文章编号:1009-3087(2016)04-0123-06 DOI:10.15961/j.jsuese.2016.04.017

# 改进的安全策略评价管理决策图

罗霄峰[1],罗万伯[2*]
(1.四川大学 锦江学院,四川 成都 610065;2.四川大学 计算机学院,四川 成都 610065)

**摘 要**:针对 Canh Ngo 等为安全策略评价和管理所提出的 MIDD 和 X-MIDD 方法的不足,从一般 ABAC 模型出发,对其进行了改进。设计了新的图结构 iMIDD 和 iX-MIDD,新图的边用上结点变量值范围(简约的区间划分)及状态组成的元组表示,可更好地标注在决策中有关键作用的重要属性,有利于决策过程中更精细化处理。iX-MIDD 的决策-叶结点也做了扩展,增加了组合算法信息,便于在对访问请求进行决策时使用。给出了应用本文方法进行策略元素匹配、策略评估,以及从 iMIDD 生成 iX-MIDD 的流程。复杂度分析及仿真实验表明,本文方法的时间、空间复杂度和性能均与 MIDD 方法相当。新方法完全能用于策略管理的多种应用。

**关键词**:访问控制;决策图;安全策略;策略管理
**中图分类号**:TP309;TP391 **文献标志码**:A

## Improved Decision Diagrams for Security Policy Evaluation and Management

*LUO Xiaofeng*[1], *LUO Wanbo*[2*]
(1. Jinjiang College, Sichuan Univ., Chengdu 610065, China;
2. College of Computer Sci., Sichuan Univ., Chengdu 610065, China)

**Abstract**: In order to overcome the shortcomings of MIDDs approach proposed by Canh Ngo et al, some improvements were proposed. New graph structures whose edge is a tuple of node-variable reduced interval partition and the state value marking critical attributes were designed for the improved MIDDs and X-MIDDs, named iMIDDs and iX-MIDDs respectively. In addition, the combining-algorithm identifier was also added to iX-MIDDs decision-leaf nodes. Operations and processing of policy elements match, policy evaluation, and iMIDD to iX-MIDD transformation were introduced. Complexities analysis and simulation showed both space and evaluation time complexities of proposed approach are equivalent to the MIDDs'. New approach could be applied in various policy management problems.

**Key words**: access control; decision diagrams; security policy; policy management

安全策略在信息安全中有十分重要的作用[1]。所谓安全策略,是在一个安全域内所有关乎安全的相关活动应遵循的一套规则。这些规则定义哪些活动对系统、组织机构或实体才是安全的。安全域内一般有多种资源和应用,相应地,会有多种安全策略。这些安全策略的评价和管理,十分必要,具有极大的挑战。

安全策略由于其重要性,历来是研究的重点。其研究涉及很多方面,包括策略的表示、分析、集成、测试、评价、管理等。文献[2-3]列举了这方面主要的重要文献及成果。其中值得一提的是,Fisler 等提出的 XACML 逻辑[4],并用多端二叉决策图(multi-terminal binary decision diagram, MTBDD)[5]在 Margrave 项目中实现。

在策略表示方面,现在最有影响的是 OASIS 开发的可扩展访问控制置标语言(eXtensible access control markup language, XACML)[6]。XACML 从 1.0、2.0版到现在的3.0版,越来越强大。在本文中,如无特别说明,均指 3.0 版。用 XACML 表示的策略在一些大系统中得到了应用,而文献[2-3]都针对 XACML 策略分别进行了研究。

Rao等在分析现有研究的不足后,提出精细集

**收稿日期**:2015-09-13
**作者简介**:罗霄峰(1974—),男,博士. 研究方向:信息安全. E-mail:xluo_f@aliyun.com.cn
*通信联系人 E-mail:wbluo@scu.edu.cn

成代数(fine-grained integration algebra,FIA),用于精细地集成复杂的策略。他们还给出了利用FIA,基于MTBDD生成实际XACML集成策略的方法[2]。但是,Rao等定义的3-值FIA不能表示全部不确定的值(indeterminate values),并且也不能区分在不同域上操作时XACML对象(target)和规则评估之间的差别。至于策略评估效能问题,以及重要属性(又称关键元素)评估的处理问题等,现有的相关研究也未能很好解决[3]。针对这些不足,Ngo等提出多数据类型区间决策图(multi-data-type interval decision diagram,MIDD)解决方式,将XACML策略转换为决策树,从而有效改进策略评估性能。MIDD用结点表示属性,此时属性又称为变量或元素;边表示属性的区间划分(即属性值或变量值)。这种方法也能用于解决策略冗余检测、策略测试和比较等策略管理问题[3]。

同其他相关研究相比,MIDD方法既包含了XACML逻辑的分析,也包含了实用的实现机制,是比较好的方法。

研究发现,MIDD方法虽然能支持大部分XACML特征(如连续数据类型,复杂的比较组合算法语义的正确性,差错处理和重要属性设置等),但在实际使用中,仍存在着一些缺憾。

作者针对文献[3]的一些问题,试图提出相应的改进,以弥补其不足。

主要改进包括:

1)改进MIDD的定义,解决正确标识策略、策略集、规则和规则集中的重要属性问题;

2)改进X-MIDD的定义,支持重要属性和各种组合算法;

3)不但支持XACML表示的策略,也支持其他基于属性的安全策略。

## 1 预 备

### 1.1 问题的提出

访问(access)可以理解为主体(例如一个用户)对客体(例如一份文件)执行操作(例如读),控制是管理风险的方法。

访问控制可以理解为对访问请求的处理:准许(permit)或拒绝(deny)该请求。处理的依据就是访问控制策略。访问控制决定了谁能够访问系统,能访问系统的何种资源以及如何使用这些资源。

已经开发了多种访问控制模型,其中有一些得到了广泛应用[1]。其中基于属性的访问控制(attribute based access control, ABAC)引人注目。ABAC的概念已经存在多年,它按照规则来评估一个用实体(主体和客体)、操作以及环境的相关属性表示的请求,以控制对客体的访问[7]。ABAC是一类模型的统称。例如,CUC就是一种ABAC[8]。

XACML表示的访问控制也属于ABAC,不但用于表示访问控制策略,也用于表示访问请求和决策结果信息。

为了精细地处理访问请求,XACML定义了一种称为MustBePresent的元素,来标识一个属性值是否重要(critical)。如果一个策略或规则要求的某个属性在访问请求中不存在,策略决策点(policy decision point,PDP)将用MustBePresent来管理该缺失属性的处理:如果MustBePresent = "true",则该缺失属性值为"indeterminate";否则(即MustBePresent ="false",或没有MustBePresent),则其值为空包(empty bag)[6]。

在文献[3]的MIDD中,重要属性在内部结点中用状态值$s \in \{F, IN\}$标识。如果结点$x$是重要属性,则$s = IN$;否则,$s = F$。在单条路径时,这是没有问题的。在策略(规则)合并后,就出现问题了。

下面仍以文献[3]采用的策略例子来说明。此策略简化表示如下:

Policy{ id: $P_1$ ;combine_algo:po;
target:{ $(vol \geq 100) \wedge (vol \leq 500)$ };
children:{ $R_1, R_2$ }
}
Rule{id: $R_1$; effect:permit;
target:{
$[(100 \leq \underline{vol} \leq 150) \wedge (12 \leq t \leq 17) \wedge (3 \leq p \leq 4)] \vee$
$[(300 \leq vol \leq 500) \wedge (1 \leq p \leq 2)] \vee$
$[(100 \leq vol \leq 500) \wedge (6 \leq t \leq 9) \wedge (1 \leq p \leq 2)]$
};
obligations:{ $O_1$, permit}
}
Rule{id: $R_2$ ;effect:deny;
target:{
$[(vol = 100) \wedge (t = 17)] \vee$
$[(100 \leq vol \leq 300) \wedge (t = 9)] \vee$
$[(vol = 500) \wedge (t \geq 12)]$
};
obligations:{ $O_2$, deny}
}

策略例子中$vol$为属性volume的省略,$t$为属性

time 的省略,$p$ 为属性 price 的省略。其有一项 *vol* 带下划线,表示其值是重要的。其余都没有下划线,表示它们不是重要值。

图 1 是规则 $R_1$ 对象(target) 的 3 个子 target 的 MIDD 图,其中:

图 1(a) 对应于 $100 \le \underline{vol} \le 150 \wedge 12 \le t \le 17 \wedge 3 \le p \le 4$;图 1(b) 对应于 $300 \le vol \le 500 \wedge 1 \le p \le 2$;图 1(c) 对应于 $100 \le vol \le 150 \wedge 6 \le t \le 9 \wedge 1 \le p \le 2$。

图 1(a) 的根结点为 *vol*(IN),表示此结点代表的属性 *vol* 是重要属性;图 1(b) 和(c) 根结点均为 *vol*(F),表示 *vol* 不是重要属性[3]。

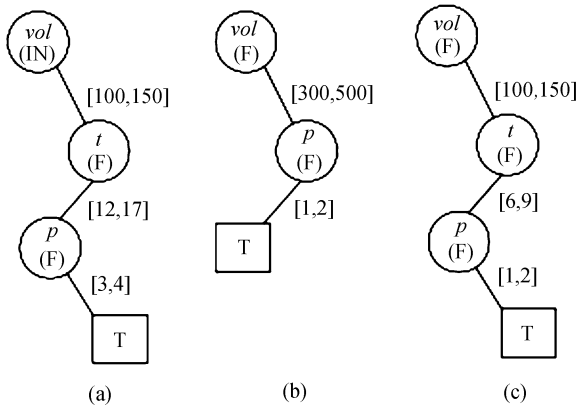


**图 1　$R_1$ 对象表达式的 MIDD**

**Fig. 1　MIDDs of the $R_1$ target expression**

图 2 是 $R_1$ 对象的 MIDD 图,根结点为 *vol*(F),其中,F 表示 *vol* 为非重要属性,而图 1(a) 根结点为 *vol*(IN),IN 表示 *vol* 为重要属性。显然图 2 已经丢掉了图 1(*a*) 中 *vol* 的重要属性信息[3]。

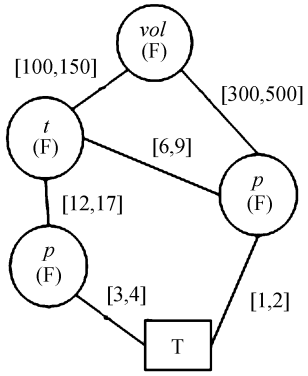


**图 2　$R_1$ 对象的 MIDD**

**Fig. 2　MIDD of the $R_1$ Target**

XACML 还定义了多种规则组合算法(rule-combining algorithms)和策略组合算法(policy-combining algorithms)(表 1)。组合算法必须包含在策略或策略集中,指导 PDP 进行策略处理。

**表 1　XACML 组合算法**

**Tab. 1　XACML combining-algorithms**

| 组合算法 | 简记 |
|---|---|
| permit-overrides | po |
| deny-overrides | do |
| first-applicable | fa |
| only-one-applicable | ooa |
| ordered-permit-overrides | opo |
| ordered-deny-overrides | odo |
| permit-unless-deny | pud |
| deny-unless-permit | dup |

遗憾的是,像 XACML 那样定义的策略及规则组合算法在文献[3]定义的 MIDD 和 X-MIDD 中也难以表示出来。事实上,人们可能用多种方式定义组合算法,以满足需要。例如,文献[9]提出弱一致性、强一致性、弱多数、强多数和超多数许可等原则来定义组合算法。显然,更难用 MIDD 法处理。

### 1.2　基本概念

不失一般性,文献[3]中的下述记号可以推广到 ABAC 中。

1)令集 $D_i$ 表示连续数据类型 $i$ 的全部有序域。$D_i$ 可以是任何元素的值集。例如,CUC 中主体属性、客体属性和周景属性等的值集,XACML 中 target 元素、condition 元素等的值集,也可以是判断结果(逻辑值) 等。

利用 $D_i$,一个访问请求 $X$ 可以表示为:

$$X = D_1 \times D_2 \times \cdots \times D_n \tag{1}$$

2) $V_M$ 表示匹配元素评估结果值集。

对于传统的访问控制,结果值集是2值逻辑,$V_M := \{T, F\}$。此处 $T$ 表示 true,或匹配;$F$ 表示 false,或不匹配。*XACML* 中 $V_M := \{T, F, IN\}$,IN 表示 indeterminate,即无法确定。

匹配元素评估 $f$ 可以表示为:

$$f: D_1 \times D_2 \times \cdots \times D_n \to V_M \tag{2}$$

3) $E := \{P, D\}$ 表示逻辑决策结果集。此处 P 表示准许(permit),D 表示拒绝(deny)。

4) $V_R$ 表示规则、策略和策略集元素的决策结果集。

对于传统的访问控制,$V_R := \{P, D\}$。

对于 XACML 表示的访问控制,$V_R := \{P, D, N, IN_P, IN_D, IN_{PD}\}$。此处 P 和 D 与传统访问控制相同;N 表示不可用(notapplicable);$IN_P$ 表示无法确定,可能评估为准许而不是拒绝(indeterminate" which could have evaluated to "permit",but not

"deny");$IN_D$ 表示无法确定,可能评估为拒绝而不是准许("indeterminate" which could have evaluated to "deny", but not "permit");$IN_{PD}$ 表示无法确定,可能评估为准许或拒绝("indeterminate" which could have evaluated to "deny" or "permit")。

规则、策略和策略集元素决策 $f$ 可以表示为:

$$f: D_1 \times D_2 \times \cdots \times D_n \rightarrow V_R \quad (3)$$

**定义** 1(数据区间) 一个数据区间 $I \subset D_i$ 是域 $D_i$ 中的一个值范围。

数据区间可能是点,也可能是由两个端点组成的范围。根据端点值是否包含在内,这个范围可能是全开,全闭,或半开、半闭的区间。

**定义** 2(区间划分) 域 $D_i$ 中的一个区间划分 $P$ 是该域的一个不相交的区间集[3]。即:

$p = \{I \subset D_i : \forall I_i, J_j \in P, i \neq j, I_i \cap I_j = \varnothing\}$。

**定义** 3(覆盖) 给定一个域 $D_i$,如果划分集 $P(D_i) = \{P_1, P_2, \cdots, P_n\}$ 满足

$$D_i = \bigcup_{p \in P(D_i)} (\bigcup_{I \in p} I),$$

则称 $P(D_i)$ 是域 $D_i$ 的一个覆盖。

应用中需要的是无相交区间的覆盖,即:

$$\forall i, j \in \{1, n\}, i \neq j : p_i \cap p_j = \varnothing,$$

则 $P(D_i)$ 称为不相交覆盖。

**定义** 4(简约的区间划分) 在覆盖相同数据范围的区间划分中,具有最少区间数的那个划分称为简约的区间划分[3]。

## 2 改进的决策图方法

### 2.1 改进的决策图

**定义** 5(改进的 MIDD) 改进的 MIDD(iMIDD, Improved MIDD)是式(2)的一种有根、有向的无循环图表示:

1) iMIDD 的外结点包含值 T,表示 true,或 matched。外结点也称为 T - 叶结点。

2) iMIDD 的每个内结点是一个元组($x$, $C$)。其中,$x$ 为结点变量;$C$ 为元组($ed$, $c$) 数组,$ed$ 为一条输出边,$c$ 为 $ed$ 的下结点。下结点可能是内结点,也可能是外结点。

3) iMIDD 的边 $ed$ 是一个元组($p$, $s$)。其中,$p$ 是上结点变量的一个简约的区间划分;$s \in \{F, IN\}$,是其状态值。如果上结点变量 $x$ 是关键变量,$s$ = IN,否则 $s$ = F。

**定义** 6(改进的 X-MIDD) 改进的 X-MIDD (Improved X-MIDD, iX-MIDD)是式(3)的一种有根、有向的无循环图表示:

1) iX-MIDD 的外结点是一个元组($e$, $ca$, $O$)。其中,$e \in \{P, D\}$,是规则或策略的结果(effect);$ca$ 是规则或策略组合算法;$O$ 是义务 - 忠告列表,可以缺少。外接点又称为决策 - 叶结点。

2) iX-MIDD 的每个内结点是一个元组($x$, $O$, $C$)。其中,$x$ 为结点变量。$C$ 为元组($ed$, $c$) 数组。$c$ 为 $ed$ 的下结点。下结点可能是内结点,也可能是外结点。$ed$ 为一条输出边,是一个元组($p$, $s$),$p$ 是上结点变量的一个简约的区间划分;$s \in V_R$,是其状态值。如果结点变量 $x$ 不是关键变量,则 $s$ = N;否则 $s = IN_e$。此处 $e$ 等于外结点($e$, $ca$, $O$) 中的 $e$。$O$ 同外结点定义,但当 $ed.s = N$ 时,$O$ 为空。

图 3 和 4 分别是第 2.1 节策略例 $R_1$ 和 $R_2$ 的 iMIDD 和 iX-MIDD 图。

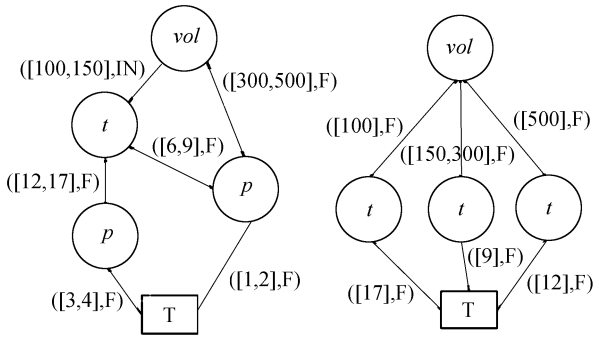


**图 3 对象元素 iMIDD 例**

**Fig. 3 Sample iMIDDs of the target elements**

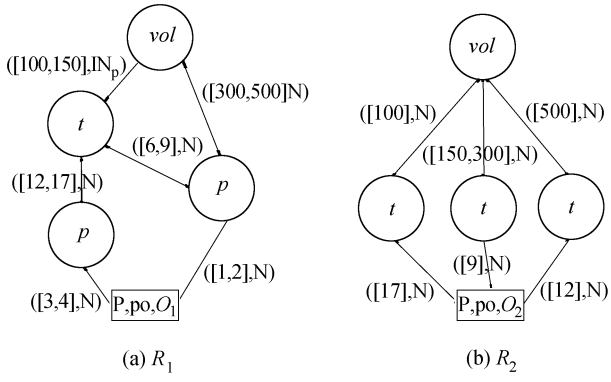


**图 4 规则 $R_1$ 和 $R_2$ 的 iX-MIDD**

**Fig. 4 iX-MIDDs of Rules $R_1$ and $R_2$**

下面对改进作简要说明。

首先,将图 3 和 4 与图 2 和文献[3]的图 5(a)对比,不难看出它们的区别:MIDD 和 X-MIDD 中丢失的重要属性标记,在改进图中完全保存了下来。重要属性在安全策略中可能举足轻重。该修改,正好可以充分发挥重要属性在决策中的作用,有利于

决策过程中更精细化处理。例如,服务质量(quality of service,QoS)系统可以根据是否满足重要属性条件,提供不同服务质量的服务。

如果适当修改 MIDD 及 X-MIDD 子图合并规则,也可能在结点保留重要属性标记。例如,只要至少有一个子图结点标记为重要属性,合并图的相应结点就标记为重要属性。但会产生新问题:不应该提供高品质服务的用户,可能被提供高品质服务了。

其次,该修改显式地在叶结点中保留了组合算法,可以更好地表示各种规则组合算法和策略组合算法。组合算法的目的,主要是在决策过程中应用,而规则和策略组合算法直接影响策略判决[2,7,10-12]。访问控制系统通常有多条安全策略。接收到一个访问请求时,系统要用这些安全策略对此请求进行评估。由于网络通信等错误,一个准许(permit)的规则,可能返回{P,NA},即 permit 或 not applicable。拒绝(deny)的规则,也可能返回{D,NA}。文献[7]指出,不同的组合算法处理这些实际结果,会得到不同的评估结果。文献[3]也使用了组合算法。但它只是将其应用于合并 X-MIDD 的过程中。X-MIDD 形成后,就没有组合算法的任何信息了。事实上,子策略集、子策略等都可能定义自己的组合算法。例如,如前所述,XACML 就支持这样的定义。既然 X-MIDD 无法表示组合算法,在决策过程中当然不能使用了。该修改则可以表示并直接支持多种组合算法的使用。

第三,分析和讨论基于一般 ABAC 模型,而不只限于 XACML,因而适用性更好。

### 2.2 使　用

可以使用改进的决策图对式(1)形式的访问请求进行是否匹配策略要求的评估及是否批准的决策评估。

在下面讨论中,假定 target 元素用 iMIDD 表示,策略用 iX-MIDD 表示。

#### 2.2.1 iMIDD 评估

根据 iMIDD 评估请求 $X$ 的处理从 iMIDD 的根开始,主要规则如下:

1)对于每个内结点($x_i$,$C$),如果请求 $X$ 的 $x_i$ 值处于数组元素($ed$,$c$)的边 $ed$ 的区间划分 $p$ 中,即 $x_i \in ed.p$,则选择 $ed$ 的下结点 $c$。

2)评估过程中,如果 $\nexists x_i \in x$,则返回当前结点的状态。该状态由该结点的数组 $C$ 确定:如果 $C$ 中各元素的 $ed$ 中,没有一个 $ed$ 的状态是 IN,则返回 F;否则,返回 IN。

3)如果达到 T - 叶结点,则返回 T。

#### 2.2.2 iX-MIDD 评估

根据 iX-MIDD 对请求 X 进行判断,决定是否准许。处理从 iX-MIDD 的根开始,主要规则如下:

1)对于每个内结点($x_i$,$C$),如果请求 $X$ 的 $x_i$ 值在数组元素($ed$,$c$)的边 $ed$ 的区间划分 $p$ 中,即 $x_i \in ed.p$,则选择 $ed$ 的下结点 $c$。

2)评估过程中,如果 $\nexists x_i \in x$,,则返回($s$,∅,∅)。其中,$s$ 由当前结点的数组 $C$ 确定:如果 $C$ 中各元素的 $ed$ 中没有一个 $ed.s$ 是 N,则 $s = \mathrm{IN}_e$,此处 $e$ 等于外结点($e$,$ca$,$O$)中的 $e$,即 $e$ 无法确定;否则,$s$ = N,即不可用。

3)如果达到外结点,则返回外结点($e$,$ca$,$O$)。

#### 2.2.3 iMIDD 转换为 iX-MIDD

为了进行规则评估,需要将 iMIDD 转换为 iX-MIDD。转换步骤如下:

第 1 步:利用区间的并操作 ∧(intersect)[3],合并规则中全部元素的 iMIDD。

第 2 步:用($e$,$ca$,$O$)代替 T - 叶结点中的 T,将 iMIDD 的 T - 叶结点改为决策 - 叶结点。其中,$e \in$ {P,D},是规则的结果;$ca$ 是规则或策略组合算法;$O$ 是义务 - 忠告列表,可以缺少(为空)。

第 3 步:对于 iMIDD 的每个内结点 $m := (x, C)$:

1)将 $C$ 中 $ed.s$ 从{F,IN}转换到 $V_R$。

① 如果 $ed.s$ = F,则转换成新状态值 $N$。

② 如果 $ed.s$ = IN,则转换成新状态值 $\mathrm{IN}_e$,此处 $e$ 等于决策 - 叶结点的 $e$。

2)增加 $O$。如果 $C$ 中各元素的 $ed$ 中至少有一个 $ed.s \neq$ N,$O$ 同决策 - 外结点的 $O$;否则 $O$ 为空。

策略管理的其他应用,如策略测试,策略比较,逆请求等,iX-MIDD 也完全可以胜任。限于篇幅,就不在这里介绍了。

### 2.3 复杂度及性能分析

文献[3]指出,即使在最坏情况下,X-MIDD 空间复杂度也与策略的规模,策略树的高度,以及对象表达逻辑式的复杂度均无关。上述结论,完全适合于 iX-MIDD,因为 iX-MIDD 并未改变 X-MIDD 的基本结构。

当然,由于 iX-MIDD 的修改,需要存储是否是重要属性的信息,因而要略微增加存储空间。假定 X-MIDD 的结点数(不包括叶结点)为 $n$,边数为 $k$,叶结点数为 $m$,iX-MIDD 比 X-MIDD 增加的存储单位约等于 $k - n + m$。

X-MIDD 时间复杂度的分析及结论也完全适用

于iX-MIDD,就不细述了。至于具体的运算时间增加,基本上可以忽略,因为后者只是改变了获取重要属性信息的位置。

鉴于iX-MIDD复杂度与X-MIDD几乎没有区别,可以预计,其性能与后者差不多。

为了对比,进行模拟实验。实验使用第2.1节策略例,其X-MIDD和iX-MIDD各有13个结点和26个简约区间划分。设计的访问请求案例24个,覆盖了全部的结点和区间划分,以及全部可能的路径。选择访问请求决策处理做实验,因为它是安全策略管理中最重要和每次访问必需的应用。鉴于Windows系统是事件驱动、而非实时操作系统,为尽量减少Windows系统后台程序产生的计时精度,访问请求决策对比实验,每次进行$5\times10^4$次,重复20遍。

实验环境是:计算机型号Dell N5110,处理器Intel(R) Core(TM)i7-2670QM CPU @2.20 GHz,内存8.00 GB,操作系统64位Win 7家庭普通版,Service Park 1。实验结果见表2。

**表2 访问请求决策实验结果**

**Tab.2 Simulation of access request decision**

| 项目 | 决策图基本信息存储量/Byte | 24案例$5\times10^4$次决策用时/ms |
|---|---|---|
| X-MIDD | 496 | 1 299 |
| iX-MIDD | 564 | 1 367 |

iX-MIDD图结构基本信息所需存储量比X-MIDD略大,因为需要在每个简约区间划分上存储是否是重要属性的信息,以及叶结点上存储组合算法信息。iX-MIDD方法5万次决策用时略多,则是因为有3个简约区间划分是重要属性,而示例策略的X-MIDD没有重要属性了,因此,前者略为增加了一点处理时间。模拟实验中,一个访问请求案例增加时间(单位:μs):

$(1\ 367-1\ 299)\cdot1\ 000/(24\cdot50\ 000)\approx0.057$,

不到1 μs。可见其性能与原方法差不多。

实验结果验证了前面的分析结论,即改进使存储量略有增加,操作步骤也略有增加。因为操作步骤的增加,略为增加了一点处理时间,但两者性能相当。

## 3 结 论

在安全策略的实际应用中,策略评估十分重要,特别是策略数量很大时,性能问题特别突出。同已有研究相比,MIDD和X-MIDD有很大的改进。但是,这种方法也有缺陷,它基于XACML,无法很好地标注在决策中有关键作用的重要属性,特别是有关元素的子MIDD(及X-MIDD)合并时,容易丢失重要属性标记,也不方便使用组合算法。

针对MIDD和X-MIDD方法的不足,首先基于一般ABAC模型,而不只限于XACML,分析讨论更一般化,扩大了适用范围。其次,设计了新的图结构,将图的边扩展为由上结点变量值范围(区间划分)及状态组成的元组,可比MIDD更好地标注在决策中有关键作用的重要属性。有关属性的子iMIDD合并时,不会丢失重要属性标识,有利于决策过程中更精细化处理。改进的X-MIDD中,除了边的扩展有利于保持重要属性标识外,决策-叶结点也做了扩展,增加了组合算法信息,可保证在决策过程中应用组合算法,对访问请求做出正确判决。给出了应用本文方法进行策略元素匹配、策略评估,以及从iMIDD生成iX-MIDD的流程。最后分析了本文方法的时间、空间复杂度。分析和模拟实验均表明,新方法复杂度与MIDD方法相当,性能也与文献[3]方法相当。新方法完全能用于策略评估、策略测试、策略比较和逆请求等多种策略管理应用。

iMIDD和iX-MIDD图中内部结点的安排次序会影响图结构,从而影响计算效能,这与MIDD图类似。作者计划进一步研究,希望能找出一些规律,以便应用于构建优化的iMIDD和iX-MIDD。策略冲突检测在策略管理中十分必要,也计划将本文方法用于它。另外,从应用的角度,开发工具包,以便用户使用,也是急待进行的。

**参考文献:**

[1]罗万伯,刘嘉勇,戴宗坤,等.信息安全应用基础[M].重庆:重庆大学出版社,2005.

[2]Rao P,Lin D,Bertino E,et al. Fine-grained integration of access control policies[J]. Computers and Security,2011,30(2/3):91-107.

[3]Ngo C,Demchenko Y,Laat C D. Decision diagrams for XACML policy evaluation and management[J]. Computers & Security,2015,49(5):1-16.

[4]Fisler K,Krishnamurthi S,Meyerovich L A,et al. Verification and change-impact analysis of access-control policies[C]//Proceedings of the 27th International Conference on Software Engineering. New York:ACM,2005:196-205.

(下转第210页)