doi:10.11959/j.issn.1000-436x.2017285

基于谱聚类的访问控制异常权限配置挖掘机制

房梁[1,2]，殷丽华[3]，李凤华[2]，方滨兴[1,3,4]

(1. 北京邮电大学网络空间安全学院，北京 100876；2. 中国科学院信息工程研究所信息安全国家重点实验室，北京 100093；
3. 广州大学网络空间先进技术研究院，广东 广州 510006；4. 电子科技大学广东电子信息工程研究院，广东 东莞 523808)

摘 要：将强制访问控制、自主访问控制等访问控制系统迁移为基于角色的访问控制系统可极大提高对用户权限的管理效率。为保证系统的安全性需要在迁移过程中生成正确的角色，而原系统中存在的异常权限配置给角色生成带来了极大的挑战。忽略这些异常权限配置将导致生成的角色中包含错误的权限，增加信息泄露的概率。针对访问控制中的异常权限配置发现问题，提出一种基于谱聚类的异常权限配置挖掘机制。实验结果证明，所提方案可以实现更准确的权限配置发现。

关键词：访问控制；异常权限配置；谱聚类

中图分类号：TP309.2 **文献标识码**：A

Spectral-clustering-based abnormal permission assignments hunting framework

FANG Liang[1,2], YIN Li-hua[3], LI Feng-hua[2], FANG Bin-xing[1,3,4]

(1. School of CyberSpace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China;
2. State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China;
3. Cyberspace Institute of Advanced Technology, Guangdong University, Guangzhou 510006, China;
4. Institute of Electronic and Information Engineering of UESTC in Guangdong, Dongguan 523808, China)

Abstract: Migrating traditional access control, such as mandatory and discretionary access control, into role-based access control (RBAC) lightens a practical way to improve the user-permission management efficiency. To guarantee the security of RBAC system, it is important to generate proper roles during the migration. However, abnormal user-permission configurations lead to wrong roles and cause tremendous security risks. To hunt the potential abnormal user-permission configurations, a novel spectral clustering based abnormal configuration hunting framework was proposed and recommendations were given to correct these configurations. Experimental results show its performance over existing solutions.

Key words: access control, abnormal configurations, spectral clustering

1 引言

为了保护隐私信息不会被非法用户访问，需要对信息的访问行为进行有效控制。访问控制机制实现了对资源访问的有效监控，保障被授权用户在合法的条件下获得有效的访问权限，防止资源被非授权用户访问，阻止信息非授权流动。针对不同应用环境，研究人员相继提出强制访问控制（MAC, mandatory access control）、自主访问控制（DAC, discretionary access control）、基于角色的访问控制[1,2]（RBAC, role-based access control）、面向网络空间的访问控制[3]（CoAC, cyberspace-oriented access control）等访问控制模型。其中，RBAC 模型因其具有高效、可扩展性强等特性得到了工业界及学术界的广泛关注。相比 MAC、DAC 等模型用户权限一一手动设定的方式，RBAC 模型通过引入角色的

收稿日期：2017-03-27；**修回日期**：2017-11-06

基金项目：国家重点研发计划基金资助项目（No.2016YFB0801001）；国家自然科学基金资助项目（No.61672515）；东莞市引进创新科研团队计划基金资助项目（No.201636000100038）

Foundation Items: The National Key Research and Development Program of China (No.2016YFB0801001), The National Natural Science Foundation of China (No.61672515), Dongguan Innovative Research Team Program (No.201636000100038)

概念将权限设定的过程进行了简化。RBAC 模型将权限映射到相应角色之后将角色赋予不同的用户，对用户访问和操作资源的行为进行抽象。当此类权限需要修改时，仅需将赋予对应角色的权限进行更新即可实现对应用户权限的自动更新。RBAC 帮助简化认证管理并增强安全性政策执行。同时，RBAC 已被证明可以很好地解决数据库、云等环境下的授权管理问题[4,5]。因此，将非 RBAC 系统（如 MAC、DAC 等）迁移为 RBAC 系统可以有效提高系统的管理效率以及安全性。

在迁移过程中，最重要的是如何生成准确的角色。如果生成的角色中包含错误的权限—角色分配关系，将导致错误的权限被赋予相关用户，带来严重的系统安全隐患。针对角色生成问题，研究者提出了一系列的角色挖掘方案[6~8]。角色挖掘旨在将初始用户—权限分配（UPA, user-permission assignment）矩阵分解为用户—角色分配（UA, user-role assignment）矩阵和角色—权限分配（PA, permission- role assignment）矩阵。这些算法大多假设原始 UPA 矩阵中不包含异常权限配置，但在实际应用中该假设并不成立。根据达特茅斯大学的一个研究小组对 4 个大型金融机构中的员工权限的调研结果显示，其中，超过 50%的员工的权限存在异常[9]。这些异常权限配置增加了敏感信息泄露的概率。但如何判别并处理这些异常权限配置尚未得到广泛研究。

为了解决上述异常权限配置发现问题，本文提出了一种基于谱聚类的异常权限配置挖掘框架。该框架可有效发现 RBAC 系统中潜在的异常权限配置，同时提供适当的修正方案。本文的主要贡献如下。

针对 RBAC 系统中的精准用户聚类问题，本文提出一种自适应谱聚类算法。其中，针对访问控制权限表示方式的特点，设计了一种混合距离度量函数。通过与传统的距离度量函数相比，本文所提出的距离度量函数可有效提高聚类结果的精度。此外，针对传统谱聚类算法参数计算主观性较强所带来的聚类结果误差问题，提出一种自适应参数计算方法。实验结果证明本文所提出的自适应参数计算方法在有效减少主观性的基础上实现了较为精确的聚类。

本文给出一系列异常权限配置挖掘规则，在聚类结果上，利用该规则可有效发现并处理异常权限配置。实验结果证明，同已有方案相比，本文所提方案可最大限度地实现异常权限配置挖掘。

2 相关工作

现有的角色挖掘研究可分为基于聚类方法的角色挖掘和基于二进制矩阵分解的角色挖掘等 2 种。本节对以上 2 种角色挖掘框架进行简要介绍。

2.1 基于聚类方法的角色挖掘

Schlegelmilch 等[10]提出了一种基于层次聚类的角色挖掘方案，并结合 ORCA 工具实现了角色的可视化展示。但该方案要求生成的角色中权限不能重叠，即分配给某一角色的权限不能分配给其他角色，这与实际应用需求相违背。针对这一问题，基于枚举及排序的思想，Vaidya 等[11]提出一种候选角色生成排序算法。该算法通过枚举用户权限之间的交集，之后，对生成的新集合求交集，直到枚举出了所有可能的集合。最后，对所枚举出的集合按照其包含用户的数量进行排序。这种方式的时间复杂度随着求交前角色数量呈指数级增长。为此，Vaidya 等[11]提出一种 Fast Miner 算法。该算法通过将枚举过程限制为仅枚举两两初始化角色之间的交集的方式达到降低时间复杂度的目的。但上述方案不能完全反映 RBAC 系统的语义角色。为了优化角色语义并降低角色挖掘框架的复杂度，Molly 等[12]提出了一种基于加权结构复杂度的角色挖掘框架。但是该方案允许管理员直接给用户分配相应权限，违背了 RBAC 将用户与权限进行隔离的设计初衷。

2.2 基于二进制矩阵分解的角色挖掘方法

Vaidya 等[6]证明 RMP 是 NPC 问题，并将 RMP 角色挖掘映射到 Minimum Tilling 问题和 Discrete Basis 问题上。Frank 等[13]在 RBAC 模型的基础上引入了业务角色（business role）和技术角色（technical role）的概念，并通过这 2 个角色进一步细化了系统中不同业务间的相关性。此外，为了使挖掘出的角色可以更准确地反映访问控制系统的各类约束（如最小特权和角色使用基数等），研究者提出了基于约束的角色挖掘框架[14,15]。

但以上方案均假设原始 UPA 不包含异常权限配置，这使生成的角色可能包含错误的权限配置。为在生成角色时消除异常权限配置， Molly 等[7]将奇异值分解、非负矩阵分解等矩阵分解算法引入角色挖掘中，并证明以上矩阵分解算法可一定程度识别系统中的异常权限配置。此外，Bauer 等[8]提出一个基于关联规则的角色生成方案，该方案通过分析授权的频繁模式挖掘其中的异常权限配置。但这些

方案需要人工设定挖掘时所需的参数，算法性能一定程度取决于管理员的相关背景知识。同时，以上方法所生成的角色仅是权限间的简单排列组合，缺少相应的角色语义信息。因此，如何实现自适应的异常权限配置挖掘同时增加所挖掘出的角色语义信息是亟待解决的问题。

3　问题描述与预备知识

3.1　研究背景

将 MAC、DAC 等访问控制系统迁移为 RBAC 系统可显著提高访问控制权限的管理效率、灵活性以及系统安全性。为了准确反映原访问控制系统的功能同时保证新系统的安全性，需要生成的角色中的权限分配尽可能准确。然而，已有角色挖掘方案在设计时未考虑原始系统中存在的异常权限配置，这些错误的权限一旦分配给非法用户，将导致非法用户浏览、修改或删除某些隐私信息（如身份信息、家庭地址等），给系统带来安全隐患。如果在生成角色时忽略这些异常权限，这些权限配置关系可能被带入最终角色中，从而影响所有与此角色相关的用户。为解决这一问题，本文提出以下解决方案：首先，根据用户具有权限的相似性对用户聚类，之后，通过分析聚类结果对其中可能存在的异常权限配置进行挖掘。

3.2　系统框架

如图 1 所示，本文提出的异常权限配置挖掘框架主要包含以下 3 个阶段：原始数据预处理阶段、用户聚类阶段和异常权限配置挖掘阶段。

1) 原始数据预处理阶段。RBAC 系统中用户权限以布尔矩阵的形式进行描述，矩阵中不同的列表示不同的权限，行表示单个用户的权限集合。矩阵中“1”表示此权限被授予该用户，而“0”表示此权限未被授予该用户。由于系统中的大量用户具有相同的权限，如不对这些重复用户进行合并，会增加用户相似性计算的计算量，导致计算资源的浪费。因此，该阶段中对原始 UPA 矩阵中具有相同权限的用户进行合并，减小计算用户相似度的计算量。但在构建用户亲和度矩阵时考虑全部用户，因此，不会造成频繁出现的权限集与一个孤立的权限集相等的情况，对聚类结果无影响。

2) 用户聚类阶段。以去除重复用户后的 UPA 为基础，对其中具有“相似”权限的用户进行聚类。由于传统聚类方法所使用的距离度量函数（DMF, distance measurement function）（如欧式距离、曼哈顿距离等）大多仅对非布尔数据敏感，而 RBAC 系统多采用布尔向量的形式对用户进行描述。因此，针对 RBAC 系统中权限表示方式的特点，本文设计一种新的聚类度量方法，提高了聚类结果的准确性。此外，对于聚类算法中参数的计算，为了减小主观因素对聚类结果的影响，本文提出一种参数自适应的谱聚类用户聚类算法。该算法在聚类时不需要提前设定参数，可实现自动聚类。

3) 异常权限配置挖掘阶段。本文提出一套异常权限挖掘规则，并在聚类结果的基础上使用该规则对访问控制系统中可能存在的异常权限配置进行挖掘，并给出相应的修正意见。通过多轮迭代，当无异常权限配置被挖掘出时异常挖掘框架停止。

3.3　预备知识

谱聚类将聚类问题转化为图的最优划分问题，是一种点对聚类算法，对数据聚类具有很好的效果[16]。

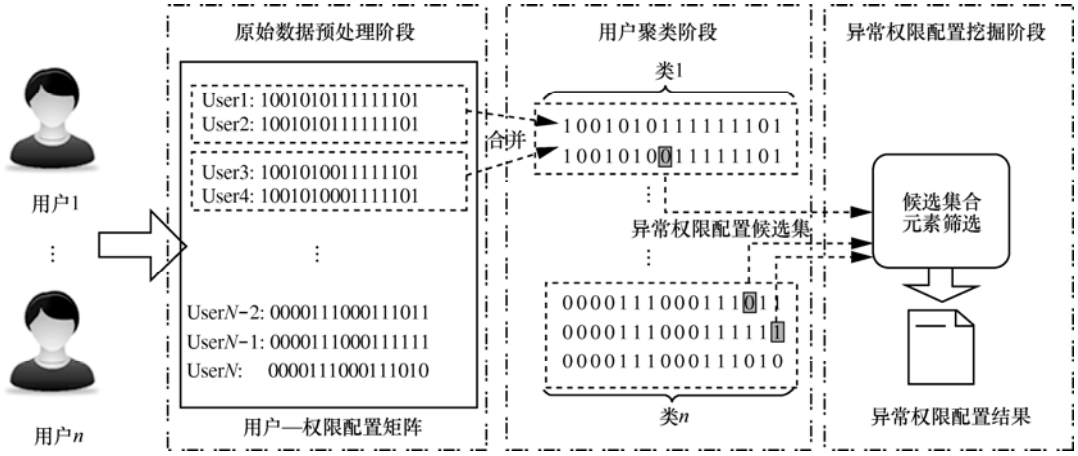


图 1　系统框架

与传统的聚类算法相比，谱聚类具有在任意形状的样本空间上聚类且收敛于全局最优解的优点。具体来说，给定一组数据

$$X=\{x_1,x_2,\mathrm{L}\ ,x_n\}$$

谱聚类算法的流程如下。

Step1 构建相似度图$G=(V,E)$，其中，顶点v_i表示用户x_i，边e_{ij}表示用户x_i和x_j之间的相似度$s_{ij}(s_{ij}\geqslant 0)$。通常谱聚类算法需要构造用户全连通图，并计算全部用户间的相似性，构造用户亲和度矩阵。亲和度矩阵定义为

$$\boldsymbol{A}_{ij}=\exp\left(\frac{-d^2(x_i,x_j)}{\sigma^2}\right),\ i\neq j,A_{ii}=1$$

其中，x_i,x_j表示不同用户，$d(x_i,x_j)$表示用户距离，σ表示局部比例系数。

Step2 计算亲和度矩阵的前k个最大的特征值，将对应的特征向量构建新的数据特征空间。

Step3 利用*K*-means、fuzzy C-means(FCM)等*K*-way划分算法对新数据特征空间中的点进行聚类，之后，将聚类结果映射回原数据空间。

但现有谱聚类方案不适用于访问控制系统中用户的聚类，具体原因有以下几点。

1) 现有的DMF（如文献[16~18]中所使用方案）和矩阵分解（如文献[19,20] 中所使用方案）虽已被证明在非布尔形式的数据上可以获得较好的聚类效果，但由于RBAC系统中权限的表示使用布尔矩阵的形式，导致直接使用现有DMF可能会丢失数据的一些重要特性，无法准确度量RBAC系统中用户的相似度。因此，需要设计合适的DMF从而实现不同用户的精确地区分。

2) 作为影响聚类结果的重要参数，局部比例系数（LSP, local scaling parameter）的选择会影响相同簇内元素亲和力的计算，而亲和力的计算对聚类精度有着直接的影响。与此同时，聚类个数（CN, cluster number）的选择也决定了聚类结果的精度。然而，传统谱聚类算法中LSP和CN通常需要手动设置，导致聚类结果一定程度上取决于系统管理员的经验。如何消除这些主观因素，设计一种自适应的参数计算算法也是亟待解决的问题。

4 用户聚类算法

为了提供精确的用户聚类结果，本文提出一种改进的参数自适应的谱聚类算法，具体构造方案如下。

4.1 距离函数构造

汉明距离（Hamming distance）广泛应用于度量具有相同长度的数据间对应位不同的数量。对于集合$X=\{x_1,x_2,\mathrm{L}\ ,x_n\}$和$Y=\{y_1,y_2,\mathrm{L}\ ,y_n\}$，其汉明距离定义为

$$\begin{aligned}H(X,Y)&=\left|X\ \mathrm{U}\ Y\right|-\left|X\ \mathrm{I}\ Y\right|\\&=\sum_{i=1}^{n}x_i+\sum_{i=1}^{n}y_i-2\sum_{i=1}^{n}\min\{x_i,y_i\}\end{aligned}\qquad(1)$$

其中，$x_i,y_i\in\{0,1\}$。其他DMF（如欧式距离、曼哈顿距离等）虽然也可度量等长向量间的相似度，但当向量中数据为布尔形式时，这些方案无法准确反映不同向量间的区别或等价于汉明距离。例如，存在以下用户Alice = {11110000}和Bob = {11111111}（其中，“1”表示用户具有该权限，“0”表示用户不具有该权限）。2个人的汉明距离为4，而欧氏距离和曼哈顿距离分别为2和4。可以看出，欧式距离的结果缩小了不同用户间的差异，而曼哈顿距离在计算布尔向量时等价于汉明距离。

但单独使用汉明距离不能完整反映用户之间的差异。即使汉明距离相等，不同用户也可能存在差异。例如，对于$comparison_1$和$comparison_2$中的用户，其汉明距离均为5。但同$comparison_2$相比，$comparison_1$中用户间的不同权限占全部权限的比例更小。这意味着相比于$comparison_2$中的用户，$comparison_1$中的用户具有更高的相似度。

$$comparison_1=\begin{cases}1111111111111111\\1111100000011111\end{cases}$$

$$comparison_2=\begin{cases}1100000100001111\\0000000100000000\end{cases}$$

为了更准确地对不同用户进行区分，需要考虑用户权限差异比例的不同。Jaccard系数被广泛应用于度量2个向量中相同元素占全部元素的比例。Jaccard系数为

$$Jaccard(X,Y)=\frac{\left|X\ \mathrm{I}\ Y\right|}{\left|X\ \mathrm{U}\ Y\right|}=\frac{\sum_{i=1}^{n}\min\{x_i,y_i\}}{\sum_{i=1}^{n}\max\{x_i,y_i\}}\qquad(2)$$

定义了2个用户所有权限中相同权限的比例，因此，不同权限的比例定义为

$$Jaccard_D(X,Y)=1-Jaccard(X,Y) \tag{3}$$

其中，$x_i,y_i\in\{0,1\}$ 中。与汉明距离相同，单纯使用 Jaccard 距离也无法准确地反映用户之间的相似度。例如，对于 $comparison_3$ 和 $comparison_4$ 中的用户，其 Jaccard 距离均为 0.5。但相比于 $comparison_4$ 中用户，$comparison_3$ 中用户具有更多不一致的权限，相似度较低。

$$comparison_3=\begin{cases}11111111\\11110000\end{cases}$$

$$comparison_4=\begin{cases}11000000\\01000000\end{cases}$$

因此，将 Jaccard 距离和汉明距离相结合，可以同时反映不同权限的个数和不同权占全部权限的比例，使距离度量函数可以同时反映多个维度的信息，增加了距离度量函数的准确性。因此，将式(1)与式(3)进行组合，本文所提出的 DMF 定义为

$$\begin{aligned}d(X,Y)&=H(H,Y)Jaccard_D(X,Y)\\&=\frac{\left[\sum_{i=1}^{n}x_i+\sum_{i=1}^{n}y_i-2\sum_{i=1}^{n}\min\{x_i,y_i\}\right]^2}{\sum_{i=1}^{n}x_i+\sum_{i=1}^{n}y_i-\sum_{i=1}^{n}\min\{x_i,y_i\}}\end{aligned} \tag{4}$$

4.2　局部比例系数计算

作为谱聚类算法的重要参数，用户间距离相等时，LSP 越大，则用户相似度越低，用户属于同一类的概率越小，用户间的可区分度越高。因此，LSP 的选择对谱聚类的结果有很大的影响。传统算法大多需要预先人为设定 LSP 或相关参数，例如，文献[17]中首先计算用户 x_i 同其他用户的相似度并按由大到小的顺序排序，之后选择排序为 7 的相似度作为最终的 LSP。但通过实验验证，该方法无法有效对本文所用数据集中的用户进行区分。可以看出，这些方式较为依赖参数设定者的专业知识。因此，为消除人为因素对聚类结果的影响，提出一种自适应的 LSP 计算方法。

$$\sigma_i=\left|\overline{x}-\sqrt{\frac{\sum_{k=1}^{n}\left|x_k-\overline{x}\right|^2}{n}}\right| \tag{5}$$

其中，n 代表与用户 x_i 的距离不为∞的用户的个数，$\overline{x}$ 表示这 n 个用户同 x_i 之间的平均距离，$\sqrt{\frac{\sum_{k=1}^{n}\left|x_k-\overline{x}\right|^2}{n}}$ 表示 x_i 同其他用户距离的标准差，标准差越小，相应的 σ_i 越大，用户间的可区分度越高。经实验结果证明，本文所提出的自适应 LSP 计算方法在不同数据集上均可取得较好的结果。

4.3　聚类个数计算

传统的聚类方法预先手动设置聚类个数的方式使聚类结果具有一定的主观性。针对这一问题，本文采用基于本征间隙的聚类个数计算方法实现聚类个数的自动选择[21]。

k 个理想的彼此分离的类在亲和度矩阵中的表现为该亲和度矩阵对角线上分布 k 个全 1 分块矩阵，其余位置均为 0。但实际应用中，很难获得这样的亲和度矩阵。实际的亲和度矩阵中对角线上的分块矩阵元素不全为 1，对角线分块矩阵外的元素也不全为 0。因此，亲和度矩阵可看作是加入扰动的理想 0-1 矩阵。根据矩阵摄动理论，此时亲和度矩阵前 k 个特征解与理想亲和度矩阵的 k 重最大特征值所对应的特征解非常接近。因此如果聚类个数为 k，将特征值按从大到小排列，则前 k 个特征值接近于 1，而第 $k+1$ 个特征值会明显变小。这样就在第 $k+1$ 个和第 k 个特征值之间产生一个大的落差，该落差就叫做本征间隙（eigengap）。因此，需要首先计算规范化亲和度矩阵的特征值并按顺序从大到小排列为

$$\{\lambda_1,\lambda_2,\mathrm{L}\ ,\lambda_n\}$$

则本征间隙序列可表示为

$$\{g_1,g_2,\mathrm{L}\ ,g_{n-1}\mid g_i=\lambda_{i+1}-\lambda_i\}$$

本征间隙值越大，相应特征向量所构成的子空间越稳定。因此，第一个极大本征间隙出现位置确定为聚类个数。

本文所提谱聚类算法（如算法 1 所示）的主要时间是耗费在亲和度矩阵构造、特征值计算以及聚类阶段。设系统中共有 n 个用户，在消除重复用户后具有 m 个用户，则共需要比较 $\frac{m^2}{2}$ 次，因此，亲和度矩阵构造的时间复杂度为 $O(m^2)$。而特征值计算及聚类算法的时间复杂度则主要采用已有算法，具体时间复杂度取决于所采用算法的复杂度。本文实验阶段的特征值计算算法复杂度为 $O(n^3)$。最终的聚类算法采用 *K*-means 进行聚

类，时间复杂度为 $O(tkn)$，其中，t 为循环计算的次数，k 为聚类个数。由于 t,k 相比于 n 来说都很小，因此，*K*-means 算法的复杂度近似等于 $O(n)$。

算法 1 基于谱聚类的用户聚类算法

输入 用户—权限分配矩阵: ***UPA***

输出 聚类结果: *Clusters*

1) 按行遍历 ***UPA***，合并其中的相同用户，构建无重复用户矩阵 ***UPA_FILTERED***

2) 获取 ***UPA_FILTERED*** 的行数： r

3) 初始化亲和度矩阵 $\boldsymbol{A}$ = Ones((*r*, *r*))

4) for x, y in ***UPA_FILTERED*** do

5) 计算汉明距离 $Hamming(x, y)$

6) 计算 Jaccard 距离 $Jaccard_D(x, y)$

7) x, y 间的距离为

$d(x, y) = Hamming(x, y)Jaccard_D(x, y)$

8) $A(x, y) = d(x, y)$

9) for $i \leqslant r$ do

10) $$\overline{x} = \frac{sum(A[i:])}{n}$$

11) 计算局部比例系数为

$$\sigma_i = \left| \overline{x} - \sqrt{\frac{\sum_{k=1}^{n} \left| x_k - \overline{x} \right|^2}{n}} \right|$$

12) for x, y in $\boldsymbol{A}$ do

13) $$A(x, y) = \exp\left(\frac{-d^2(x, y)}{\sigma_x \sigma_y} \right)$$

14) 构造 Laplacian 矩阵， $\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{A} \boldsymbol{D}^{-\frac{1}{2}}$，其中， $\boldsymbol{I}$ 为单位矩阵， $\boldsymbol{D}$ 为对角矩阵，

$$D_{ii} = \sum_{j=1}^{r} A(i, j)$$

15) 计算 L 的特征值 $\{\lambda_1, \lambda_2, \mathrm{L}\ , \lambda_n\}$ 及对应的特征向量，并对特征值进行排序，计算本征间隙位置，确定聚类个数 $n\_cluster$

16) 把 $n\_cluster$ 个特征(列)向量排列在一起组成 $r \cdot n\_cluster$ 的矩阵，之后使用 *K*-means 算法进行聚类。

5 异常权限配置预测

5.1 异常权限配置定义

UPA 中的异常权限配置定义如下。

定义 1 正异常权限配置(positive abnormal configuration)。若某用户类中被授予某权限的用户占类中全部用户的比例小于阈值 τ_p 时，对应的用户—权限配置为正异常权限配置。

如表 1 所示，集合中除用户 B 外其余用户均未被授予权限 P5。若 $\tau_p = 0.25$，则该权限配置为正异常权限配置。这类异常权限配置会导致非法用户访问受保护的资源从而造成信息的泄露。

表 1 正异常权限配置实例

用户	P1	P2	P3	P4	P5	P6	P7	P8	P9
A	1	1	1	1	0	0	0	0	0
B	1	1	1	1	1	0	0	0	0
C	1	1	1	1	0	0	0	0	0
D	1	1	1	1	0	0	0	0	0
E	1	1	1	1	0	0	0	0	0

定义 2 负异常权限配置(negative abnormal configuration)。若某一用户类中，未被授予某权限的用户占类中全部用户的比例小于阈值 τ_n 时，对应的用户—权限配置为负异常权限配置。

如表 2 所示，该集合中除用户 C 外，其余用户均被授予权限 P6，若 $\tau_n = 0.25$，则该权限配置可标记为负异常权限配置。

表 2 负异常权限配置实例

用户	P1	P2	P3	P4	P5	P6	P7	P8	P9
A	1	1	1	1	0	1	0	0	0
B	1	1	1	1	0	1	0	0	0
C	1	1	1	1	0	0	0	0	0
D	1	1	1	1	0	1	0	0	0
E	1	1	1	1	0	1	0	0	0

5.2 异常权限配置检测流程

Step1 聚类。采用第 4 节的方法对 ***UPA*** 进行聚类，获取相似用户集合。

Step2 聚类结果预处理。在对异常权限配置进行检测前，首先，对聚类结果进行预处理。对于每个类簇，构建其特征模式向量。具体构建方式如下。对于特征模式向量中的元素，如果类中对应位置的列向量元素均为 1，那么将特征向量中该位置置为 1，其余位置全部置为 0。

Step3 异常权限配置规则匹配。该阶段中，根据预先设定好的异常权限配置挖掘规则筛选异常权限配置候选集，具体的规则如下。

规则 1　如果某权限仅被赋予一个类中的用户，同时被授予该权限的用户占类中用户的比例小于阈值 τ_p，则将此配置标记为正确配置。

规则 2　如果某权限被赋予多个类中的用户，若当前类中被授予该权限的用户占其全部用户的比例小于阈值 τ_p，那么除该类外，若其余包含该权限的类的特征模式向量的交集。

1) 为当前类特征模式向量的子集，则定义该权限配置为正确配置，不做处理。

2) 不为当前类特征模式向量的子集，则定义该权限配置为正异常权限配置，将该权限配置加入异常权限配置候选集，并在权限配置矩阵中将对应位置置为 0。

规则 3 如果某权限未被赋予类中用户的比例占类中全部用户的比例小于阈值 τ_n 时，则定义这类配置为负异常权限配置，将这些权限配置加入异常权限配置候选集，并在权限配置矩阵中将对应的位置置为 1。

Step4　交叉聚类。在依据用户—权限矩阵进行聚类并挖掘异常权限配置后，将该矩阵行列进行互换，得到该矩阵的转置矩阵。之后根据本文所提出的聚类算法进行交叉聚类。在得到聚类结果后，根据 Step 3 中所提规则构造异常权限配置候选集。

Step5　异常权限配置决策。根据以上步骤可以获取 2 个异常权限配置候选集合，对 2 个集合进行交运算，得到公共集合。定义该公共集合内的元素为最终的异常权限配置，并根据相应的修改原则对相应的元素进行更新。

Step6 在对全部聚类结果进行处理后，可以得到一个新的用户—权限矩阵，之后将该矩阵作为本文所提出异常权限配置挖掘框架的输入并迭代执行，直到无异常权限配置被检测出或交叉聚类的异常权限配置候选集合交集为空时算法停止。

6　实验分析

本节主要展示本文所提异常权限配置检测算法与其他相关算法结果的对比，从而验证本文所提方案的有效性。 首先，介绍实验所采用的数据集和评价方法，然后，设计了多组实验从不同的角度对比分析了本文算法的性能。

6.1　数据集介绍

为了验证本文所提方案的有效性，本节采用与文献[22~24]中相同的数据集对聚类结果及异常权限配置挖掘结果进行验证。实验数据集介绍如表 3 所示。

表 3　　**实验数据集介绍**

数据集	用户个数	权限个数
Healthcare	46	46
University	493	56
Emea	35	3 046
Firewall1	365	1 417
Firewall2	325	1 179

6.2　聚类结果评估

为验证本文所提 DMF 的有效性，将其同欧式距离、汉明距离、余弦距离和 Jaccard 距离进行对比。如图 2 所示，相比于其他距离函数，本文所提出的 DMF 生成的聚类个数更多，可对用户进行更为有效的区分。同时，如图 3 所示，虽然本文提出的聚类算法增加了聚类个数，但最终生成的角色数量无明显增加。这说明传统方式下的聚类结果不够精确，一个类中可能会发掘出多个角色。

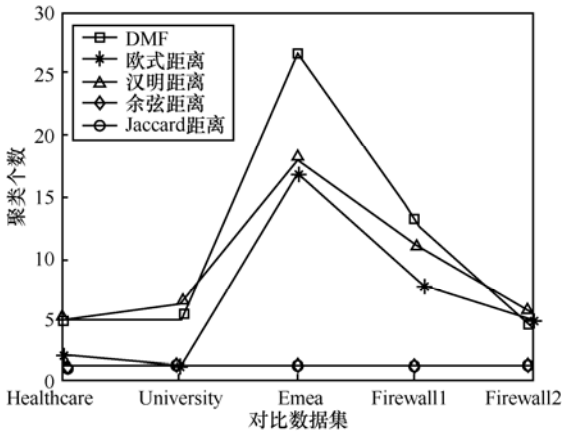


图 2　聚类个数对比

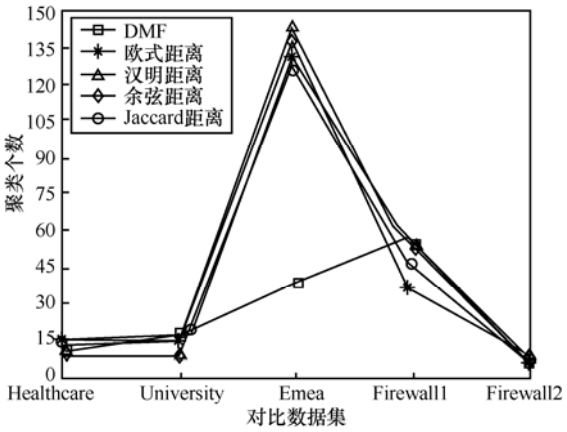


图 3　角色个数对比

本文所提方案虽然聚类个数更多，但是单个类中的角色个数更少，因此，在角色数并未明显增加的情况下，本文所得到的角色可以更准确地反映不同角色之间的差异。

如果聚类方法有效，则类内用户的差异应较小，类间用户差异应较大。因此，通过对比不同算法所得聚类结果的类内用户相似度以及类间用户相似度可对本文所提聚类算法结果进行验证。类内相似度定义为 *Inner _ Similarity*，*Inner _ Similarity* 代表了类内用户两两之间的平均相似度，如果分类较为准确，则类内用户两两之间相似度应较高，因此，两两之间相似度的平均值也应该较高。

$$Inner\_Similarity=\frac{\sum_{i=1}^{n}\sum_{j=n+1}^{n}d(x_i,x_j)}{\frac{n(n+1)}{2}} \tag{6}$$

其中，x_i,x_j 表示同一集群中的不同用户，$d(x_i,x_j)$ 表示 x_i,x_j 之间的距离，n 是类内的用户数。

类间相似度定义为 *Between _ Similarity*，*Between _ Similarity* 代表类间用户两两之间的平均相似度。如果分类较为准确，类间用户两两之间的相似度应较低，因此，两两之间相似度的平均值也较低。

$$Between\_Similarity=\frac{\sum_{i=1}^{m}\sum_{j=1}^{n}d(x_i,y_j)}{mn} \tag{7}$$

其中，x_i,y_j 分别表示不同集群中的用户，$d(x_i,y_j)$ 表示 x_i,y_j 的距离，m,n 表示不同类内的用户数。

虽然通过计算方差的方法也可有效反映类内用户间相似度的分布。但如果类内两两用户间的相似度均较低，也会出现方差较小的情况。因此，使用平均相似度进行结果评判较为合理。

图 4、图 5 展示了本文所提出的距离度量函数与其他 4 种传统的距离度量函数所得到的聚类结果的类内相似度与类间相似度的对比结果。由图 4 可知，采用本文所提出距离度量函数所得到的聚类结果类内相似度可达 90%，相比于其他方案具有更高的类内相似度。由图 5 可知，本文所提聚类算法所得结果的类间相似度仅为 3.4%，聚类结果具有更小的类间相似度。以上结果证明采用本文所提出的距离函数所得到的聚类结果可以更有效的对不同用户进行区分。

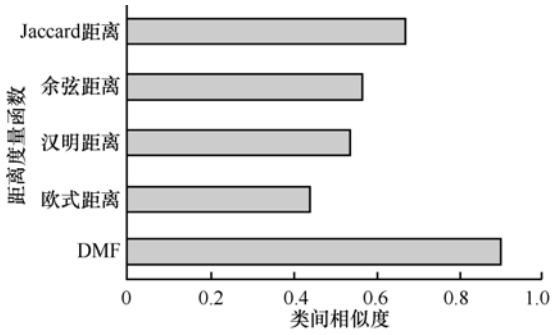


图 4　类内相似度对比

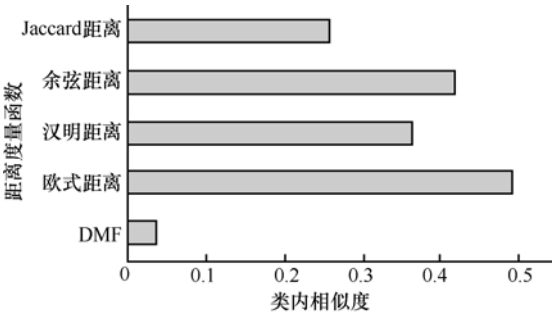


图 5　类间相似度对比

6.3　预测结果准确率

为了评估本文所提出异常权限配置挖掘框架的性能，将其同文献[7,24]中的算法进行对比。由于无法准确获取原始 ***UPA*** 中的异常权限配置信息，因此将原始 ***UPA*** 看作是无噪声数据。通过随机修改原始 ***UPA*** 中数据的方式向其异常权限配置中添加异常权限配置，之后，通过本文所提出的异常权限配置挖掘框架进行异常权限配置检测。同文献[24]类似，由于负异常权限配置仅会给用户执行相应的操作带来影响，而不会增加系统中信息泄露的风险，而正异常权限配置则赋予用户过多的权限，极大增加了信息泄露的概率。因此，在向原始 ***UPA*** 中添加异常权限配置信息时，应添加更多的正异常权限配置。同时，文献[7]的分析结果说明大多系统中异常权限范围在 5%~15%，因此，本文实验方案中选取对应阈值 $\tau_p=\tau_n=0.15$。

以 University 数据集为例，同文献[7,24]相同，向其中添加 197 个正异常权限配置，23 个负异常权限配置，分别记为 P_{add} 和 N_{add}。在进行异常权限配置挖掘后，通过与原始 UPA 对比，定义未发掘出的异常权限配置为 P_{left} 和 N_{left}。如表 4 所示，相

比文献[7]与文献[24]中所提出的算法，本文所提异常权限配置检测算法发现 90%以上的异常权限配置，正异常权限配置挖掘准确率可达 96%。证明本文所提方案在最大限度发现系统中的异常权限配置的同时也最大程度地消除了正异常权限配置，减小信息泄露的概率。

表 4　　异常权限配置检测结果对比

数据集	初始异常配置数		对比算法	最终异常配置数	
	P_{add}	N_{add}		P_{left}	N_{left}
University	197	23	文献[7]算法（SVD）	35	18
			文献[7]算法（NMF）	28	30
			文献[7]算法（BNMF）	30	16
			文献[7]算法（LPCA）	20	22
			文献[24]算法（PSRM）	16	14
			本文算法	6	14

此外，为了验证本文所提方案的稳定性，对以上实验重复 50 次，每次随机向 ***UPA*** 中添加 190 个正异常权限配置和 30 个负异常权限配置，实验结果如图 6 所示。可以看出本文所提出的异常权限配置检测算法具有较好的稳定性。由图 7(a)和图 7(b)可知，相比于 PSRM 算法[24]，本文所提出的异常权限挖掘框架可在挖掘更多的正异常配置同时保证负异常配置的挖掘精度。

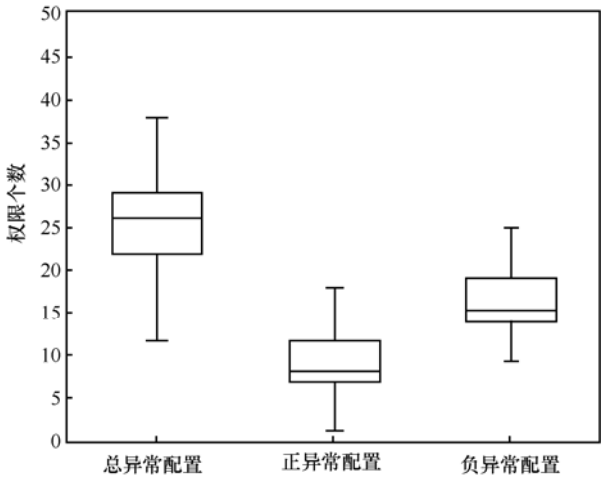
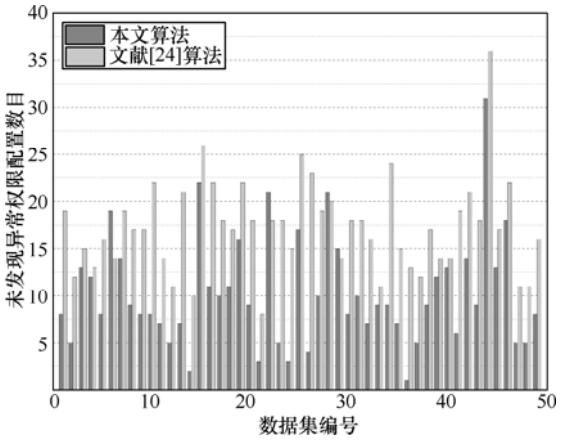


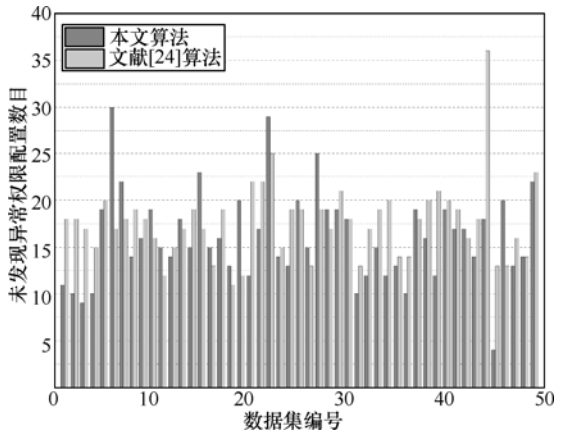
图 6　算法稳定性

7　结束语

为了实现精准的用户聚类，本文首先提出了一种权限敏感的距离度量函数及一种自适应的聚类参数计算方案。并结合该度量函数设计了一种面向 RBAC 系统的参数自适应谱聚类算法。



(a) 正异常权限配置对比



(b) 负异常权限配置对比

图 7　剩余异常权限配置对比

相比传统的谱聚类算法，在减小主观因素对聚类结果影响的基础上实现了较好的聚类效果。

在此基础上，本文提出了一套异常权限配置挖掘规则。实验结果证明，相比其他异常配置挖掘方案，本文所提方案可最大限度地对访问控制系统中的异常权限配置给予正确修订。

参考文献：

[1] ZHANG D, RAMAMOHANARAO K, EBRINGER T, et al. Permission set mining: discovering practical and useful roles[C]//The 24th Annual Computer Security Applications Conference. 2008: 247-256.

[2] YOUNIS Y A, KIFAYAT K, MERABTI M. An access control model for cloud computing[J]. Journal of Information Security and Applications, 2014, 19(1): 45-60.

[3] 李凤华, 王彦超, 殷丽华, 等. 面向网络空间的访问控制模型[J]. 通信学报, 2016, 37(5): 9-20.
LI F H, WANG Y C, YIN L H, et al. Novel cyberspace-oriented access control model[J]. Journal on Communications, 2016, 37(5): 9-20.

[4] YU X, XU P, ZHANG T, et al. Research and implementation of role-based access control model of fundamental spatial database

system of Jilin water resources[C]//The 2013 International Conference on Information System and Engineering Management.2013:83-86.

[5] WANG Y C, LI F H, XIONG J B, et al. Achieving lightweight and secure access control in multi-authority cloud[C]//The 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications. 2015:459-466.

[6] VAIDYA J, ATLURI V, GUO Q. The role mining problem: a formal perspective[J]. ACM Transactions on Information and System Security, 2010, 13(3).

[7] MOLLOY I, LI N.H, QI A, et al. Mining roles with noisy data[C]//The 15th ACM Symposium on Access Control Models and Technologies. 2010:45-54.

[8] BAUER L, GARRISS S, REITER M. K. Detecting and resolving policy misconfigurations in access-control systems[J]. ACM Transactions on Information and System Security, 2011, 14(1).

[9] DAS T, BHAGWAN R, NALDURG P. Baaz: a system for detecting access control misconfigurations[C]//The 19th USENIX Security Symposium. 2010: 161-176.

[10] SCHLEGELMILCH J, STEFFENS U. Role mining with Oracle[C]// The 10th ACM Symposium on Access Control Models and Technologies. 2005: 168-176.

[11] VAIDYA J, ATLURI V, WARNER J. Roleminer: mining roles using subset enumeration[C]//The 13th ACM Conference on Computer and Communications Security. 2006: 144-153.

[12] MOLLOY I, CHEN H, LI T et al. Mining roles with semantic meanings[C]//The 13th ACM Symposium on Access Control Models and Technologies.2008: 21-30.

[13] FRANK M, BUHMAN J M, BASIN D. Role mining with probabilistic model[J]. ACM Transactions on Information and System Security, 2013, 15(4).

[14] HARIKA P, NAGAJYOTHI M, JOHN J C, et al, Meeting cardinality constraints in role mining[J]. IEEE Transactions on Dependable and Secure Computing, 2015, 12(1):71-84.

[15] JAFARIAN J H, TAKABI H, TOUATI H, et al. Towards a general framework for optimal role mining: a constraint satisfaction approach[C]//The 20th ACM Symposium on Access Control Models and Technologies.2015:211-220.

[16] LUXBURG U V. A tutorial on spectral clustering[J]. Statistics and Computing, 2007, 17(4): 395-416.

[17] ZELNIK-MANOR L, Self-tuning spectral clustering[J].Advances in Neural Information Processing Systems, 2004, 14:1601-1608.

[18] YAN J, CHENG D, ZONG M, et al. Improved spectral clustering algorithm based on similarity measure[C]//The 10th International Conference on Advanced Data Mining and Applications.2014: 641-654.

[19] GHOSHDASTIDAR D, DUKKIPATI A.Spectral clustering using multilinear svd: Analysis, approximations and applications[C]//The 29th Conference on Artificial Intelligence.2015:2610-2616.

[20] LU H, FU Z, SHU X, Non-negative and sparse spectral clustering [J]. Pattern Recognition, 2014, 47(1): 418-426.

[21] 孔万增, 孙志海, 杨灿,等. 基于本征间隙与正交特征向量的自动谱聚类[J]. 电子学报，2010, 38(8): 1880-1885.

KONG W Z, SUN Z H, YANG C, et al. Automatic spectral clustering based on eigengap and orthogonal eigenvector[J]. Acta Electronica Sinica, 2010, 38(8):1880-1885.

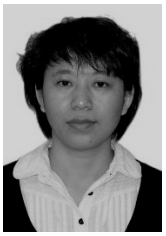
[22] STOLLER S D, YANG P, RAMAKRISHNAN C R, et al. Efficient policy analysis for administrative role based access control[C]//The 2007 ACM Conference on Computer and Communications Security. 2007: 445-455.

[23] ENE A, HORNE W, MILOSAVLJEVIC N, et al. Fast exact and heuristic methods for role minimization problems[C]//The 13th ACM Symposium on Access Control Models and Technologies.2008: 1-10.

[24] YIN L, FANG L, NIU B, et al. Hunting abnormal configurations for permission-sensitive role mining[C]//The 2016 IEEE Military Communications Conference. 2016: 1004-1009.

作者简介：

房梁（1989-），男，山西太原人，北京邮电大学博士生，主要研究方向为信息安全、访问控制。



殷丽华（1973-），女，辽宁朝阳人，博士，广州大学教授、博士生导师，主要研究方向为信息安全、安全性评估。

李凤华（1966-），男，湖北浠水人，博士，中国科学院信息工程研究所副总工、研究员、博士生导师，主要研究方向为网络与系统安全、信息保护、隐私计算。

方滨兴（1960-），男，江西万年人，中国工程院院士、广州大学教授，主要研究方向为计算机体系结构、计算机网络与信息安全。