

文章编号:1006-3080(2016)06-0858-05

DOI:10.14135/j.cnki.1006-3080.2016.06.017

一种基于安全策略的云数据访问控制优化技术

苏雪¹, 宋国新²

(1. 武汉铁路职业技术学院, 武汉 430205;

2. 华东理工大学信息科学与工程学院, 上海 200237)

摘要:云存储安全给访问控制技术带来了新的挑战。提出了一种基于安全策略的云数据访问控制优化方法,基本思想是利用数据关联关系进行安全策略精化,基于属性的安全策略中融合了角色和信誉特征。利用数据关联关系对云数据进行细粒度划分,得到相应的数据块;对策略中的规则和数据块进行匹配和冲突消除,得到精化的安全策略。理论分析和实验结果表明,本文方法对于数据量大的云数据访问控制具有较高的效率。

关键词:云存储;安全策略;信誉;访问控制;优化

中图分类号:TP311

文献标志码:A

A Security Policy-Based Approach to Optimizing Cloud Data Access Control

SU Xue¹, SONG Guo-xin²

(1. Wuhan Railway Vocational College of Technology, Wuhan 430205, China;

2. School of Information Science and Engineering,

East China University of Science and Technology, Shanghai 200237, China)

Abstract:Cloud storage arouses new challenges to access control techniques. This paper proposes a security policy-based method to optimize cloud data access control. The basic idea is to refine security policies according to correlations among cloud resources. Attribute-based security policies comprehend the role of subject and his credit as well. Cloud data are partitioned to fine-grained data blocks, which is guided by correlations among the data. Security policy refinement is obtained by matching policy rules with data blocks and resolving rule conflicts. Theoretical analysis and experiment results show that the proposed method is efficient for access control of cloud data with huge quantity.

Key words:cloud storage; security policy; credit; access control; optimization

数据安全是云计算应用面临的主要挑战之一^[1],访问控制是数据安全和网络通信安全的基本方法。以 DAC, MAC, RBAC^[2-3] 为代表的访问控制方法在操作系统、数据库安全领域已经得到广泛应用,基于安全策略的访问控制方法适合于网络环境下托管数据的安全管理。XACML (eXtensible Access Control

Markup Language) 是一种基于 XML 格式的访问控制标准,被广泛应用于网络应用和网络服务中^[4]。XACML 是一种层次化语言模型,它提供了策略集、策略和规则 3 种对象及对象之间的融合算法,用于仲裁安全规则之间的冲突并设定默认解决方案,从而保证访问控制决策的一致性。

收稿日期:2016-09-02

基金项目:湖北省教育厅科学技术规划项目(B2015412);国家科技重大专项项目(2010ZX03004-003-03)

作者简介:苏雪(1972-),女,硕士,副教授,主要研究方向为计算机网络通信与信息系统。E-mail:yuanyuan720306@qq.com

基于 XACML 的访问控制研究已有大量成果。Hu 等^[5]提出了一种应用于 Web 访问控制中基于逻辑的 XACML 策略管理机制,并对规则的冲突和冗余进行检测和消除。Wang 等^[6]提出一种相关类型的策略优化引擎,并对层次化的 XACML 利用树状图分析和优化。然而上述文献并未充分考虑到策略的安全性,同时基于策略本身的优化方法,效率依然取决于策略长度,并未从根本上优化决策算法的复杂度。Pei 等^[7]提出了规则优化方法,但策略中没有考虑角色和信誉的因素。Said^[8]提出了一个评估策略执行效率的框架,并形式化地描述了 XACML 及策略相似性的度量方法。Bertolino 等^[9]对 XACML 的自动测试方法进行了研究,但对于具体的优化方式并未给出可行性方案。

本文在基于属性的安全策略中融合了角色和信誉特征,并提出了一种基于策略的云数据访问控制优化算法 pbacPDP (policy-based access control Policy Decision Point)。利用数据关联关系对云数据进行划分,对策略和数据块进行匹配和冲突消除,得到优化的安全策略。

1 安全策略模型

1.1 策略的描述

一个 XACML 策略可以包含多条规则,在一个规则中,主体、资源、行为构成其基本属性,同时规则可以具有执行条件约束。规则具有“允许”和“拒绝”两种效用。由多个规则构成的策略具有一个融合算法,用来解决规则之间的冲突。同样,策略集也具有融合算法,解决策略之间的冲突。XACML 自带的融合算法有 4 种,分别为允许优先算法 (Permit-Override)、拒绝优先算法 (Deny-Override)、首次适用算法 (First-Applicable) 和唯一适用算法 (Only-one-Applicable)。XACML 安全策略模型的形式化描述如下:

$Policyset ::= \langle target \rangle, \{ \langle Policyset \rangle | \langle Policy \rangle \} +, \langle CA \rangle$

$Policy ::= \langle target \rangle, \{ \langle rule \rangle \} +, \langle CA \rangle$

$rule ::= (\langle target \rangle, \langle effect \rangle, \langle condition \rangle, \langle obligation \rangle)$

$target ::= \{ (\langle role \rangle, \langle object \rangle, \langle action \rangle) \} *$

$request ::= \{ (\langle subject \rangle, \langle object \rangle, \langle action \rangle) \} +$

$effect ::= 'permit' | 'deny'$

$CA ::= 'PO' | 'DO' | 'FA' | 'OA'$

安全模型中角色 (Role) 和主体 (Subject) 是多

对多的关系。规则 (Rule) 在策略中出现的次序通过函数 $seq: Rules \rightarrow N$ 定义。

设 $S = \{s_1, s_2, s_3, s_4\}$ 为云存储数据资源集,它们之间存在的关联关系如图 1 所示,资源集的数据块划分为 $\{d_1, d_2, d_3, d_4, d_5, d_6\}$ 。

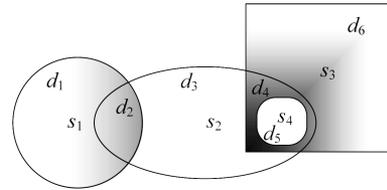


图 1 数据资源关联关系示例

Fig. 1 An example of data resource correlation

图 2 给出了一个 XACML 安全策略案例,标识为 P_{01} ,定义了对数据的访问控制策略,它包含 r_1, r_2, r_3 3 个规则。

```
<policy policyID="P01" CA="Deny-Overrides">
  <target>
    <Action>Read, Write</Action>
  </target>
  <Rule RuleID="r1" Effect="Permit">
    <target>
      <Role>Student, Faculty</Role>
      <Resource>s1, s2</Resource>
      <Action>Read, Write</Action>
    </target>
    <Condition> Time ∈ [8:00, 17:00] credit ≥ 60
  </Condition>
  </Rule>
  <Rule RuleID="r2" Effect="Permit">
    <target>
      <Role>Student</Role>
      <Resource>s4</Resource>
      <Action>Read</Action>
    </target>
  </Rule>
  <Rule RuleID="r3" Effect="Deny">
    <target>
      <Role> Student, Staff</Role>
      <Resource>s2, s3</Resource>
      <Action>Write</Action>
    </target>
    <Condition>Time ∈ [6:00, 18:00] credit ≤ 90
  </Condition>
  </Rule>
</policy>
```

图 2 XACML 安全策略示例

Fig. 2 An example of XACML security policy

1.2 访问控制过程

访问控制的主要模块包括策略执行点 (PEP) 和策略决策点 (PDP)。PEP 接受用户请求、创建一个 XACML 请求并发送到 PDP,而 PDP 评估请求并返回一个响应。该响应可以是允许访问 (permit),也可以是拒绝访问 (deny)。假定 Alice 是学生,她的信誉值为 85,系统接受到如下请求:

$req (Subject = 'Alice', Resource = 's_2')$

Action = 'Write', Condition = '12:00')

策略 P_{01} 中规则 r_1 和 r_3 与请求匹配, 根据规则 r_1 该请求得到的相应是 'permit', 根据规则 r_3 该请求得到的相应是 'deny'。因为策略 P_{01} 的融合算法 CA 选取的是 'deny-override', 所以最终的控制决策是 'deny'。

云存储提供了海量数据的管理和服务功能, 用户的量大, 需求个性化要求强。对于每个用户请求, 决策过程需要逐个检查规则是否匹配, 并根据规则条件和融合算法进行最终决策, 对于云存储管理是一个巨大的负担。本文的目标是提出细粒度的访问控制优化技术, 在实施安全策略的同时提高决策的效率。

2 访问控制优化技术

2.1 基本定义

为了实现资源的独立存储, 同时考虑到资源安全属性, 通过资源分隔算法将资源切分为相互独立且具有安全属性的数据块。由于资源是策略的一部分, 是 XACML 中规则的一个基本属性, 将规则绑定到资源上, 资源交集生成的新数据块将继承所有相交资源的规则, 实现了策略在资源维度上的投影。通过规则优化算法对每个数据块上的规则进行冲突和冗余的检测和消除, 实现策略的最优化。策略中的规则之间存在冗余和冲突, XACML 提供的 4 种融合算法虽然可以解决冲突, 但是决策效率较低, 并不具备系统性的规则优化方法。本文提出的规则优化算法是在决策之前对所有相关的规则在数据块维度上进行优化。

定义 1(规则匹配) 假设云端用户的请求为 $req = (\langle sub \rangle, \langle obj \rangle, \langle act \rangle)$, 规则 $r = (\langle tg \rangle, \langle eff \rangle, \langle cond \rangle, \langle obl \rangle)$ 。如果 $role(sub)$ 与 tg 相交, 并且 $cond$ 满足, 则称规则 r 与请求 req 匹配。

记为 $r | = req$ 。

定义 2(规则冲突) 设 r_i, r_j 为安全策略中的规则, 如果存在请求 req 使得 $req | = r_i \wedge req | = r_j$ 成立, 则称 r_i 与 r_j 冲突, 记为 $r_i \otimes r_j$ 。两者的相交的部分记为 Δ_{r_i, r_j} , 包含 Δ_{r_i} 属于 r_i , Δ_{r_j} 属于 r_j 。

当 $r_i \otimes r_j$ 且 $r_i \cdot effect = r_j \cdot effect$ 时, 有 $\Delta_{r_i} = \Delta_{r_j}$ 。若删除 Δ_{r_i} 或 Δ_{r_j} 对策略的最终决策不产生影响, 则其可移除。

2.2 优化算法 pbacPDP

在不同的策略融合算法作用下, 移除判定准则不相同, 分别定义 4 种融合算法中的移除准则。

准则 Cr1 允许优先算法

当 $r_i \otimes r_j$ 且 $r_i \cdot effect = permit$, 则 Δ_{r_j} 可移除。

当 $r_i \otimes r_j$ 且 $r_i \cdot effect = deny$, 则 Δ_{r_i} 可移除。

准则 Cr2 拒绝优先算法

当 $r_i \otimes r_j$ 且 $r_i \cdot effect = deny$, 则 Δ_{r_j} 可移除。

当 $r_i \otimes r_j$ 且 $r_i \cdot effect = permit$, 则 Δ_{r_i} 可移除。

准则 Cr3 首次适用算法

假设 r_i, r_j 在策略中的出现顺序为 $seq(r)$ 。

当 $r_i \otimes r_j$ 且 $seq(r_i) < seq(r_j)$, 则 Δ_{r_j} 可移除。

当 $r_i \otimes r_j$ 且 $seq(r_i) > seq(r_j)$, 则 Δ_{r_i} 可移除。

准则 Cr4 唯一适用算法

若 r_i 与 r_j 同时匹配一个请求, 共同匹配部分为 Δ_{r_i, r_j} , 非唯一匹配返 "NotApplicable", 该情况下, 移除 Δ_{r_i} 与 Δ_{r_j} 。

优化的访问控制算法 pbacPDP 的伪代码如图 3 所示。RuleSet_i 表示绑定到数据块上的规则集合, 算法根据上述优化规则对 RuleSet_i 规约。

```

// 输入: 安全策略P和用户请求req
// 输出: 访问控制决策 "permit" or "deny"
Refine(P)
  foreach rule ∈ GetRule(P) do //bind rules to segment
    foreach s ∈ S do
      if s ⊂ GetRelatedResource(rule) then
        bind(rule, s);
    ← with bound rules on each element;
    foreach g ∈ G do //refine rules on each segment
      foreach pair(ri, rj) ∈ CruleSet2 do
        if ri ⊗ rj then
          coupleSet.add(Δri, rj); //overlap of ri, rj
          foreach Δri, rj ∈ coupleSet do
            case CA = Permit-override then
              按Cr1执行;
            case CA = Deny-override then
              按Cr2执行;
            case CA = First-applicable then
              按Cr3执行;
            case CA = Only-one-applicable then
              按Cr4执行;
  P' := Refined(P);
  Return PDP(P', req)

```

图 3 pbacPDP 算法伪代码

Fig. 3 Pseudo-codes of the pbacPDP algorithm

定理 1(算法的正确性) 假设 P 为原始安全策略, P' 为优化后的细粒度安全策略, 对于每个请求 req 数据块, $P | = req$ 当且仅当 $P' | = req$ 。

证明:

(1) 必要性: 假设 $P | = req$, 如果策略 P 中没有规则冲突, 则 $P' = P$, 从而显然 $P' | = req$ 。假设策

略 P 中存在规则冲突,不妨令 $r_i \otimes r_j$ 。

情况 1:如果 $r_i \cdot \text{effect} = \text{permit}$,CA 采纳的是准则 Cr1,则无论 $r_j \cdot \text{effect} = \text{permit}$ 或 deny ,删除 Δ_{r_j} 后,最终对 req 的决策仍然是 permit ,所以 $P' | = \text{req}$ 。

情况 2:如果 $r_i \cdot \text{effect} = \text{deny}$ 并且 $r_j \cdot \text{effect} = \text{deny}$,则 $\Delta_{r_i} = \Delta_{r_j}$,删除 $\Delta_{r_i} (\Delta_{r_j})$ 仍然 $P' | = \text{req}$ 。然而,如果 $r_i \cdot \text{effect} = \text{deny}$ 并且 $r_j \cdot \text{effect} = \text{permit}$,CA 采纳的是准则 Cr1,删除 Δ_{r_i} 最终对 req 的决策仍然是 permit ,所以 $P' | = \text{req}$ 。同样地,如果 CA 采纳 Cr2、Cr3、Cr4 准则,结论成立。

(2)充分性:假设 $P' | = \text{req}$,如果策略 P 中没有规则冲突,则 $P' = P$,从而显然 $P | = \text{req}$ 。假设策略 P 中存在规则冲突,不妨令 $r_i \otimes r_j$ 。

情况 1:如果 $r_i \cdot \text{effect} = \text{permit}$,CA 采纳的是准则 Cr1,则 P' 为 P 删除 Δ_{r_j} 后的策略,依据 $r_i \cdot \text{effect} = \text{permit}$ 可知,先前的策略对 req 的决策是 permit ,所以 $P | = \text{req}$ 。

情况 2:如果 $r_i \cdot \text{effect} = \text{deny}$ 并且 $r_j \cdot \text{effect} = \text{deny}$,则 $\Delta_{r_i} = \Delta_{r_j}$,删除 $\Delta_{r_i} (\Delta_{r_j})$ 。 $P' | = \text{req}$ 意味着存在其他规则 r_k ; $r_k \cdot \text{effect} = \text{permit}$,所以 $P | = \text{req}$ 。然而,如果 $r_i \cdot \text{effect} = \text{deny}$ 并且 $r_j \cdot \text{effect} = \text{permit}$,CA 采纳的是准则 Cr1,删除 Δ_{r_i} 最终对 req 的决策与未删除前一样,均为 permit ,所以 $P | = \text{req}$ 。

类似地,CA 采纳 Cr2、Cr3、Cr4 时结论可证。

3 访问控制性能分析

3.1 复杂度分析

假设在策略 P 中,存在 k 个规则 $\text{rules} = \bigcup_{i=1}^k r_i$; n 个资源 $\text{resources} = \bigcup_{i=1}^n \mathfrak{R}_i$; 通过调用以上算法得到 m 个数据块 $\text{blocks} = \bigcup_{i=1}^m b_i$ 。假设在每个数据块 i 上绑定 h_i 个规则 $\text{rules}_{\text{block}_i} = \bigcup_{j=1}^{h_i} r_j b_j$,这些规则产生 c_i 个冲突 $\text{conflicts}_{\text{block}_i} = \bigcup_{j=1}^{c_i} c b_j$ 。

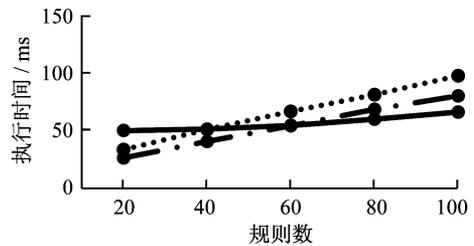
在访问控制优化算法中,计算时间包含数据块划分时间、规则精化时间、冲突检查时间和冲突解决时间。数据块划分的时间复杂度为 $O(n \log_2 m)$,规则精化复杂度为 $O(km)$,冲突检查时间复杂度为 $O(h_i \log_2 h_i)$,冲突解决的时间为 $O(c_i)$,从而访问控制优化算法的总时间复杂度为 $O(n \log_2 m + km + \sum_{i \in 1..|\text{Conflicts}|} (h_i \log_2 h_i + c_i))$ 。

算法需要额外的空间开销,用以存储数据块以

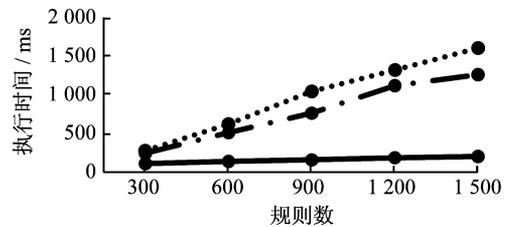
及匹配规则,其空间复杂度为 $\Theta(m)$ 。

3.2 实验分析

为分析本文方法对不同的规则数、规则冲突数、资源耦合度的影响,进行了两组实验。实验 1 的规则数分别为 20、40、60、80、100,规则冲突数为 60,资源元数为 64;实验 2 的规则数分别为 300、600、900、1 200、1 500,规则冲突数为 60,资源元数为 160。假定云数据访问请求者平均有 80%符合信用标准。实验采用 SUN-XACML 的 API,实验环境为: Intel(R) Core(TM) i3-2330M CPU 2.30 GHz, 4 GB RAM, Win7 操作系统。方法比较对象分别是 simplePDP^[4] 和 melcoePDP^[10],其中 simplePDP 使用列表结构遍历规则进行匹配,melcoePDP 则采用数据属性簇遍历。实验结果如图 4 所示。可以看出,在规则数和资源元数较少时,本文方法与 simplePDP 和 melcoePDP 相比在效率上相仿或者略低;但随着规则数和资源元的增加,本文方法的效率比 simplePDP 和 melcoePDP 具有明显的提高,适用于数据量大的云存储访问控制。



(a) 实验1的性能图



(b) 实验2的性能图

●●● simplePDP; ●●● melcoePDP; ●●● pbacPDP

图 4 算法对比分析

Fig. 4 Comparison analysis of algorithms

4 结束语

本文提出了一种基于安全策略的云数据访问控制优化方法。该方法利用 XACML 描述访问控制策略,在安全策略中融合了用户角色和信誉特征,适合网络环境中数据资源的访问控制。利用数据关联关系对云数据进行细粒度划分,得到相应的数据块。进一步对策略中的规则和数据块进行匹

配和冲突消除,得到优化的安全策略,可以对云存储资源进行细粒度的访问控制。实验结果表明,本文提出的 pbacPDP 算法对于数据量大的云数据安全具有较高的效率和适应性。

进一步的研究工作包括对云数据访问控制模型和系统的设计、第三方的用户信用管理和维护、相关工具的研究和开发。

参考文献:

- [1] YANG K, JIA X, REN K, *et al.* DAC-MACS: Effective data access control for multiauthority cloud storage systems[J]. IEEE Transactions on Information Forensics and Security, 2013, 8(11): 1790-1801.
- [2] MASOOD A, BHATTI R, GHAFOR A, *et al.* Scalable and effective test generation for role-based access control systems [J]. IEEE Transactions on Software Engineering, 2009, 35(5): 654 - 668.
- [3] ARDAGNA C A, VIMERCATI D C, PARABOSCHI S. Expressive and deployable access control in open Web service applications[J]. IEEE Transactions on Services Computing, 2011, 4(2): 96-109.
- [4] GODIK S, MOSES T. Extensible access control markup language (XACML) version1. 1 [EB/OL]. [2003-04-10]. <http://docs.oasis-open.org/XACML>.
- [5] HU H X, AHN G J. Discovery and resolution of anomalies in web access control policies[J]. IEEE Dependable and Secure Computing, 2013, 10(6): 341-354.
- [6] WANG Y Z, FENG D G, ZHANG L W. XACML policy evaluation engine based on multi-level optimization technology[J]. Journal of Software, 2011, 22(2): 323-338.
- [7] PEI X, YU H, FAN G. Achieving efficient access control via XACML policy in cloud computing[C]//Proceedings of the 27th International Conference on Software Engineering and Knowledge Engineering. USA: KSI Research Inc, 2015: 110-115.
- [8] SAID M, SHEHAB M, ANNA S. Adaptive reordering and clustering-based framework for efficient XACML policy evaluation[J]. IEEE Service Computing, 2011, 4(4): 300-313.
- [9] BERTOLINO A, DAOU DAGH S, LONETTI F. Automated testing of extensible access control markup language-based access control systems [J]. IET Software, 2013, 7(4): 203-212.
- [10] JAJODIA S, SAMARAT P. A logical language for expressing authorizations [C] //Proceedings of IEEE Symposium on Security & Privacy. USA: IEEE, 1997: 31-42.

(上接第 844 页)

- [17] WANG L X, MENDEL J M. Back-propagation fuzzy system as nonlinear dynamic system identifiers [C]//IEEE International Conference on Fuzzy Systems. USA: IEEE, 1992: 1409-1418.
- [18] MENDEL J M. Computing derivatives in interval type-2 fuzzy logic systems [J]. IEEE Transactions on Fuzzy Systems, 2004, 12(1): 84-98.
- [19] KAMALI C, PASHILKAR A A, RAOL J R. Evaluation of recursive least squares algorithm for parameter estimation in aircraft real time applications [J]. Aerospace Science and Technology, 2011, 15(3): 165-174.
- [20] 王立新. 模糊系统与模糊控制教程[M]. 北京: 清华大学出版社, 2003: 155-163.

欢迎订阅

《华东理工大学学报(自然科学版)》

地址:上海市梅陇路 130 号 436 信箱 邮编:200237

邮发代号:4-382