔离
中图分类号：TP393
文献标识码：A
doi: 10.11959/j.issn.1000−436x.2020088

Multi-tenant virtual domain isolation construction method based on L-DHT

CAO Lifeng, LU Xin, GAO Zhensheng, DU Xuehui
College of Cryptogram Engineering, Information Engineering University, Zhengzhou 450001, China

Abstract: Aiming at the problem of security isolation of multi-tenant data in cloud environment, a tenant virtual domain isolation construction method based on L-DHT was proposed. Firstly, through the design of multi-tenant isolation mapping algorithm based on label-hash mapping, the balanced mapping mechanism of tenant resources was constructed to realize the distributed management of tenant resources. Secondly, for the security isolation and access between tenant data mapped to the same storage node, based on the predicate encryption mechanism, through the effective binding of security labels and tenant data, a tenant data isolation storage algorithm based on label predicate encryption was designed. Finally, by the design of multi-dimensional tenant data isolation control rules and using the analysis and authentication of security labels, independent, logical and secure virtual domains between tenants were built hierarchically. The security analysis shows that the method constructs tenant virtual domains which are secure and non-interference with each other. The simulation results show that the mapping algorithm can achieve a better dynamic load balance. The efficiency and security of data access are verified by the comparative analysis of tenant data retrieval efficiency and authentication access security.

Key words: tenant virtual domain, domain isolator, security label, multi-tenant mapping, data isolation

---

收稿日期：2019−11−28；**修回日期**：2020−04−07
通信作者：卢新，1209774364@qq.com
基金项目：国家自然科学基金资助项目（No.61502531, No.61702550）；国家重点研发计划基金资助项目（No.2018YFB0803603, No.2016YFB0501901）
Foundation Items: The National Natural Science Foundation of China (No.61502531, No.61702550), The National Key Research and Development Program of China (No.2018YFB0803603, No.2016YFB0501901)

1　引言

云计算是一种面向多租户实现资源与应用复用的服务平台，由于具有虚拟化、开放性、共享性等特性，因此存在着租户难管理、难隔离，数据易泄露等安全性问题[1-2]。云安全联盟也将不安全的网络和数据隔离列为云安全的首要威胁之一[3]。

租户的安全隔离是指云环境下多个租户在共享资源的同时，保证租户的私有数据不受其他租户的影响，防止租户间的信息流动和业务的干扰[4]。针对隔离安全需求，多种隔离方法应运而生，例如同态加密存储、虚拟机隔离、访问控制等[5-8]。虽然上述这些隔离方法在一定程度上实现了租户数据间的隔离，但大多只侧重于加密隔离存储、租户物理隔离、数据访问控制等某一层面的隔离，没有充分考虑多个层次安全隔离之间的关系。

针对现阶段租户数据隔离研究中存在的问题，本文从数据隔离多个层面下的安全需求出发，对租户数据的安全隔离展开了详细的研究。本文细粒度地将租户虚拟域的构建划分为隔离存储层、虚拟网络隔离层及管理隔离层，利用从整体到局部的思路解决不同层次下租户数据隔离的安全需求。首先，通过引入虚拟域安全标签，改进一致性 Hash 算法，提出了一种负载均衡的多租户隔离映射算法，完成租户资源分布式均衡映射；然后，通过安全标签与存储数据的有效绑定，提出了一种租户数据隔离存储算法，解决租户共享存储时隔离与访问的安全问题；最后，在租户数据隔离控制规则的约束下，利用对安全标签的解析与认证，实现了对虚拟域内租户数据传输的安全控制。

2　相关研究

“多租户架构”最早出现于软件应用领域，是云计算中最基本的特性之一[9]。“多租户架构”要求多个租户在共享应用、资源和服务的同时保证租户间的相互隔离[10]。因此，租户数据的安全隔离是多租户架构设计的关键。近年来，多种租户隔离技术相继出现。从租户隔离的不同层面来看，租户隔离技术可分为基于硬件层面的隔离、操作系统层面的隔离、中间件层面的隔离、租户网络层面的隔离和存储层面的隔离。不同层次下云租户数据的安全隔离研究如下。

1) 基于硬件层面的隔离是指通过扩展硬件或提高性能的方式来保证租户间的安全隔离，主要包括物理空间的隔离、事务处理时间的隔离、硬件控制租户数据流等方式[11-12]。基于硬件层面的隔离实现技术思路简单，隔离程度高，但设备性能要求高，消耗大，成本高。

2) 操作系统层面的隔离是指借助虚拟机来实现租户间计算资源的共享，同时保证租户间的隔离，安全隔离性良好，主要包括虚拟机管理层和虚拟机层两方面的隔离方案[13]。Roy 等[14]提出了一种基于虚拟机的信息流控制方式，不仅能跟踪程序变量级的租户信息流，还能保护操作系统中租户的资源。杨永娇等[15]提出了一个应用于 Xen 云平台的安全隔离架构，加密保护了虚拟机中的数据，提高了虚拟机间的 I/O 和访问内存的隔离安全性。Malka 等[16]基于硬件辅助虚拟化技术，针对虚拟机隔离安全提出了环形 rLOMMU 模型，提升了隔离性能，但没有提出系统的虚拟机隔离方案。单纯的虚拟机隔离容易出现多租户混乱的问题[17]，同时若虚拟机隔离控制被攻破，将会威胁到其他虚拟机安全[18]。

3) 中间件层面的隔离是指租户间共享操作系统资源，典型的当属容器技术。容器技术[19]是轻量级的虚拟化技术，由于自身并不实现操作系统级的服务，因此占用资源少，共享程度高，具有代表性的容器技术包括 LXC（Linux container）和 Docker。但由于容器是为提供高效轻量服务所设计的，因此对其的隔离仅停留在软隔离层面，存在共享内核代码的安全威胁，并且容器中的错误容易扩散到主机中，造成宕机并威胁到其他容器的安全，进而威胁到租户数据的安全[20]。

4) 租户网络层面的隔离是指租户间网络的逻辑隔离，以网络虚拟化为基础，保证租户间的数据传输的安全[21-22]。传统数据中心网络中一般采用虚拟局域网（VLAN, virtual local area network）技术，租户独享 VLAN，VLAN 间相互隔离无法通信，由于 VLAN ID 的限制和租户规模的逐渐扩大，出现了 VLAN 数量不足和配置复杂的问题。为适应大规模数据中心租户网络的隔离需求，许多新的技术应运而生。Overlay 网络架构是解决多租户网络隔离的最佳方式之一，其实现的典型技术方案为可扩展虚拟局域网（VXLAN, virtual extensible LAN）和 NVGRE（network virtualization using generic routing encapsulation）[23-24]。VXLAN 技术增加了 VLAN 的数量，实现了跨域的二层互连，但虚拟隧道终端节点的负担过大，影响了网络整体的性能；NVGRE 协议主要通过采用路由封装协议来进行封装，其最

大子网划分数量充足且广播方式更加灵活，但没有采用标准传输协议，兼容性差，设备要求高。

此外，Amamou 等[25]设计了一种多链路透明互联（TRILL, transparent interconnection of lots of links）的多租户网络架构，主要利用 TRILL 协议完成路由和桥接，提高了租户网络的扩展性，但也提高了安全性和故障隔离的难度。严立宇等[26]提出了一种多租户虚拟网络隔离的分布式实现方法，该方法分摊了单一节点的网络流量，降低了单点故障损失的可能性，但其属于静态的部署方式，没有考虑到租户的动态迁移。孙延涛等[27]提出了一种基于分布式哈希表（DHT, distributed Hash table）的数据中心网络租户隔离技术，将租户隔离分布式地部署在多个位置解析器上，解决了租户间的安全隔离和虚拟机迁移的问题，但没有考虑系统整体负载情况及租户数据流动的安全控制。上述方案的研究主要通过为租户划分私有网络，单纯利用租户网络标识来实现隔离，但未涉及具体的数据流动规则和租户网络管理的负载均衡方法。

5) 存储层面的隔离是指租户在数据共享存储资源的同时，保证租户数据的隔离存储与访问。针对租户数据的隔离存储，常见的有独立数据库、独立数据架构和共享数据架构 3 种，这 3 种措施的共享程度依次增高但隔离程度逐渐降低，无法有效地兼顾共享程度与隔离效果[28]。全同态加密算法[29]因其可以计算加密数据的特性，在云计算隔离存储和计算安全层面的必要性日益凸显。光焱等[30]提出了一种基于身份的全同态加密体制，但其在密文运算中需要引入运算密钥，并没有基于身份信息进行运算。段然等[31]提出一种 NTRU（number theory research unit）格上高效的基于身份的全同态加密体制，减小了密钥和密文的尺寸，提高了加解密的效率，但是缺乏对用户属性信息的认证。此外，杜瑞忠等[32]提出了一种基于封闭环境加密的云存储方案，保护了云存储中用户数据的机密性，但无法支持密文检索。Iiya 等[33]提出了一种多租户原型系统，系统基于多权限属性来保证租户数据的隔离存储，但在认证过程中要频繁地颁发用户证书来保证安全性，增大了系统的开销。上述多租户数据存储方案主要集中在数据加密存储和提高存储效率 2 个方面，缺少隔离存储与认证访问有效结合的安全方案设计，并且对数据检索信息的安全性考虑不足。

3 租户虚拟域隔离构建思想

3.1 相关术语

术语 1 虚拟域（VD, virtual domain）。同一租户所处的逻辑安全域，包括租户资源及用户在内的成员集合。

术语 2 域隔离器（DR, domain isolator）。管理租户资源信息的核心路由器/交换机。

每个域隔离器内置域解析网关，包括标签生成、标签映射索引、地址解析、标签认证与解析、资源管理与调度等功能组件，分别完成租户标签的生成与分配，标签与租户信息的映射索引，对域内数据访问位置的重定向，标签的解析、认证及处理，租户资源的统一管理与调度等功能。域隔离器功能结构如图 1 所示。

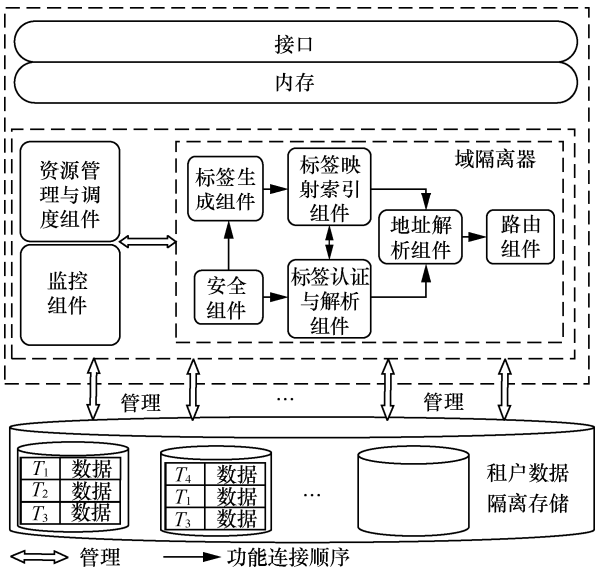


图 1 域隔离器功能结构

术语 3 安全标签（SLebal, security label）。租户虚拟域及租户虚拟域内资源、数据的唯一标识，面对不同层次的隔离需求，本文设计了不同形式的安全标签 SLebal={TID, DLebal, VLebal}。其中，TID 为虚拟域安全标签，DLebal 为数据存储标签，VLebal 为数据控制标签。

1) TID。租户虚拟域的区分标识，TID=Hash(ID||T)，其中，“||”为连接操作；ID 为租户唯一标识；T 为租户完成映射后与 DR 协商生成的秘密序列号，用于模糊租户身份信息，防止恶意租户的假冒及篡改等威胁。

2) DLebal。租户共享存储时，用于标识和隔离

存储租户私有数据，并实现对存储数据访问时的属性认证，DLebal={TID, KEY, φ, TK}。KEY 是用于加密存储数据的密钥。$\varphi=\{f_1,f_2,\cdots,f_n\}$是用户访问存储数据的策略谓词集合，属性 $A=\{A_1,A_2,\cdots,A_n\}$，例如策略谓词 φ= {“USER= Bob”, “read”,…, “Data∈[2017,2019]”} 。当且仅当 $\forall f_i(A_i)=f_i(A_i^{'}),\ i\in[1,n]$($A'$是访问属性)时，用户能够获取存储的数据，并利用 KEY 解密数据密文。TK 是查询令牌，用于检索 $f(A)$=$f(A')$密文。

3) VLebal。数据控制标签用于控制租户虚拟域内外数据的安全标签， VLebal={TID,Hash_S, BAN_TIME, Trans_Type}。S 为数据的安全级别，Hash_S 为安全级别 S 的 Hash 值，BAN_TIME 为保密期限，Trans_Type 为数据传输方式。

3.2　基本思想

本文基于不同层次下租户安全隔离的需求，将租户虚拟域的构建划分为隔离存储层、虚拟网络隔离层和管理隔离层 3 个层次，在此描述了多租户虚拟域层次化隔离构建的基本思想，多租户分布式安全隔离架构如图 2 所示。

1) 本文通过在云数据中心网络部署多台域隔离器，利用域安全标签标识虚拟域，设计了多租户虚拟域映射算法，将租户资源信息均衡映射到多个域隔离器上，实现对租户资源的分布式管理与隔离，如图 2 中的管理隔离层。

2) 针对租户虚拟网络层的隔离，设计数据控制标签与租户数据分组的绑定方法，利用域隔离器的标签解析认证、地址解析等功能，实现域隔离器对租户虚拟域内外数据传输的分布式隔离管理，如图 2 中的虚拟网络隔离层。

3) 针对租户数据的隔离存储，设计基于标签谓词加密的租户数据隔离存储算法，通过数据存储标签与租户数据的安全绑定，利用域隔离器解析标签，利用 TID 值区分存储节点中的租户数据；通过引入策略谓词 φ 及查询令牌 TK，数据加密密钥对应策略谓词 φ，数据密文对应于属性集 A，实现租

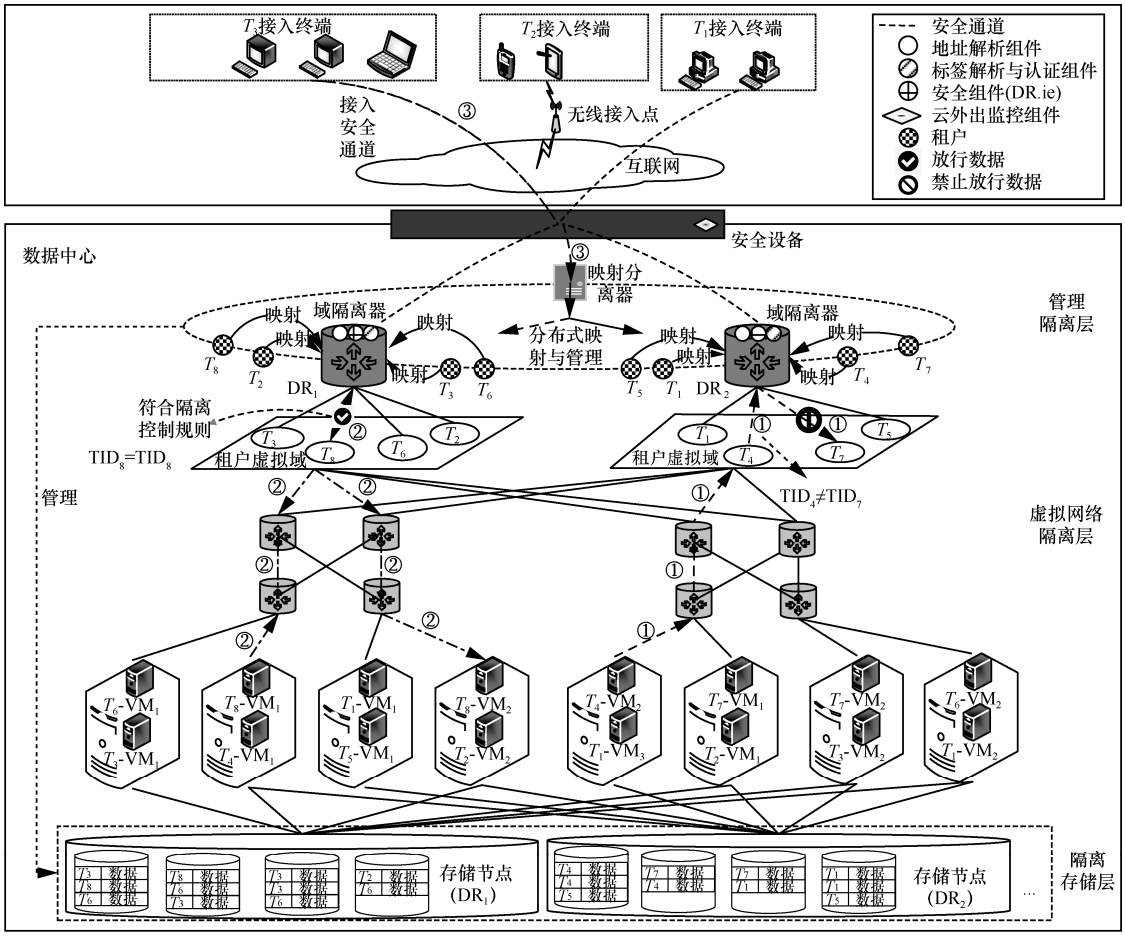


图 2　多租户分布式安全隔离架构

户数据的加密隔离存储；通过查询令牌与数据访问属性的匹配认证，实现对存储数据的快速查询与认证访问，如图 2 中的隔离存储层。

4)通过制定租户数据隔离控制规则，利用数据控制标签标记并跟踪数据，防止租户数据聚合推导导致泄密的同时，通过建立数据安全通道及数据流动控制规则，保证虚拟域租户数据传输与访问的安全隔离。图 2 中的①表示不同租户虚拟域间的通信隔离；②表示同一租户虚拟域内的在符合隔离控制规则时的相互通信；③表示租户映射或接入访问时安全通道的建立。

综上所述，本文通过引入不同层次下的安全标签，在域隔离器的分布式管理下，将租户虚拟网络的建立、数据的隔离存储与访问紧密结合，构建相互独立的租户虚拟域，来实现租户数据的安全隔离。

4 租户虚拟域隔离构建方法设计

4.1 基于标签 Hash 映射的多租户隔离映射算法

基于标签 Hash 映射的多租户隔离映射算法流程如图 3 所示，主要包含 6 个步骤，分别是初始化、虚拟分区的划分、虚拟节点的按权分配、“租户–虚拟节点–域隔离器”的双映射、虚拟节点的迁移、TID 的协商。算法具体流程如下。

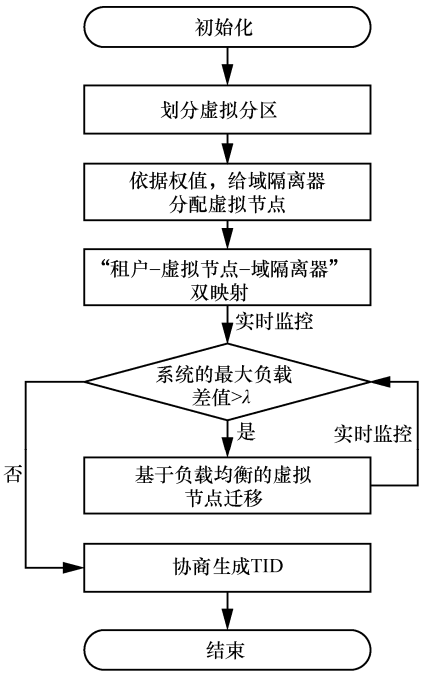


图 3 映射算法流程

步骤 1 初始化。读取域隔离器 DR=$\{DR_1, DR_2,\cdots,DR_n\}$ 的地址信息；读取租户 ID=$\{ID_1, ID_2,\cdots,ID_k\}$。

步骤 2 划分虚拟分区。设置虚拟节点（即一致性 Hash 环上虚拟出来的均匀分布的节点，一个域隔离器对应多个虚拟节点，旨在实现租户到域隔离器更均匀的映射）。

① 设 DR 个数为 n，对应 Hash 环为 Hloop_a，设置虚拟倍数 α，VR 个数 c 为 2 的幂次方倍，即 $c=2^m$，$1\leqslant\frac{\alpha n}{2}\leqslant 2$。

② 分区 Hash 环，完成“点–区”映射。将环 Hloop_a 上的值空间分别等分为 c 个子区间 SHloop_a，Hash 环上的每个区间的结束值为 VR 的 Hash 值，区间上的所有点映射到该 VR，VR 的值之间呈等差数列排列，公差为 $d_1=2^{32-m}$。

步骤 3 依据权重，分配虚拟节点。为使租户的管理隔离分布式地均匀部署在每台 DR_i 上，在此根据域隔离器吞吐量和存储容量等性能标准，引入性能基准值 $\overline{S}$ 及权重 W 的概念，通过与性能基准值比较得到每台 DR_i 的权重 W_i，并根据 W 为每台 DR 分配 VR，权重计算式为 $W_i=\frac{\overline{W}S_i}{\overline{S}}$，将 VR 的个数按照权重 W_i 分配给 DR_i，即 $N_i=\left\lfloor\frac{W_i|\text{VR}|}{\sum_{i=1}^{|\text{DR}|}W_i}\right\rfloor$。采取向下取整方式，若存在少量未分配 VR（个数为 P），则按照 $\frac{N_i}{W_i}$ 的大小依次分配给负载低的 DR（负载相同时按节点编号顺序分配），分配单位为 1。

随机在虚拟节点集合中选取 N_i 个未分配的虚拟节点，“虚–实”节点的映射以字典映射的形式被记录，即<VR_{Hash}, DR_i.address>。

步骤 4 租户到域隔离器的映射（定位）。计算租户 ID 的 Hash 值，映射到相应虚拟节点。

①计算步骤 1 中租户 ID 列表的 Hash 值。

②“值–点”映射。依次将租户 ID 的 Hash 值映射到 Hash 环 Hloop_a 上，顺时针方向寻找最近的 VR，根据“虚–实”映射结果找到所属的 DR，完成租户到 DR 的映射。对应的虚拟节点范围为 $\text{Hash}(ID_k)\leqslant\text{VR}_{\text{Hash}}\leqslant(\text{Hash}(ID_k)+\text{Len})$，其中 Len 为区间长度。

图 4 描述了当 3 台域隔离器 DR_1、DR_2、DR_3 性能权重分别为 1:1:1，虚拟节点倍数为 3 时，租户向 VR 映射及 VR 按权重随机分配到 DR 的过程。

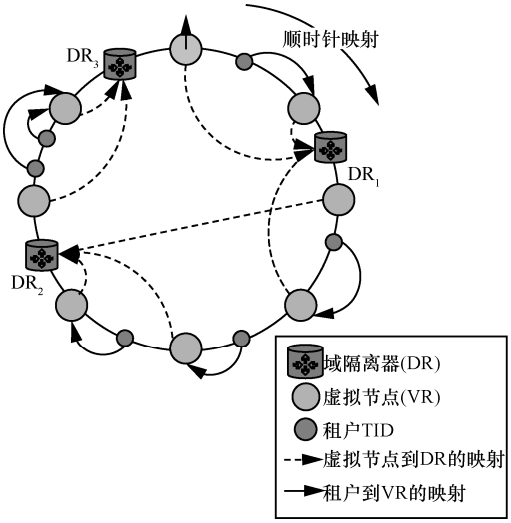


图 4　租户 Hash 映射

步骤 5　基于负载反馈的虚拟节点迁移。随着租户数量的变化，难免会出现域隔离器的数量和性能变化，导致租户分布出现倾斜，从而影响整体的隔离效果和系统性能。

虚拟节点的迁移情况共有以下 3 种：一是当全部 DR 的负载过高时需要扩充 DR 的数量；二是租户的集中映射会导致现有 DR 的负载不均衡；三是由于 DR 宕机，因此需要对释放的虚拟节点进行重分配。当隔离系统监控到租户分布出现倾斜时，根据 DR 的权重来实现虚拟节点的动态分配。针对不同情况，迁移过程如下。

针对 DR 的移除，可以利用步骤 3 中将被释放的租户按照剩余 DR 的权重重新分配；针对租户的分布不均衡和 DR 的扩容，引入负载 L（$L_i=\frac{T_i}{W_i}$,其中 T_i 为 DR_i 所承载的租户数量）和数据倾斜阈值 λ，即当系统中的最大负载差值比 $\frac{L_{\max}}{L_{\min}}-1\geqslant\lambda$ 时，系统通过调整权重并根据当前权重从高负载节点负责的虚拟节点中随机挑选并重分配给低负载节点来实现负载均衡。虚拟节点重分配后，重写“虚–实”映射表的 value 值。迁移过程分别如图 5 和图 6 所示。

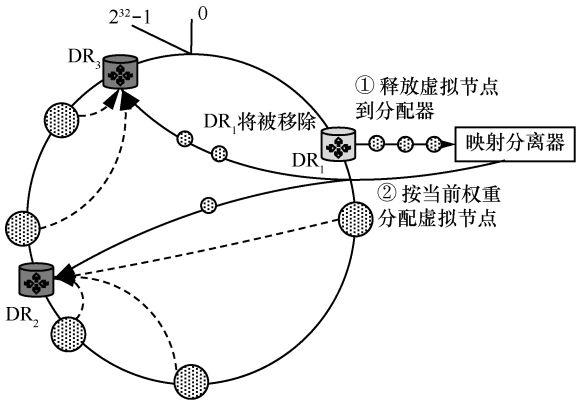


图 5　DR 移除时虚拟节点迁移

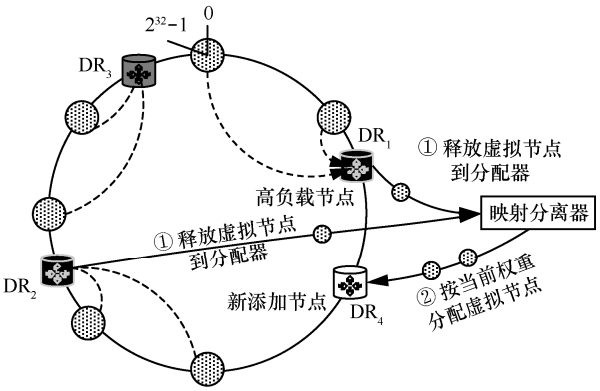


图 6　DR 增加时虚拟节点迁移

图 5 为移除 DR_1 后虚拟节点的重分配到 DR_2 和 DR_3 的过程；图 6 为高负载节点向低负载节点分配虚拟节点的过程，设其性能权重与其他 3 个 DR 相同。

步骤 6　租户完成映射或迁移后与 DR 协商生成虚拟域安全标签 TID=Hash(ID||T)，如图 7 所示。

4.2　基于标签谓词加密的租户数据隔离存储算法

针对映射到同一存储节点的租户数据，基于租户数据隔离存储的基本思想（3.2 节的 3)），租户数

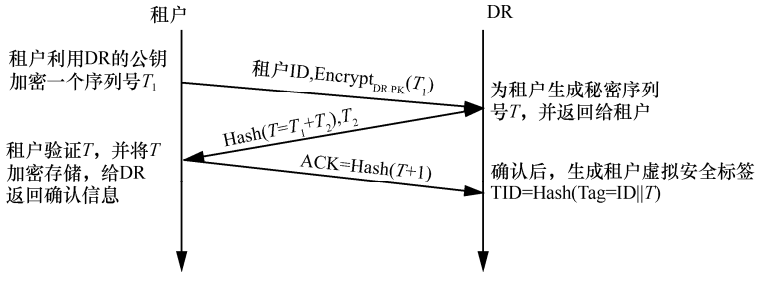


图 7 TID 的协商生成

据隔离存储算法设计如下。

假设租户 T 存储数据 Data 的存储标签为 DLebal={TID_T, KEY, φ, TK}，该算法可表示为一个五元组 ε={Setup(1^k,Data); GenToKen(φ, TID_T); Encrypt(KEY,Data); Query(TK, C, TID′, Data. A'); Decrypt (KEY, C)}。

1) Setup(1^k, Data)。输入安全参数 1^k 和数据属性 A={$A_1, A_2,\cdots, A_n$}，输出密钥 KEY 及策略谓词 φ={$f_1, f_2,\cdots, f_n$}。

2) GenToKen(φ, TID_T)。输入策略谓词 φ、TID_T，计算查询令牌 TK，过程如下。

for (i=1; $i\leqslant n$; i++)

{tk_i=Hash(f_i||TID_T);}

TK={$tk_1, tk_2,\cdots,tk_n$};

输出 TK。

3) Encrypt(KEY, Data)。输入密钥 KEY 及明文 Data，输出一个密文 C= $Encrypt_{key}$(Data)。

4) Query(TK, C, TID′, Data. A')。输入查询令牌 TK、密文 C、数据访问者的域安全标签 TID′及其访问的属性信息 A'={$A'_1,A'_2,\cdots,A'_n$}，进行租户查询属性的认证，过程如下。

j=1;

for (i=1; $i\leqslant n$; i++)

{if(Hash($f_i(A_i^{'})$||TID′)=tk_i)

{j=j+1;}

else{break;}}

if (j=n)

{Query (TK, C, A') =1; return C;}

else {Query (TK, C, A') =0; return 0;}

访问者属性信息经过 Hash 运算后生成令牌 TK，通过令牌 TK 可对存储的租户密文数据进行保密查询，防止了数据检索信息的泄密问题。只有当数据访问属性与谓词 φ 匹配认证成功时，返回密文 C，可对密文 C 进行访问。

5) Decrypt (KEY, C)。当对访问者的身份认证和访问属性认证成功后，系统会返回存储密文 C 给访问者，输入 KEY 后访问成功；否则 Decrypt (KEY, C)=0，输出“⊥”，访问失败。

为实现域隔离器对租户存储数据的快速检索及数据安全访问权限的有效判断，本文利用租户 TID 和数据令牌的唯一性，设计了双 Hash 的索引结构，如图 8 所示。

由图 8 可知，在数据存储标签的检索中，将租户标识 TID 作为第一级 Hash 索引，用于锁定租户；数据令牌的 Hash 值作为第二级索引直接指向数据存储的地址，从而一次定位租户下的某个数据。H(TK)为查询令牌的 Hash 值，与 TID 作为租户密文数据的存储安全标识。通过对令牌进行 Hash 运算，结合虚拟域安全标签来保证租户数据存储时的完整性和机密性。为方便实现数据存储标签与数据的绑定及数据的快速检索，可采用统一的数据库绑定语句来实现，绑定结构如表 1 所示。

表 1　　数据存储标签格式

字段名称	字段类型	主键	允许空	字段说明
TID	bit(32)	否	否	租户的 TID 码
H(TK)	varchar(16)	是	否	查询令牌的 Hash 值

数据存储标签加解密数据可由数据库中自带的加密函数来完成，例如 MySQL 中的 AES_ENCRYPT 函数和 AES_DECRYPT 函数。

① 当数据加密存入数据库时，示例如下。

INSERT INTO TableX(HTK, TID , C)) VALUES

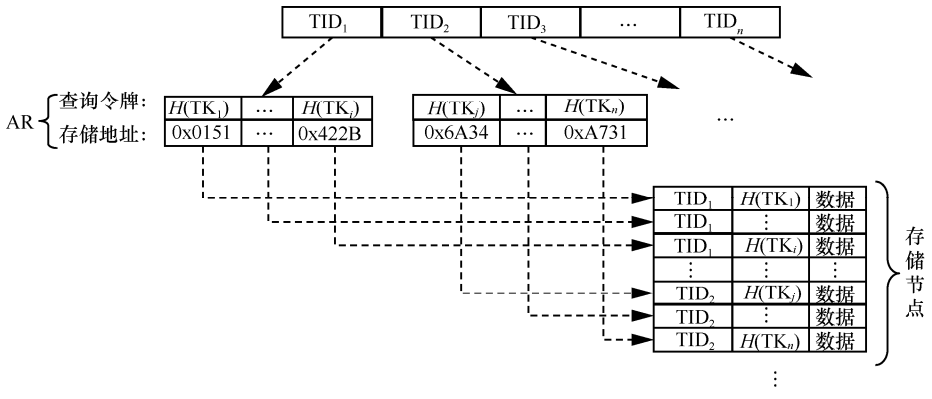


图 8　租户数据检索的索引结构示意

(Hash_TK, DLebal. TID, AES_ENCRYPT('Data' , 'KEY'))

② 当访问数据时，示例如下。

SELECT AES_DECRYPT('*C*' , 'Key') FROM TableX TID=DLebal.TID AND HTK= Hash_TK

4.3 多维度的租户数据隔离控制规则

4.3.1 多租户虚拟域内数据聚合推导控制规则

租户数据聚合推导控制规则是指为降低因数据聚合而引起的泄密风险，通过分析租户数据控制标签信息间的关系，推演出数据间由于聚合而推导出高级别信息的可能性，以此制定相应的安全控制规则，控制数据的访问安全。此关系数据主要包括相似数据与关联数据。相似数据是指标签属性及数据内容相似的同类租户数据；关联数据是指具有隐含推导关系的数据，也称为不兼容数据。

定义 1 关联数据聚合推导。令 $Data_i$ 和 $Data_j$ 具有强关联性，记作 $Data_i.Label\Theta Data_j.Label$，当 $Data_i$ 和 $Data_j$ 发生聚合时，通过聚合分析数据控制标签的信息能够推导出的数据安全级别大于 $Data_i$，$Data_j$ 安全级别的可能性超过特定的阈值，则 $Data_i$ 和 $Data_j$ 具有不兼容客体聚合推导的安全问题。

定义 2 相似数据聚合推导。当相似客体 $Data_1,Data_2,\cdots,Data_n$ 中的 $k(k\leqslant n)$个数据发生聚合时，通过分析数据的控制标签信息推导出信息的安全级别大于这 n 个数据的最高安全级别，则称这 n 个客体存在相似数据聚类推导的安全问题。记作 $Sim\_Data.S(Data_1.Label,Data_2.Label,L,Data_i.Label,k)\geqslant\forall Data.S,k\leqslant n$。

规则 1 if $Data_i.Label\Theta Data_j.Label$，then $Data_i\otimes Data_j$。

规则 1 说明若 $Data_i$ 和 $Data_j$ 为不兼容客体，则 $Data_i$ 和 $Data_j$ 聚合推导出的数据信息安全级别 S 比 $Data_i$ 和 $Data_j$ 具有的信息安全级别要高；安全级别小于 S 的用户，如果曾经访问过 $Data_i$，则禁止访问 $Data_j$，反之亦然；如果安全级别小于 S 的虚拟机中已经存在数据 $Data_i$，则不允许数据 $Data_j$ 的流入。

规则 2 if $Sim\_Data.S(Data_1.Label, Data_2.Label,\cdots,Data_k.Label)\geqslant\forall Data.S$，$k\leqslant n$，则允许访问这类数据的数目小于 k。

规则 2 说明 $Data_1,\cdots,Data_n$ 聚合推导出的数据级别 S 高于所有数据；相似数据聚类问题存在 2 种阈值的情况，一种是数量阈值，即从 $Data_1,\cdots,Data_n$ 中任取 k 个数据都可以推导出更高级别数据信息，安全级别小于 S 的用户，仅能够访问 k–1 个相似数据，或者安全级别小于 S 的虚拟机中仅仅能够同时流入 k–1 个相似数据；另一种是性质的情况，即 $Data_1,Data_2,\cdots,Data_n$ 中存在着 $Data_i,Data_j,\cdots,Data_k$ 共 k 个数据，只要包含这 k 个数据中的任何一个或者多个均能推导出高级别的数据信息，则 k 个数据也被称为例外数据，若 $Data_1,Data_2,\cdots,Data_n$ 中存在例外数据，则不允许访问例外数据。

4.3.2 多租户虚拟域内数据安全通道的控制规则

为保证租户虚拟域内数据的传输安全，确保安全通道中数据的完整性、机密性。结合数据控制标签的设计，本文制定了域内安全通道的控制规则。

定义 3 域内互联成员（IM, interconnection member）。IM 是指租户虚拟域内参与数据安全传输的成员，包括源 IM_s 和目的 IM_d。IMA 表示安全传输时一个完整的互联成员集，$\exists IM_s, IM_d\in IMA$。

定义 4 互联实体（IE, interconnected entity）。IE 是指虚拟域内互联的安全设备或者组件，对互联成员进行安全保护。

定义 5 虚拟域安全传输通道（VDST, virtual domain security tunnel）。VDST 是指为租户虚拟域内数据提供安全传输服务的通道。形式化定义为 $VDST_i=\{<IM_s.ie_s, IM_d.ie_d>, S_{VDST}, tSA_{VDST}\}$，其中 $<IE_s, IE_d>$指安全通道建立在 2 个互联实体间；S_{VDST} 指安全通道的安全级别；tSA_{VDST} 指域内安全关联，为了安全传输对所需安全要素的一种安全协定，比如加密算法、密钥协商、传输方向等。

规则 3 定级。$VDST_i$ 的安全级别是由通道所传输数据的级别决定的。规则 3 描述如下。

规则 3.1 if $(IM_s.ie_s\rightarrow IM_d.ie_d)$ then $\{S_{VDST}=S_{IM_s.ie_s}=S_{DLabel_s};\}$。

规则 3.1 说明安全通道级别与通道数据流源控制标签的安全级别保持一致。

规则 3.2 if $(IM_s.ie_s\leftrightarrow IM_d.ie_d)$ then $\{S_{VDST}=S_{IM_s.ie_s}=S_{DLabel_s}=S_{IM_d.ie_d}=S_{DLabel_d};\}$。

规则 3.2 说明当数据流向为双向时，安全通道的级别应与两端数据控制标签的安全级别一致。

规则 4 保护。虽然安全通道具有单向性，但在实际地租户虚拟域中，不免存在着双向数据流，为有效地保护安全通道中的租户数据，制定了安全关联保护规则，介绍如下。

规则 4.1 设安全关联 tSA_i 包括 $\overrightarrow{tSA_i}$ 和 $\overleftarrow{tSA_i}$，分别表示正反方向上的安全关联，假设数据流的流向为 $IM_s.ie_s \leftarrow IM_d.ie_d$，保护规则如下。

1) if ($S_{DLabel_s} > S_{DLabel_d}$)then { $I(\overrightarrow{tSA_i}) > I(\overleftarrow{tSA_i})$;}

2) else { $I(\overrightarrow{tSA_i}) < I(\overleftarrow{tSA_i})$;}

规则 4.1 说明安全通道的安全关联存在单向性，即要求正反方向上各有一个不同强度的安全关联；不论是哪个方向上的租户数据流，tSA 的安全强度都与数据的源端标签的安全级相关。$I(*)$表示安全关联的强度。

规则 4.2 设数据流的流向为 $IM_s.ie_s \leftrightarrow IM_d.ie_d$，保护规则如下。

$$IM_s.ie_s \leftrightarrow IM_d.ie_d \Rightarrow S_{IM_s.ie_s} = S_{DLabel_s} = S_{IM_d.ie_d} = S_{DLabel_d} \Rightarrow I(\overrightarrow{tSA_i}) = I(\overleftarrow{tSA_i})$$

规则 4.2 说明数据流为双向时，正反方向上 tSA 的强度相同，即此时的 tSA 是双向的。

规则 5 传输无干扰。若 $Data_1, Data_2, \cdots, Data_n$ 存在相似数据聚合推导问题或者 $Data_i$ 和 $Data_j$ 存在不兼容数据聚合推导问题，则安全通道传输这些租户数据时，应做到通道的无干扰。规则如下。

1) if (∃ (($Data_i.Label \Theta Data_j.Label$)‖($Sim\_Data.S$ ($Data_2.Label, Data_2.Label, \cdots, Data_i.Label$) ≥ $\forall Data.S$))) then {

2) $VDST_{Data_i} \cap VDST_{Data_j} = \phi$;

3) $VDST_{Data_1} \cap VDST_{Data_2} \cap \cdots \cap VDST_{Data_k} = \phi$;}

规则 5 说明租户数据间存在聚合推导问题，禁止共享安全通道进行传输。依照规则 1 和规则 2 的要求，应选择不同的安全通道进行安全传输。

由于租户共享云计算资源，并且域内资源分散分布，为保证租户域内数据的安全传输，还制定了安全通道的交换与转发规则，规则如下。

规则 6 安全交换与转发。假设数据 Data 将从用户 U_i 传输到用户 U_j，分别由互联实体 ie_i 和 ie_j 来保护，$DR.ie_k$ 为 DR 上的安全组件，$VDST_m$ 的两端互联实体分别为 ie_i 和 $DR.ie_k$，$VDST_n$ 的两端互联实体分别为 ie_j 和 $DR.ie_k$，安全交换与转发规则如下。

规则 6.1 安全通道交换。

1)if ($ie_i \xrightarrow{\text{Data from } T_i \text{ to } T_j} ie_j$ && $DR.ie_k \leftarrow$ visible (Data)) then {

2) $\begin{matrix} \text{Data} \\ U_i \end{matrix} \rightarrow \begin{matrix} \text{Encrypt}(VDST_m(\text{Data})) \\ ie_i \end{matrix} \xrightarrow{VDST_m} \begin{matrix} \text{Decrypt}(VDST_m(\text{Data})) \\ \text{Encrypt}(VDST_n(\text{Data})) \\ DR.ie_k \end{matrix} \xrightarrow{VDST_n} \begin{matrix} \text{Decrypt}(VDST_n(\text{Data})) \\ ie_j \end{matrix} \rightarrow U_j$

3) Trans_Type=1;}

其中，visible(*)是可视函数，表示对*是可见的；Encrypt()表示安全通道加密封装，Decrypt()表示安全通道解封装。规则 6.1 说明：①租户数据通过安全通过进行交换时，需要通过所属域隔离器下安全组件 $DR.ie_k$ 的认证后进行转发；②安全通道交换的数据流动方向具有单向性；③安全通道交换的实质是数据 Data 受两条安全通道保护，并在 $DR.ie_k$ 处进行安全通道的解封装、再封装处理，数据 Data 在 $DR.ie_k$ 处是原文，可对数据控制标签进行认证，认证数据的来源、聚合推导等问题。

规则 6.2 安全通道转发。假设 ie_i 与 ie_j 协商的安全通道为 $VDST_P$。

1)if ($ie_i \xrightarrow{\text{Encrypt}(VDST_P(\text{Data from } ie_i \text{ to } ie_j))} ie_j$ &&$DR.ie_k \leftarrow$ invisible (Data)) then {

2) $\begin{matrix} \text{Encrypt}(VDST_m(VDST_P(\text{Data}))) \\ ie_i \end{matrix} \xrightarrow{VDST_m} \begin{matrix} VDST_P(\text{Data}) = \text{Decrypt}(VDST_m(VDST_P(\text{Data}))) \\ \text{Encrypt}(VDST_n(VDST_P(\text{Data}))) \\ DR.ie \end{matrix} \xrightarrow{VDST_n} \begin{matrix} VDST_P(\text{Data}) = \text{Decrypt}(VDST_n(VDST_P(\text{Data}))) \\ \text{Data} = \text{Decrypt}(VDST_P(\text{Data})) \\ ie_j \end{matrix}$

3) Trans_Type=2;}

其中，invisible()是不可视函数，invisible(*)表示对*是不可见的。规则 6.2 说明安全通道转发的实质是通道间的嵌套封装后沿新的安全通道进行转发；域隔离器的安全组件 $DR.ie_k$ 对数据 Data 来说是不可见，ie_k 只对封装后的数据进行再次封装，并对数据控制标签进行认证，认证数据来源的真实性，主要适用于同一用户的远程传输。

4.3.3 租户虚拟域内外数据传输控制流程

为实现租户虚拟域内外数据流动的安全隔离，在上述数据聚合推导控制规则和安全通道控制规则的基础上，本文给出了数据传输的控制流程。设租户源端数据控制标签为 $VLabel_{ts}$={TID, Hash_*S*, BAN_TIME, Trans_Type}，目的端数据控制标签为 $VLabel_{td}$={TID, Hash_*S*, BAN_TIME,Trans_Type}，数据传输流程如下。

步骤 1　校验数据分组。域隔离器对接收到的数据分组进行安全标签的校验，如果校验失败则直接丢弃，如果校验成功则转到步骤 2。

步骤 2　认证 TID，检查聚合推导问题。租户源端 ts 向其所对应的域隔离器发送数据分组时，由域隔离器认证数据分组中 TID 并根据聚合推导控制规则检查数据属性是否存在聚合推导问题，当与目的端 td 所在的虚拟域 TID 相同并不存在聚合推导问题时，允许通信，转到步骤 3，否则不允许通信。

步骤 3　检查安全级别 S。当 ts 和 td 的 TID 相同时，域隔离器检查源端 ts 的安全级别 S_{ts} 与目的端 td 的安全级别 S_{td}，此时分为以下 3 种情况。

1) 当 $S_{ts}<S_{td}$ 时，允许发送数据分组到目的端 ds。

2) 当 $S_{ts}>S_{td}$ && BAN_TIME!=0 时，且数据分组在特殊的需求下从高安全级向低安全级传输，需要为目的端 td 设置一个临时的安全级别(S_{ts}，S_{td}，BAN_TIME)，BAN_TIME 为保密期限，记录在 DR 中，在这个保密期限内目的端 td 只能发送数据到大于或等于安全级 S_{ts} 的虚拟机，当期限解除后恢复 td 原有的安全级别。

3) 当 $S_{ts}>S_{td}$ && BAN_TIME==0 时，表示数据分组由高安全级非法向低安全级发送数据。

当为情况 1)和 2)时，转到步骤 4；当为情况 3)时，禁止通信。

步骤 4　选择传输方式。当上述认证全部成功后，域隔离器按照安全通道控制规则判断传输方式，若 Trans_Type==1，按照规则 6.1 进行交换数据分组到目的端 td；若 Trans_Type==2，按照规则 6.2 转发数据分组到目的端 td。

上述控制流程说明：1)所有转发或者交换的数据必须经过域隔离器的认证和审查，防止出现域间数据的交叉流动及域内数据的聚合推导等问题；2)针对数据的单向流动，正常情况下，低级别的信息只能通过安全通道流入高级别的目的端，信息流入模式分为读、写、更新、隔离访问；当高级别信息流入低级别目的端时，必须设置保密期限并设置目的端的安全级别范围，防止目的端向安全级别介于(S_0, S_n)的用户传递信息时造成泄密；3) 当源端与目的端的安全级别相同时，数据可以双向传输。

4.3.4　租户安全标签与数据分组的绑定方法

为了实现租户虚拟域内数据的安全传输，在不影响正常通信的情况下，需要完成数据分组与安全标签的绑定，本文在考虑租户网络的适应性及不增加数据分组长度的同时，通过将安全标签添加在数据分组 IP 头中的选项字段（IP 安全选项（IPSO, IP security option））中来完成。

IP 数据分组首部的可变部分是一个选项字段，用于排错、测量以及安全等措施，此字段的长度可变，安全选项的最大长度为 20 B。为便于数据分组携带安全标签，安全标签格式设计如表 2 所示。

表 2　　**安全标签格式设计**

字段名称	字段长度/B
类型字段	1
安全标签长度	1
TID 码	4
Hash($S\|\|T$)	4
BAN_TIME	4
Trans_Type	2
Hash(VLabel)	4

1) 类型字段，占 1 B，用于表示安全标签，以区别于其他的业务。

2)安全标签长度，占 1 B，用于表示安全标签的长度。

3) TID 码，占 4 B，数据分组的虚拟域标识。

4) Hash($S||T$)，占 4 B，其中 S 为数据分组的安全级别，T 为租户与 DR 协商生成的秘密序列码。在此加入 T 的目的是使数据分组安全级别 S 能够隐式传输，防止恶意租户得到数据分组的级别信息。

5) 保密期限 BAN_TIME，占 4 B，数据流动转发的时间限制，无限制时为全 0。

6) Trans_Type，占 2 B，标识数据传输方式。

7) Hash(VLabel)，占 4 B，用于实现整个数据控制标签的检验，防止标签中的信息被篡改。

与文献[27]中采用 IP in IP 的方式添加租户 ID 到数据分组中来控制租户数据分组流动相比，本文通过 IP 头的选项字段来携带安全标签，不仅可以实现安全标签随分组传递和认证，以及租户虚拟域内虚拟机间数据的安全流动，而且没有修改数据分组的大小和结构，对正常租户虚拟域的通信基本没有性能影响。

5　方法安全性证明

云计算中，数据的泄露可以看作非法的信息流动。因此云下租户隔离的本质要求就是对租户安全域内外的信息流的安全控制。为证明多租户虚拟隔离域构建机制的安全性，本文引入了无干扰理论，

该理论由 Goguen 等[34]提出，奠定了无干扰理论研究的基础，但只适用于具有传递性质的安全策略环境，显然在云租户隔离的环境下具有严重的局限性。因此 Meyden 针对传递性的限制问题，提出了具有 TA-安全和 TO-安全的非传递无干扰模型[35]，本文借助 TA-安全的非传递无干扰模型来证明该机制构建下的租户虚拟域的安全性。

为方便对隔离系统安全性证明的描述，将涉及的元素进行了形式化定义，如表 3 所示。

基于元素的定义，判定定理如下。

首先，介绍 ta 函数[36]。对于系统 $M(D,\mapsto)$，$v\in D$，函数 ta: $A^*\mapsto \mathrm{T}(\{\varepsilon\}, A)$，具体定义如下。

$$\mathrm{ta}_v(\alpha a)=\begin{cases}\varepsilon,\alpha a\text{为空序列}\\ (\mathrm{ta}_v(\alpha),\mathrm{ta}_{\mathrm{dom}(a)}(\alpha),a),\mathrm{dom}(a)\mapsto v\\ \mathrm{ta}_v(\alpha),\text{其他}\end{cases}$$

针对 ta 函数定义，给出了系统安全的定义[35]。在 $M(D,\mapsto)$中，对 $\forall v\in D$，$\forall\alpha\in A^*$ 和 $\alpha'\in A^*$，若 $\mathrm{ta}_v(\alpha)=\mathrm{ta}_v(\alpha')$，均有 $\mathrm{obs}_\mu(s_0\alpha)=\mathrm{obs}_\mu(s_0\alpha')$，那么系统 $M(D,\mapsto)$对于$\mapsto$是 TA-安全的。

基于上述定义，系统 TA-安全的判定定理为若系统 $M(D,\mapsto)$存在关于策略$\mapsto$的弱展开，那么 $M(D,\mapsto)$关于策略$\mapsto$是 TA-安全的，弱展开需要满足以下 3 个条件（$\sim_v$表示域 v 下的等价族）。

1) 输出一致性（OC, output consistency）。若 $s\sim_v t$，那么 $\mathrm{obs}_v(s)=\mathrm{obs}_v(t)$。

2) 弱单步一致性（WSC, weak step consistency）。若 $(s\sim_v t)\cap(s\sim_{\mathrm{dom}(a)} t)$，那么 $sa\sim_v ta$。

3) 策略服从（LR, local rule）。若 $\mathrm{dom}(a)\not\leadsto v$，那么 $s\sim_v sa$。

系统 $M(D,\mapsto)$是指满足本文云租户虚拟隔离域构建机制（包括术语、定义、算法及规则等）的云租户隔离系统，利用上述系统安全判定定理进行系统 $M(D,\mapsto)$的安全性证明。

证明前首先依据基于 L-DHT 的租户虚拟域隔离构建方法，给出所有虚拟域操作需要满足的条件，假设条件（AC, assumed condition）如下。

AC1　只读。在隔离系统 $M(D,\mapsto)$中，若 $e\in r(\mu)$，那么 $(e.\mathrm{TID}==\mathrm{TID}\mu)\wedge((\mathrm{ie}_d.S\geqslant e.S)\vee((\mathrm{ie}_d.S\leqslant e.S)\wedge(\mathrm{ie}_d.S=(e.S,\ \mathrm{ie}_d.S,\ \mathrm{ban\_time}))))\wedge\neg$ Aggregation $(\mu,\ e)\wedge\neg\mathrm{VDST}_k(\mathrm{Data},\ \mathrm{tSA}_k)\wedge(\mathrm{VLabel}_d=\mathrm{VLabel}_d)$。

AC2　只写。在隔离系统 $M(D,\mapsto)$中，若 $e\in a(\mu)$，那么 $(e.\mathrm{TID}==\mathrm{TID}\mu)\wedge((\mathrm{ie}_d.S\geqslant e.S)\vee((\mathrm{ie}_d.S\leqslant e.S)\wedge(\mathrm{ie}_d.S=(e.S,\ \mathrm{ie}_d.S,\ \mathrm{ban\_time}))))\wedge\neg$ Aggregation $(\mu,\ e)\wedge\neg\mathrm{VDST}_k(\mathrm{Data},\ \mathrm{tSA}_k)\wedge(\mathrm{VLabel}_d=(\mathrm{VLabel}_d,\ \mathrm{VLabel}_e))$。

AC3　读写。在隔离系统 $M(D,\mapsto)$中，若 $e\in w(\mu)$，那么 $(e.\mathrm{TID}==\mathrm{TID}\mu)\wedge(\mathrm{ie}_d.S=e.S)\wedge\neg$ Aggregation$(\mu,\ e)\wedge\neg\mathrm{VDST}_k(\mathrm{Data},\ \mathrm{tSA}_k)\wedge(\mathrm{VLabel}_d=(\mathrm{VLabel}_d,\ \mathrm{VLabel}_e))$。

AC4　隔离存储。在隔离系统 $M(D,\mapsto)$中，若 $\mu,v\overset{\mathrm{Map}}{\Rightarrow}\mathrm{DR}_i$，$e\in\mathrm{storage}(\mu)$，$e'\in\mathrm{storage}(v)$，那么 $(e.\mathrm{TID}==\mathrm{TID}\mu!=e'.\mathrm{TID}==\mathrm{TID}v)\wedge\mathrm{Encrypt}_{\mathrm{KEY}\mu}(e)\wedge\mathrm{Encrypt}_{\mathrm{KEY}v}(e')\wedge(\mathrm{Hash}(f(A_e)\|\ \mathrm{TID}\mu)=\mathrm{TK}_e)\wedge(\mathrm{Hash}(f(A_{e'})\|\ \mathrm{TID}v)=\mathrm{TK}_{e'})\wedge((e.\mathrm{address}\cup e'.\mathrm{address})$

表 3　　**云租户隔离系统的形式化定义**

元素集	含义
$M(D,\mapsto)$	租户隔离系统 $D=\{\mu, v, \mathrm{DR}, \cdots\}$表示租户虚拟域集合，“$\mapsto$”表示数据安全控制规则(信息流策略)
S	系统的状态集 $S=\{S_0,\cdots,S_i,\cdots,S_n\}$，$S_0$ 表示系统的初始状态，S_i 表示系统的中间某状态
A	租户动作集 $A=\{r, a, w, \mathrm{storage}, \cdots\}$，$r, a, w$, storage 分别表示读、只写、写入、隔离存储
$\mathrm{obs}_\mu(s_i)$: $s_i\times\mu\to O$	$\mathrm{obs}_\mu(s_i)$表示虚拟域 μ 在状态 s_i 下所观察到的输出，O 表示输出集
step(S, A): $S\times A\to S$	状态转换函数，$s\alpha$ 表示状态 s 经过动态序列 α 后的状态；同时，$s\times\varepsilon=s$, ε 为空序列，$s\alpha a$=step($s\alpha$,a)
dom(a)	$a\in A$，动作 a 所对应的虚拟域
(T, R)	每个虚拟域内的组成，包括用户集和资源集，$T=\{U_1, U_2,\cdots,U_n\}$，$R=\{R_1,R_2,\cdots,R_n\}$
$\mathrm{val}_\mu(s,n)$:$s\times n\to V$	在虚拟域 μ 中，名称为 n 的资源在状态 s 时的取值
$A(\mu)$:$\mu\to M(e)$	域 μ 中可被执行 A 动作的资源集合，例如：$r(\mu)$:$\mu\to M(e)$: $M(e)$为域 μ 中可读的资源集合
Aggregation:$D\times R\to T(e)$	Aggregation 分为 impat 和 sim 函数，分别表示域 μ 中与 R 不兼容和具有相似聚合问题的资源集合
Encry: $\mathrm{VDST}_k(\mathrm{Data}, \mathrm{tSA}_k)$	利用 tSA_k 对传输的数据进行安全通道的协议封装、加密、认证等处理，安全强度与安全关联相关
Decry: $\mathrm{VDST}_k(\mathrm{Data}, \mathrm{tSA}_k)$	利用 tSA_k 对安全通道中的信息进行协议解封装、解密、认证等处理

$\subset$ DR_i-St. address)。

AC5　管理隔离。在隔离系统 $M(D, \mapsto)$中，若 $\mu,\upsilon \overset{Map}{\Rightarrow} DR_i$，$\forall a \in A$; $\mu,\upsilon \in$ D-DR, $dom(a) \nrightarrow \mu$，那么 $dom(a) \neq DR$，$\upsilon \cap \mu = \varnothing$。

定理 1　云租户隔离系统 $M(D, \mapsto)$关于策略“$\mapsto$”是 TA-安全的。

根据系统 TA-安全的判定定理，要证明定理 1，必须证明系统 $M(D, \mapsto)$关于策略“$\mapsto$”的弱展开满足 OC、WSC 和 LR，证明如下。

证明

1) OC。①当 $e \in r(\mu)$时，e 是虚拟域 μ 中只读的资源集合，由 AC1 可知，资源的只读操作只在同一虚拟域内且不具有聚合推导关系时发生，并且只读操作前后资源的值未发生改变，即 $(val_\mu(s,e)= val_\mu(t,e)$，因此在等价的只读状态($s \sim_\upsilon t$)下，必有 $obs_\mu(s)=obs_\mu(t)$。②当 $e \in a(\mu)$时，e 是虚拟域 μ 中只写的资源集合，由 AC2 可知，当 $ie_d.S \geqslant e.S$，只有满足¬ Aggregation(μ, e)时才允许执行写入操作，并且在写入后并未改变原有资源的值和目的端的安全级别，也不会影响目的端对其他资源的访问关系，此时若 $s \sim_\mu t$，必有 $obs_\mu(s)=obs_\mu(t)$；当 $ie_d.S \leqslant e.S$，会在目的端写入的高级别信息的保密期限内临时调整 $ie_d.S=(e.S, ie_d.S, ban\_time)$，因此不会使高级别的信息流入低于 $e.S$ 的目的端，也不会使高于 $ie_d.S$ 的资源流入 ie_d 造成泄密，因此 $obs_\mu(s)=obs_\mu(t)$仍然成立。③当 $e \in w(\mu)$时，e 是虚拟域 μ 中读写的资源集合，由 AC3 可知，只有当位于同一虚拟域内安全级别相同时，才允许相互读写，并且由上述①和②证明可知，当¬ Aggregation(μ,e)时，执行读写操作前后满足 $val_\mu(s,e)= val_\mu(t,e)$，因此若 $s \sim_\mu t$，$obs_\mu(s)= obs_\mu(t)$成立。④依据 AC4，当 $\mu,\upsilon \overset{Map}{\Rightarrow} DR_i$，$e \in$ storage(μ)，$e' \in$ storage(υ)时，虽然不同租户的数据共享同一存储节点 DR_i-St，但会根据 TID 来区别不同租户，并且利用各自的密钥加密数据，即 $Encrypt_{KEY\mu}(e) \wedge Encrypt_{KEY\upsilon}(e')$，由于 TID 的唯一性和 KEY 的保密性，在某一状态下，不同存储地址上数据值固定且互不影响且，当 $s \sim_\mu t$ 时，虚拟域 μ 的存储地址范围相同，并且每个地址对应的值相同，因此必有 $obs_\mu(s)= obs_\mu(t)$。⑤ AC5 是域隔离器 DR 对 μ 的管理操作，因此在相同管理状态下的输出是一致的。此外，在信息的流动过程中，受安全通道的保护，资源值不会在传输时发生改变。

综上所述，系统 $M(D, \mapsto)$符合输出一致性。

2) WSC。根据无干扰模型的定义，可把 WSC 的证明等价转换为证明 $val_\mu(step(s,a))= val_\mu(step(t, a))$，即资源 e 在状态 $s \sim_\mu t$ 下执行相同操作 a 的取值是一致的。

① 当 $dom(a) \nrightarrow \mu$ 时，此时 $dom(a)$对 μ 没有任何影响，即不会改变 μ 的地址范围，更不会改变 μ 中资源的取值，即 $val_\mu(s,e)=val_\mu(t,e) \Rightarrow val_\mu(step(s,a),e)= val_\mu(step(t,a),e)$。

② 当 $dom(a) \mapsto \mu$ 时，根据 AC1~AC5，可分为以下 3 种情况讨论。

情况 1　$dom(a)=\mu$，即动作 a 为对 μ 中的资源进行读写或存取操作，$e \in (a(\mu) \vee w(\mu) \vee storage(\mu))$，由 AC2~AC4 可知，执行 a 操作后，资源 e 的安全级别包含关系、完整性、机密性等安全属性并未改变，由 WSC 性质的假设条件 $(s \sim_\upsilon t) \cap (s \sim_{dom(a)} t)$，因此 $val(s, e)=val(t, e) \Rightarrow val(step(s, a), e) = val(step(t, a), e)$。

情况 2　$dom(a) \neq \mu$，$dom(a)$=DR，根据 AC5，$dom(a)$为 μ 的管理域，由规则 7 可知，操作 a 为域隔离器 DR 对的管理操作，包括资源的回收与分配，当分配新的资源时，其资源值为 0；当回收空余资源时剩下的资源值不会改变，因此在状态 s 和 t 下，执行相同的 a 操作时，虚拟域 μ 的范围及资源的取值相同，即 $val(step(s,a),e)=val(step(t,a),e)$。

情况 3　$dom(a) \neq \mu$，μ=DR，租户执行动作 a 读写存储在 DR 上的租户信息，由 $s \sim_{dom(a)} t$ 可知，$obs_{dom(a)}(s)= obs_{dom(a)}(t)$，因此租户执行动作 a 后，将 $dom(a)$中相同状态下的租户信息写入 DR 中，由 DR 对不同虚拟域进行隔离管理，不会影响其他租户信息的值，可得 $val_\mu(step(s,a),e)= val_\mu(step(t,a),e)$。

综上所述，当 $(s \sim_\upsilon t) \cap (s \sim_{dom(a)} t)$ 时，均有 $val_\mu(step(s, a), e)=val_\mu(step(t, a), e) \Rightarrow obs_\mu(step(s, a))= obs_\mu(step(t,a))$，即 $sa \sim_\mu ta$ 成立。

3) LR。针对 LR 的证明，可以等价为证明 $dom(a) \nrightarrow u \Rightarrow obs_u(s)=obs_u(step(s,a))$，通过分析 AC1~AC5 可知，在此分为 3 种情况分析。

情况 1　针对 $e \in r(\mu)$时，根据 AC1 可知，由于只读操作 a 执行前后虚拟域 u 的地址范围及资源值

并未发生改变，符合 $\mathrm{dom}(a)\nrightarrow\mu$，显然有 $\mathrm{obs}_u(s)=\mathrm{obs}_u(\mathrm{step}(s,a))$。

情况 2 针对 $e\in(a(\mu)\vee w(\mu)\vee\mathrm{storage}(\mu))$，由于动作 a 中包含对虚拟域 u 的写入操作，在此通过逆否命题来进行证明，即证明 $\mathrm{obs}_u(s)\neq\mathrm{obs}_u(\mathrm{step}(s,a))\Rightarrow\mathrm{dom}(a)\mapsto\mu$，由 AC2~AC4 可知，在安全策略$\mapsto$限制下，执行动作 a，操作前后 μ 的地址范围及资源 e 的取值会发生变化，即 $\mathrm{val}_\mu(s,e)\neq\mathrm{val}_\mu(\mathrm{step}(s,a),e)\Rightarrow\mathrm{obs}_u(s)\neq\mathrm{obs}_u(\mathrm{step}(s,a))$，又因域 $\mathrm{dom}(a)$的信息流入虚拟域 u，因此 $\mathrm{dom}(a)\mapsto u$ 成立，由此可知 $\mathrm{obs}_u(s)\neq\mathrm{obs}_u(\mathrm{step}(s,a))\Rightarrow\mathrm{dom}(a)\mapsto u$ 成立，即 $\mathrm{dom}(a)\nrightarrow u\Rightarrow\mathrm{obs}_u(s)=\mathrm{obs}_u(\mathrm{step}(s,a))$成立。

情况 3 针对 $\mathrm{dom}(a)\neq\mu$ 时，由于 $\mathrm{dom}(a)\nrightarrow\mu$，由 AC5 可知，$\mathrm{dom}(a)\neq\mathrm{DR}$，因此虚拟域 μ 的地址范围不会改变，又因为 $\mathrm{dom}(a)\nrightarrow\mu$，所以动作 a 执行后 μ 中资源的值仍然不会改变，由此可知 $\mathrm{val}_\mu(s,e)=\mathrm{val}_\mu(\mathrm{step}(s,a),e)\Rightarrow\mathrm{obs}_u(s)=\mathrm{obs}_u(\mathrm{step}(s,a))$。

根据 $\sim_\mu$ 的定义，若 $\mathrm{dom}(a)\nrightarrow u$，则有 $\mathrm{obs}_u(s)=\mathrm{obs}_u(\mathrm{step}(s,a))$，即 $s\sim_\mu sa$ 成立。

综上所述，定理 1 符合 TA-安全判定定理的 3 个条件，证明了基于 L-DHT 的多租户虚拟隔离域隔离构建方法的无干扰性，能够有效隔离租户虚拟域，保证租户数据的传输和存储安全。证毕。

6 方法可行性验证与分析

6.1 域隔离器分组解析能力可行性验证

基于胖树（Fat-Tree）交换网络，本文对域隔离器的解析能力进行了仿真测试，测试环境为 OPNET 14.5+Visual studio 2013，实验环境为 Windows 10。

利用 OPNET 模拟 8 台计算机和一台域隔离器，配置域隔离器的 CPU 为 Inter i7-4770 3.4 GHz，内存 16 GB，每台计算机创建 25 个线程，每个线程大约每秒创建 100~200 个解析请求，域隔离器负责接收分组并进行解析。模拟 8 台服务器来分别向该域隔离器不间断发送分组。统计域隔离器处理解析请求时的 CPU 占用率（每隔 1 min 计算一次 CPU 平均占用率及每秒平均解析请求的数量，总共记录 10 次），统计结果如图 9 所示。

由图 9 可知，每秒处理 24 000~41 000 个的解析请求数量，该域隔离器 CPU 的占用率在 48%~82%。面对 64 端口组成的 Fat-Tree 网络，所能支持的服务器总数为 32 768，设每台服务器可创建 8 台虚拟主机，网络中可作为域隔离器的核心交换机有 1 024 个，每台核心交换机最少只需负责 256 台虚拟主机的解析任务，对比仿真实验结果，实际网络中的核心交换机的负载数远小于模拟实验中的负载数，因此域隔离器有足够的解析能力。

6.2 多租户虚拟域的隔离映射算法性能验证

为了验证多租户隔离映射算法的性能，对本文算法进行了仿真实验，实验包括 Hash 算法的对比与选择、DR 虚拟倍数的确定及负载均衡性的对比验证。实验环境为 python3.6.1。

1) Hash 算法的对比与选择

不同的 Hash 算法具有不同的哈希情况，直接关系到租户 TID 在 Hash 环上的分布，因此对一致性 Hash 算法中常见的 3 种哈希函数（KETAMA_Hash 算法、FNV1_32_Hash 算法、CRC32_Hash 算法）进行了数据分布均衡性的对比实验，实验设置 10 个映射节点和 10 000 个不同的

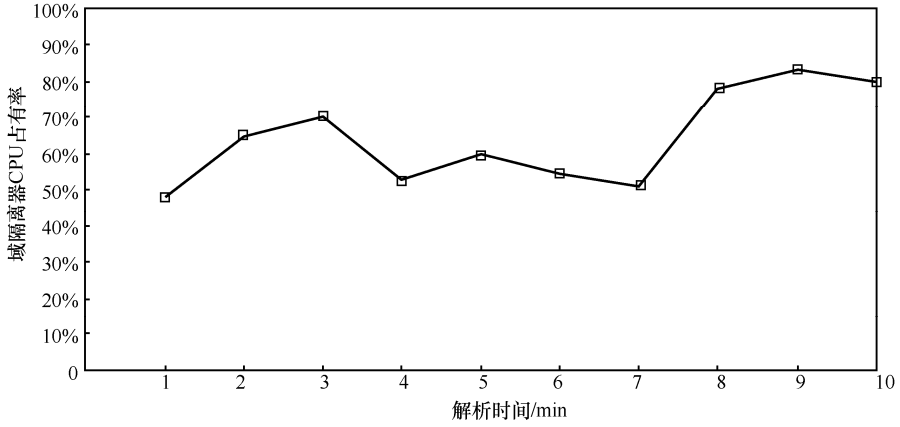


图 9 域隔离器 CPU 占用率

TID 码。映射结果如图 10 所示，其中曲线上的数字表示该 Hash 算法映射后最高和最低节点的负载数量。

在此引入系统负载差值的概念，用来描述系统整体的负载偏差情况，如式(1)所示。

$$\text{SysLoadDeviation}=\sum_{i=0}^{N}|L(i)-\overline{L}| \tag{1}$$

根据式(1)计算每个 Hash 算法的系统负载差值分别如下：KETAMA_Hash 为 648，FNV1_32_Hash 为 3 412，CRC32_Hash 为 2 160。由图 10 中曲线的变化情况和上述差值的结果可知，KETAMA_Hash 算法具有更好的哈希性，这是因为 KETAMA_Hash 算法使用的是基于 md5 的哈希函数，随机性、哈希性更好，映射更加均匀，因此本文利用其作为租户映射过程中的 Hash 运算。

2) DR 虚拟倍数的确定

虚拟节点的数量影响着租户分布式映射的结果，虚拟节点设置得更多，可以使租户的映射更均匀，DR 的负载更平衡，但虚拟节点的映射管理更复杂，容易增加虚拟节点的迁移数量；虚拟节点设置越少，无法有效保证租户初始的均衡映射，负载的不均衡也会导致频繁迁移。

因此，本文针对不同数量的域隔离器（数量的选取模拟 FAT-Tree 网络中不同端口数量下核心交换机的数量），通过不断调整虚拟倍数，找到合适的虚拟节点数量以保证租户初始时的均衡映射。在实验中，针对不同数量的域隔离器分别进行 10 次映射取平均值，每次通过随机生成 10 000 个不同的租户 TID 码，假设每台域隔离器的初始权重相同，租户均衡映射的检验标准为不断地调整虚拟倍数，并设置域隔离器间的负载差值比的阈值为 10%，即 $\frac{L_{\max}}{L_{\min}}-1\leqslant 10\%$，实验结果如图 11 所示。

图 11 中给出了不同 DR 数量下虚拟倍数的数值设置，并得出如下结论。域隔离器的数量与虚拟节点的倍数间近似成反比，当域隔离器数量越大时，虚拟倍数波动范围越小，但所需的虚拟节点总数量

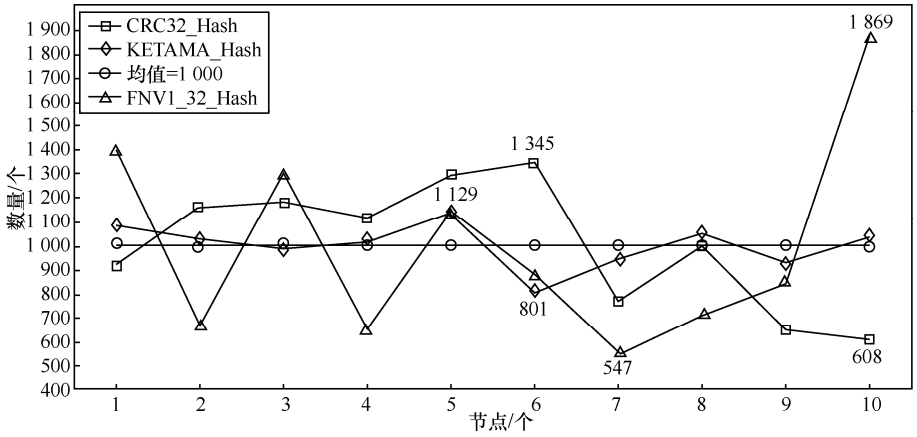


图 10　不同 Hash 算法的 TID 码分布情况

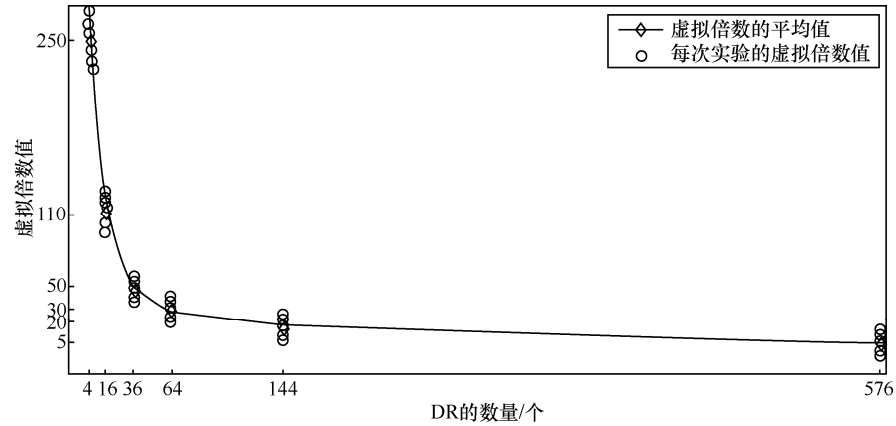


图 11　不同 DR 数量下虚拟倍数的数值设置

逐渐增大，这是因为域隔离器数量增大时，需要提高虚拟节点的数量来进一步分散租户的映射。

3) 负载均衡性的对比验证

本文为验证算法的负载平衡能力，以 10 000 个租户 TID 码为测试用例，对现阶段应用广泛的传统的一致性 Hash 算法[27]、Rendezvous Hash 算法[37]、Maglev 一致性 Hash 算法[38]和本文算法在映射完成后的最大负载差值比进行对比，如图 12 所示。实验利用了 6.2 节中确定的虚拟倍数，如表 4 所示。

表 4 实验参数

租户 TID 数量	DR 个数	虚拟节点倍数（与 DR 个数对应）
10 000	16、36、64、144	110、50、30、20

由图 12 可知，比起现有的 3 种映射算法，本文算法在租户映射时分布更加均匀，能够使节点的负载更平衡。

为测试 DR 变化后算法的负载平衡能力，本文通过增加一个 DR，对比 Rendezvous Hash 算法、Maglev 一致性 Hash 算法和本文算法负载调整后节点的最大负载差值比及 TID 迁移数量，如图 13 所示，其中柱形图的高度和其上方的第一个数值表示算法的最大负载差值比，上方的第二个数值表示算法负载调整过程中的迁移数量。

图 13 中给出了每种算法的最大负载差值比和 TID 迁移数量，通过对比可知，在增加一个 DR 后，本文算法的最大负载差值与 Maglev 一致性 Hash 算法相似，但本文算法在迁移数量上远低于 Maglev 一致性 Hash 算法，这是因为 Maglev 一致性 Hash 算法在负载调整过程中存在联动干扰的问题，即某个节点的增删会使查找表其他位置填充的槽位标号发生变化，导致迁移数量增多，而本文算法不会增加其他节点中额外的迁移；本文算法的迁移数量与 Rendezvous Hash 算法相似，但最大负载差值更小，迁移后的负载更均衡，这是因为 Rendezvous Hash 算法在迁移过程中过多地依赖 Hash 算法的随机性。相比之下，本文算法中加入了阈值和节点权值的概念，使负载调整的过程中能够按照阈值进行平衡，更加灵活，尽可能在满足负载差值需求的情

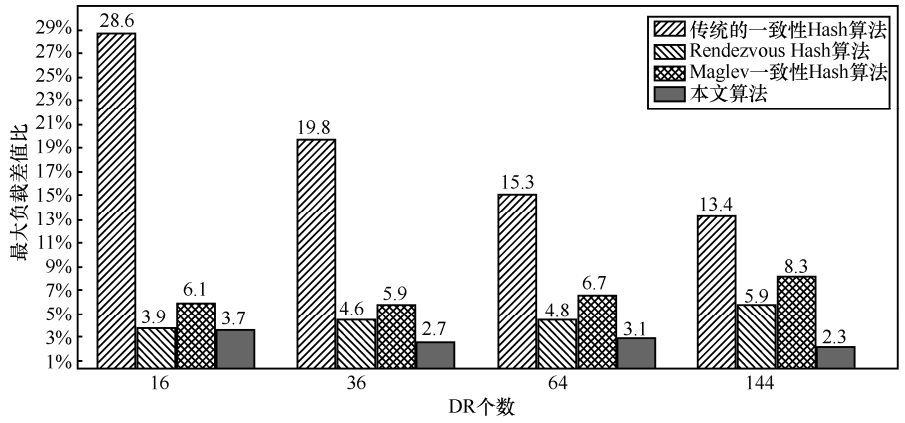


图 12 不同算法负载结果对比

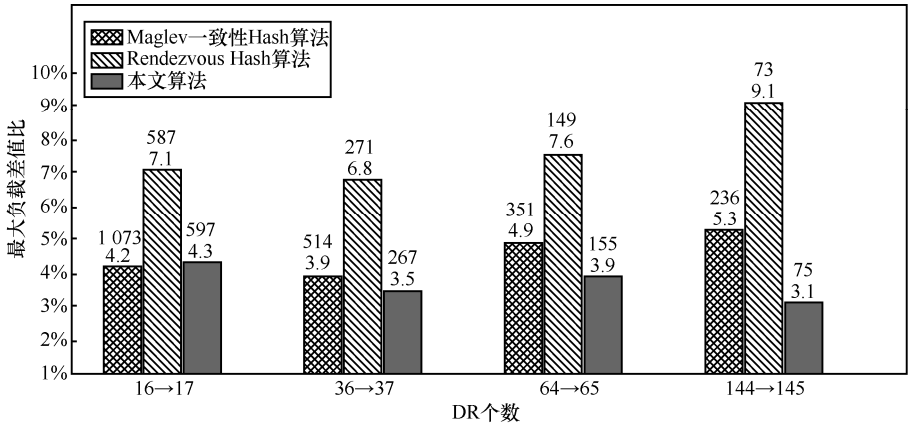


图 13 增加一个 DR 后不同算法的负载调整能力对比

况下降低租户的迁移数量。可见，本文算法能够更好地适应 DR 数量的动态变化。

此外，针对传统的一致性 Hash 算法，由于没有过多考虑负载平衡调整的过程，当添加节点或节点宕机时，只与 Hash 环上邻近的节点进行租户信息的迁移，会造成严重的负载失衡情况。

6.3　隔离存储算法有效性分析

1) 数据的访问效率分析

由于 TID 和 TK 表示租户存储数据的唯一性，本文所提出的双 Hash 索引方法利用 TID 锁定租户，利用 Hash 运算后的 H(TK)直接指向数据存储地址，能够一次获取到租户数据的存储地址，避免了对索引表的遍历，时间复杂度为 $O(1)$。为更好地体现该方法的检索效率，本文还分析了该方法与 3 种常见索引方法的时间复杂度对比，如表 5 所示。

表 5　不同索引方法的时间复杂度对比

索引方法	时间复杂度
二叉树索引	$O(\text{lb}\,N)$~$O(N)$
B 树索引	$O(\log N)$
Hash 索引	$O(1)$
本文索引方法	$O(1)$

由表 5 中的时间复杂度可以看出，本文索引方法的时间复杂度低于二叉树索引和 B 树索引方法，这是因为基于树的索引方法要从根节点到叶节点逐层进行关键字的查找，增加了检索的时间复杂度，即使改进过后的 B 树索引方法也需要对最底层的叶节点进行查找，而 Hash 索引方法利用数组的方式可以通过 Hash 值直接定位数据存储的地址，大大提高了数据检索的效率。虽然现有的 Hash 索引方法具有 $O(1)$的时间复杂度，但由于 Hash 函数的碰撞特性，容易将不同数据映射为同一 Hash 值，导致检索错误，容易降低检索效率。但本文方法将 Hash 索引分为 2 个步骤，先定位租户，缩小范围，再从租户内部定位租户数据存储的位置，有效降低了碰撞的概率，保证了租户数据检索的效率。

2) 数据认证访问的安全性分析

针对数据的隔离存储安全，已经在 5.1 节给出了安全性证明，在此主要分析数据访问的安全性。

首先，针对数据访问时检索的安全性，数据的访问效率分析中所提到的 3 种常见的索引方法在检索的过程中均使用了数据的关键字，如果攻击者获取到数据检索的关键字，则容易从关键字中得到数据的部分信息，从而引起租户数据信息的泄密，然而本文的索引方法利用了 TID 和 H(TK)作为索引，具有单向性、完整性和保密性，攻击者即使得到 TID 和 H(TK)也无法知道数据的任何信息，从数据检索信息的角度，防止了租户数据信息的泄露。

其次，针对租户数据的认证访问的安全性，本文从防假冒、防篡改、防泄露及属性信息的安全性等方面进行如下分析。

① 防假冒。本文通过在租户访问的每个属性认证过程中都加入了租户虚拟域安全标签 TID，由于 TID 中含有秘密序列号 T，其他租户无法获取并且即使获取到 TID 也不能判断其属于哪个租户，防止了其他恶意租户通过身份的假冒来骗取租户数据。

② 防篡改。本文在访问属性与策略谓词的匹配认证过程中，利用了 Hash 函数的单向性，攻击者无法得到 Hash 前的属性信息和 TID，防止了恶意租户对租户访问属性信息的获取和篡改。

③ 防泄露。本文将所有的数据属性拆分为策略谓词，利用循环地属性认证，只有当每个访问属性都认证成功，即所有访问属性的 Hash 值与令牌相匹配时，才允许从存储节点中返回租户数据，保证了数据访问的安全性。

④ 属性信息的安全性。本文为保证数据和访问者属性信息的安全，借助 Hash 值的唯一性和单向性，利用 Hash($f(A)$‖TID)与 TK 的数值对比来实现认证，避免暴露属性信息，防止了属性信息的泄露。

此外，该方法基于 Hash 算法的特性来设计，TID 和令牌的 Hash 还利用到数据的检索中，避免了 Hash 的多次计算，具有较高的效率。为更好地体现本文认证访问的有效性，本文给出了本文算法与 2 种常见属性认证访问方法（基于属性的访问控制（ABAC, attribute based access control）方法[39]和非对称谓词加密方法[7]）的对比，如表 6 所示。

表 6　不同认证访问方法对比

访问方法	数据安全性	细粒度认证	复杂度	云环境适应性	扩展性
ABAC	○	✓	✓	✓	✓
非对称谓词加密方法	✓	✓	✓	○	✓
本文方法	✓	✓	×	✓	✓

注：✓、○、×分别表示高、中、低。

由表 6 可知，ABAC 和非对称谓词加密方法虽然在一定程度上实现了属性的认证，保证了存储数

据访问认证的安全性，但没有对参与认证的属性信息进行保密性和完整性的处理，大多以明文的形式进行认证，存在数据属性信息泄露的风险。

7 结束语

多租户安全域的构建是租户数据安全隔离的保证。为此，本文提出了一种基于 L-DHT 的租户虚拟域隔离构建方法，针对不同层次下的安全隔离设计了不同形式的安全标签。首先，通过结合域安全标签和一致性 Hash 算法，给出了一种多租户的分布式隔离映射算法，完成了租户资源到域隔离器的均衡映射，实现了域隔离器对租户资源的分布式管理；然后，通过数据存储标签与租户数据的安全绑定，给出了一种基于标签谓词加密的租户数据隔离存储算法，解决租户数据的隔离存取；最后，设计了多维度的租户数据隔离控制规则，利用数据控制标签与租户数据分组的绑定、解析与认证，建立了数据传输的安全通道，实现了对租户域内数据流的安全控制与隔离访问。本文方法通过层次化地分析租户虚拟域内数据的安全隔离需求，构建起租户相互间隔离的虚拟域，从而实现了租户数据间的安全有效隔离。通过仿真对比实验和无干扰理论，分别证明了本文方法的可行性及安全有效性。接下来，将对租户虚拟域边界的自动化识别以及边界内外租户信息流的安全控制技术展开详细的研究。

参考文献：

[1] LELE A. Cloud computing, in book: disruptive technologies for the militaries and security[M]. Berlin: Springer, 2018.

[2] COOK A, ROBINSON M, FERRAG M A, et al. Internet of cloud: security and privacy issues, in book cloud computing for optimization: foundations, applications, and challenges[M]. Berlin: Springer, 2018.

[3] WALIA M K, HALGAMUGE M N, HETTIKANKANAMAGE N, et al. Cloud computing security issues of sensitive data, in book: handbook of research on the IoT, cloud computing, and wireless network optimization[M]. Hershey: IGI Global, 2019.

[4] 石勇, 郭煜, 刘吉强, 等. 一种透明的可信云租户隔离机制研究[J]. 软件学报, 2016, 27(6): 1538-1548.
SHI Y, GUO Y, LIU J Q, et al. Trusted cloud tenant separation mechanism supporting transparency[J]. Journal of Software, 2016, 27(6): 1538-1548.

[5] 李顺东, 窦家维, 王道顺. 同态加密算法及其在云安全中的应用[J]. 计算机研究与发展, 2015, 52(6): 1378-1388.
LI S D, DOU J W, WANG D S. Survey on homomorphic encryption and its applications to cloud security[J]. Journal of Computer Research and Development, 2015, 52(6): 1378-1388.

[6] 杨艳, 陈性元, 杜学绘. 多机构身份及属性加密机制综述[J]. 通信学报, 2018, 39(10): 118-129.
YANG Y, CHEN X Y, DU X H. Survey of multi-authority identity-based and attribute-based encryption scheme[J]. Journal on Communications, 2018, 39(10): 118-129.

[7] 杨丹婷. 谓词加密的理论研究及推广应用[D]. 南京: 南京理工大学, 2015.
YANG D T. Research on predicate encryption theory and its popularization[D]. Nanjing: Nanjing University of Science & Technology, 2015.

[8] SUKMANA M I.H, TORKURA K A., GRAUPNER H, et al. Unified cloud access control model for cloud storage broker[C]//2019 International Conference on Information Networking (ICOIN). Piscataway: IEEE Press, 2019: 60-65.

[9] ZHOU H Z, BA H H, WANG Y J. Tenant-oriented monitoring for customized security services in the cloud[J]. Symmetry, 2019, 11(2), 252.

[10] 易倍汀. 基于 SaaS 平台的多租户间数据共享机制的设计与实现[D]. 北京: 北京邮电大学, 2014.
YING B T. The design and implementation on multi-tenant data sharing mechanism based on SaaS platform[D].Beijing: Beijing University of Posts and Telecommunications,2014.

[11] ZHANG D F, WANG Y, SUH G E, et al. A hardware design language for timing-sensitive information-flow security[J]. ACM Sigplan Notices, 2015, 50(4): 503-516.

[12] YOON M K, SALAGEGHEH N, CHEN Y, et al. PIFT: predictive information-flow tracking[C]//ACM SIGARCH Computer Architecture News. New York: ACM Press, 2016: 246-253.

[13] 郑显义, 史岗, 孟丹. 系统安全隔离技术研究综述[J]. 计算机学报, 2017, 40(5): 1057-1079.
ZHENG X Y, SHI G, MENG D. A survey on system security isolation technology[J]. Chinese Journal of Computers, 2017, 40(5): 1057-1079.

[14] ROY I, PORTER D E, BOND M D, et al. Laminar: practical fine-grained decentralized information flow control[C]//Proceedings of the 2009 ACM SIGPLAN Conference on Programming Language Design and Implementation. New York: ACM Press, 2009: 63-74.

[15] 杨永娇, 严飞, 于钊, 等. 一种基于 VT-d 技术的虚拟机安全隔离框架研究[J]. 信息网络安全, 2015 (11): 7-14.
YANG Y J, YAN F, YU Z, et al. Research on VT-d based virtual machine isolation framework[J]. Netinfo Security, 2015(11): 7-14.

[16] MALKA M, AMIT N, BEN-YEHUDA M, et al. rIOMMU: efficient IOMMU for I/O devices that employ ring buffers[J]. ACM SIGPLAN Notices, 2015, 50(4): 355-368.

[17] 吴泽智, 陈性元, 杜学绘, 等. 基于双层信息流控制的云敏感数据安全增强[J]. 电子学报, 2018, 46(9): 2245-2250.
WU Z Z, CHEN X Y, DU X H, et al. Enhancing sensitive data security based-on double-layer information flow controlling in the cloud[J]. Acta Electronica Sinica, 2018, 46(9): 2245-2250.

[18] JITHIN R, CHANDRAN P. Virtual Machine Isolation[C]// International Conference on Security in Computer Networks and Distributed Systems. Berlin: Springer, 2014: 91-102.

[19] 缪天翔. 虚拟化环境下操作系统安全性和性能的研究[D]. 上海: 上海交通大学, 2015.
MIAO T X. Research on operating system security and performance in virtualized environments[D]. Shanghai: Shanghai Jiao Tong University, 2015.

[20] QIN G, ROY G, GROOKS D, et al. Cluster optimisation using cgroups at a Tier-2[J]. Journal of Physics: Conference Series. 2016, 762(1): 012010.

[21] RANJBAR A, ANTIKANINEN M, AURA T. Domain isolation in a multi-tenant software-defined network[C]//IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC). Piscataway: IEEE Press, 2015: 16-25.

[22] 黄世轩. 基于SDN的数据中心网络流量优化策略的研究[D]. 西安: 西安电子科技大学, 2017.
HUANG S X. Research of traffic optimization strategy in data center network based on SDN[D]. Xi'an: Xidian University, 2017.

[23] SALAH K, CALERO J M A, ZEADALLY S, et al. Using cloud computing to implement a security overlay network[J]. IEEE Security and Privacy. 2013, 11(1): 44-53.

[24] KINOSHITA J, MAEDA K, YABUSAKI H, et al. Realization of VXLAN gateway-based data center network virtualization[C]//5th IIAI International Congress on Advanced Applied Informatics(IIAI-AAI 2016). Piscataway: IEEE Press, 2016: 884-887.

[25] AMAMOU A, HADDADOU K, PUGOLLE G. A TRILL-based multi-tenant data center network[J]. Computer Networks, 2014, 68(8): 35-53.

[26] 严立宇, 祖立军, 叶家炜, 等. 云计算网络中多租户虚拟网络隔离的分布式实现研究[J]. 计算机应用与软件, 2016, 33(11): 93-98.
YAN L Y, ZU L J, YE J Y, et al. Research on distributed virtual network isolation in multi-tenant cloud-computing network[J]. Computer Applications and Software, 2016, 33(11): 93-98.

[27] 孙延涛, 位月, 耿岚岚, 等. 一种基于DHT的数据中心网络租户隔离技术[J]. 北京交通大学学报(自然科学版), 2018, 42(5): 55-60.
SUN Y T, WEI Y, GENG L L, et al. A data center network tenant isolation technology based on DHT[J]. Journal of Beijing Jiaotong University(Science Edition), 2018, 42(5): 55-60.

[28] 李满. 面向 SAAS 多租户的数据隔离模式系统研究与实现[D]. 成都: 西南交通大学, 2018.
LI M. Research and implementation of data isolation mode customization system for SaaS multi-tenants[D]. Chengdu: Southwest Jiaotong University, 2018.

[29] GENTRY C. Fully homomorphic encryption using ideal lattices[C]//41st Annual ACM Symposium on Theory of Computing (STOC 2009). New York: ACM Press, 2009: 169-178.

[30] 光焱, 祝跃飞, 费金龙, 等. 利用容错学习问题构造基于身份的全同态加密体制[J]. 通信学报, 2014, 35(2): 111-117.
GUANG Y, ZHU Y F, FEI J L, et al. Identity-based fully homomorphic encryption from learning with error problem[J]. Journal on Communications, 2014, 35(2): 111-117.

[31] 段然, 顾纯祥, 祝跃飞, 等. NTRU 格上高效的基于身份的全同态加密体制[J]. 通信学报, 2017, 38(1): 66-75.
DUAN R, GU C X, ZHU Y F, et al. Efficient identity-based fully homomorphic encryption over NTRU[J]. Journal on Communications, 2017, 38(1): 66-75.

[32] 杜瑞忠, 王少泫. 基于封闭环境加密的云存储方案[J]. 通信学报, 2017, 38(7): 1-10.
DU R Z, WANG S X. Cloud storage scheme based on closed-box encryption[J]. Journal on Communications, 2017, 38(7): 1-10.

[33] IIYA A S, SERGEY V Z. An access control model for cloud storage using attribute-based encryption[C]//2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus). Piscataway: IEEE Press, 2017: 578-581.

[34] GOGUEN J A, MESEGUER J. Inference control and unwinding[C]//1984 IEEE Symposium on Security and Privacy. Piscataway: IEEE Press, 1984: 75-86.

[35] MEYDEN R V D. What, indeed, is intransitive noninterference? [C]//12th European Symposium On Research In Computer Security (ESORICS 2007). Berlin: Springer, 2007: 235-250.

[36] 吕从东. 基于无干扰模型的云计算中信息流安全研究[D]. 北京: 北京交通大学, 2016.
LYU C D. Research on information flow security of cloud computing based on noninterference models[D]. Beijing: Beijing Jiaotong University, 2016.

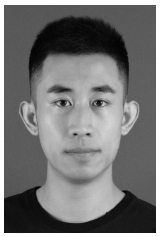
[37] ESTRIN D, HANDLEY M, HELMY A, et al. A dynamic bootstrap mechanism for rendezvous-based multicast routing[C]//IEEE Conference on Computer Communications. Piscataway: IEEE Press, 1999: 1090-1098.

[38] DENIEL E E, CHENG Y, CARLO C, et al. Maglev: a fast and reliable software network load balancer[C]//Proceedings of the 13th Usenix Conference on Networked Systems Design and Implementation. New York: ACM Press, 2016. 523-535.

[39] 王小明, 付红, 张立臣. 基于属性的访问控制研究进展[J]. 电子学报, 2010, 38(7): 1660-1667.
WANG X M, FU H, ZHANG L C. Research progress on attribute-based access control[J]. Acta Electronica Sinica, 2010, 38(7): 1660-1667.

[作者简介]

曹利峰（1981– ），男，河南禹州人，信息工程大学副教授，主要研究方向为网络安全、信息安全等。



卢新（1995– ），男，山东济南人，信息工程大学硕士生，主要研究方向为信息安全、云计算安全。

高振升（1995– ），男，河南洛阳人，信息工程大学硕士生，主要研究方向为信息安全、区块链安全。

杜学绘（1968– ），女，博士，河南辉县人，信息工程大学教授，主要研究方向为信息安全、空天网络安全、云计算与大数据安全。