基于内容和地点维度的机密信息降级策略

朱 浩[1,2] 庄 毅[1] 薛 羽[1] 丁卫平[1,2]
(南京航空航天大学计算机科学与技术学院 南京 210016)[1]
(南通大学计算机科学与技术学院 南通 226019)[2]

摘 要 目前机密信息降级策略的研究主要集中在信息降级的内容、地点、时间等维度上,每个维度的策略都有一定的局限性,攻击者将会利用其他维度的漏洞,非法获取额外的机密信息。降级策略需要综合考虑多个维度来确保机密信息的可信降级。为此,利用攻击者知识模型,提出了一种基于内容和地点维度的降级策略。内容维度的关键思想是攻击者不允许通过滥用降级机制来获取额外的机密信息,而地点维度控制机密信息仅能通过特定的语句进行降级。此外,建立了该策略实施的类型规则,并证明了类型规则的可靠性。

关键词 信息流控制,降级策略,机密性,无干扰
中图法分类号 TP311 **文献标识码** A

Declassification Policy Based on Content and Location Dimensions

ZHU Hao[1,2] ZHUANG Yi[1] XUE Yu[1] DING Wei-ping[1,2]
(School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)[1]
(School of Computer Science and Technology, Nantong University, Nantong 226019, China)[2]

Abstract Current research on declassification policies mainly involves content, location, time and other dimensions, and each of them has some limitations. Attacker could learn more confidential information than intended by using the vulnerability of other dimensions. A synthesis of different dimensions in declassification policy would further improve assurance that confidential information is being declassified properly. This paper proposed a declassification policy based on the content and location dimensions, using attacker knowledge model. The key idea of content dimension of the policy is that attacker is not allowed to increase observations about confidential information by causing misuse of the declassification mechanism, and that location dimension of the policy controls confidential information is declassified only through the declassification statement. Additionally, we established type rules of policy enforcement and proved its soundness.

Keywords Information-flow controls, Declassification policy, Confidentiality, Non-interference

机密性是软件的可信性属性之一。保护机密性的常用方法是访问控制和加密技术,但是它们都不能完全确保端到端的机密性[9]。为了克服上述问题,近年来,研究人员广泛采用信息流控制策略来研究机密性问题[11,12]。在基于语言的信息流控制策略中应用最广泛的是无干扰[2]策略:低机密性的输出不依赖于高机密性的输入,使得攻击者无法从低机密性的输出中推导出高机密性的输入。但是,无干扰的限制性太强,很多实用软件由于功能的要求,不可避免地需要违反无干扰策略。例如,登录口令检查程序在拒绝错误口令登录时,却泄漏了关于口令的部分机密信息(程序的拒绝登录行为排除了一个错误的口令,缩小了攻击者的口令搜索空间)。针对这种需求,需要放松无干扰的限制性,机密信息降级策略是一种放松的无干扰策略。它的基本思想是允许程序由于功能需要而将高机密性信息降级为低机密性信息,但是需要通过制定和实施相关的安全策略,对程序中高机密性信息的降级加以分析和控制,以防止恶意代码非法降级高机密性信息(该攻击也称为信息清洗攻击)。

目前机密信息降级策略的研究主要集中在信息降级的内容、地点、时间等维度上[3-7]。本文主要关注内容和地点维度的策略。文献[5]提出的"渐进释放"(gradual release)是一个基于地点维度的降级策略,它限定高机密信息只能通过特定的降级语句来释放到低机密性变量,但是没有对降级的内容进行限定,那么攻击者可能利用机密信息的降级通道,在程序中正确的降级地点泄漏额外的高机密信息。文献[4]提出的"定界释放"(delimited release)是一个基于内容维度的降级策略,它对降级语句中降级表达式内容做了限定,使得攻击者不能通过程序运行的最终存储状态推导出额外的高机密信息。但是它没有对除了降级语句以外的高机密信息的降级进行限

到稿日期:2011-09-18 返修日期:2011-11-25 本文受航空基金(2010ZC13012),江苏省普通高校研究生科研创新计划项目(CXLX11_0205)资助。

朱 浩(1977-),男,博士生,讲师,CCF会员,主要研究方向为信息安全、智能计算,E-mail:searain@ntu.edu.cn;庄 毅(1956-),女,教授,主要研究方向为信息安全、分布式计算;薛 羽(1981-),男,博士生,主要研究方向为智能计算、信息安全;丁卫平(1979-),男,博士生,讲师,主要研究方向为智能计算。

制，攻击者就可能在除了降级语句以外的其他程序点降级高机密性信息；而且“定界释放”策略的控制粒度过于粗糙，只关注程序运行的最终存储状态，对程序运行过程中的各个存储状态没有做限定，这就可能导致高机密信息在中间存储中泄漏。基于以上两个降级策略各自的优缺点，本文提出一种结合地点和内容维度的“渐进定界”策略：首先限定高机密信息的降级只能通过降级语句来完成，在程序的其他任何地点都不能存在高机密信息的降级；其次在程序中的每一个降级语句点，采用“定界释放”的策略来确保每一个降级点不存在超过必要降级内容限度的额外高机密性信息的降级。

本文第1节描述了程序语言模型；第2节阐述了攻击者的知识模型；在此基础上，第3节提出了结合内容和地点维度的机密信息降级策略，给出了策略实施的类型规则，并进行了可靠性证明和约束性分析；最后总结全文。

1 语言模型

1.1 安全格与信息流

机密信息降级策略所研究的程序语言是一种具有安全类型的语言，其中每个常量和变量都具有一个安全级别，本文分低机密性 L 和高机密性 H 两级。对攻击者而言，低机密性 L 的数据是完全公开的，而高机密性 H 的数据是保密的。令常量的机密性级别恒为 L，变量的机密性级别可通过机密性环境 Γ 获得（Γ 是一个从变量到其机密性级别的映射）。令集合 $SC=\{L,H\}$，SC 中偏序关系 $\sqsubseteq$ 表示机密性级别的高低，$L\sqsubseteq H$。$(SC,\sqsubseteq)$ 构成一个安全格，$(SC,\sqcup,\sqcap)$ 是格 $(SC,\sqsubseteq)$ 诱导的代数系统，其中，$\sqcup$ 和 $\sqcap$ 是定义在集合 SC 上的两个二元运算：$\forall \sigma_1,\sigma_2\in SC$，$\sigma_1\sqcup\sigma_2$ 表示 σ_1 与 σ_2 的最小上界，$\sigma_1\sqcap\sigma_2$ 表示 σ_1 与 σ_2 的最大下界。例如，表达式 $x+y$ 的机密性级别为 $\Gamma(x)\sqcup\Gamma(y)$。

程序中的信息流可分为显式流和隐式流[1]。显式流是由赋值语句引起的，比如 $x:=y$ 语句，变量 y 的信息直接流入了变量 x。语句 $x:=y$ 中从变量 y 到 x 的显式流是合法的，当且仅当 $\Gamma(y)\sqsubseteq\Gamma(x)$ 成立。隐式流是由程序的控制结构引起的，比如 if b then $x:=1$ else $x:=0$，条件表达式 b 的信息间接地流入选择语句的被赋值变量 x 中。循环结构也存在类似的隐式流。语句 if b then $x:=1$ else $x:=0$ 中的隐式流是合法的，当且仅当 $\Gamma(b)\sqsubseteq\Gamma(x)$ 成立；语句 if $x\geqslant y$ then $z:=w$ else $i:=i+1$ 中的隐式流是合法的，当且仅当 $\Gamma(x)\sqcup\Gamma(y)\sqsubseteq\Gamma(z)\sqcap\Gamma(i)$。

1.2 语法和语义

本文采用具有安全类型的确定性顺序式语言，即 While 语言。它由表达式 e 和命令 c 组成，用巴科斯范式描述如下：

$$e::=n \mid v \mid e_1\ \mathrm{op}\ e_2$$

$$c::=\mathrm{skip} \mid v:=e \mid c_1;c_2 \mid \mathrm{if}\ e\ \mathrm{then}\ c_1\ \mathrm{else}\ c_2 \mid \mathrm{while}\ e\ \mathrm{do}\ c \mid v:=\mathrm{declassify}(e)$$

其中，n 是常量，v 是变量，$\mathrm{Vars}(e)$ 表示表达式 e 中变量的集合，op 是算术或逻辑运算符；$v:=\mathrm{declassify}(e)$ 是机密信息的降级语句，它的功能是将表达式 e 的高机密性级别 H 降低到低机密性级别 L，并赋值给变量 v，其中 $\Gamma(v)=L$，e 称为降级表达式。

程序语言的结构化操作语义采用转移系统来描述。配置（Configuration）具有两种形式：①$\langle m,c\rangle$，表示命令 c 将在存储 m（存储是变量到其取值的映射，可扩展为对一个表达式求值）上开始执行；②$\langle m,stop\rangle$，表示命令执行完的终止配置状态。结构化操作语义中的转移关系是小步转移，可描述为 $\langle m,c\rangle\rightarrow_t\langle m',c'\rangle$ 或 $\langle m,c\rangle\rightarrow_t\langle m',stop\rangle$，表示命令 c 在存储 m 上单步执行后到达配置 $\langle m',c'\rangle$ 或命令 c 执行完毕，到达终止配置 $\langle m',stop\rangle$，其中 t 是转移事件，表示一个赋值操作，对变量 x 赋值为 n 的事件记作 (x,n)。如果转移关系不影响存储状态，那么转移事件可以记作 $\in$ 或省略。用 $\overset{*}{\rightarrow_{\vec{t}}}$ 表示 $\rightarrow_t$ 的自反传递闭包，其中，$\vec{t}$ 是转移事件序列，$\langle m,c\rangle\overset{*}{\rightarrow_{\vec{t}}}\gamma$ 表示从 $\langle m,c\rangle$ 经过零步或多步转移后到达配置 γ，当不考虑配置 γ 时，可记作 $\langle m,c\rangle\overset{*}{\rightarrow_{\vec{t}}}$。此外，$\langle m,c\rangle\Downarrow m'$ 表示 $\langle m,c\rangle\overset{*}{\rightarrow_{\vec{t}}}\langle m',stop\rangle$。

2 攻击者知识

在攻击者知识[10]的表示中，有两个关键元素：存储的 L 投影和可观察事件。

存储 m 的 L 投影表示将存储 m 限制在仅对低机密性变量进行映射，记为 m_L。如果所有的低机密性变量在存储 m_1 和 m_2 映射下变量值相等，称存储 m_1 和 m_2 的 L 投影相等，记为 $m_{1L}=m_{2L}$ 或 $m_1=_L m_2$；形式上，$m_{1L}=m_{2L}\Leftrightarrow\forall v.\ \Gamma(v)=L\Rightarrow m_1(v)=m_2(v)$。

对于攻击者而言，对低机密性变量的赋值或命令的执行终止称为可观察事件；一个特殊的可观察事件是 $v:=\mathrm{declassify}(e)$，其称为降级事件。用 ℓ 表示可观察事件，用 $\vec{\ell}$ 表示可观察事件序列，用 $\vec{\ell}_n=\ell_1\cdots\ell_n$ 表示序列长度为 n 的可观察事件序列。不可观察事件是指对高机密性变量的赋值。假设程序执行产生的转移事件序列为 $\vec{t}$，其中包括了可观察事件和不可观察事件，依次提取 $\vec{t}$ 中的所有可观察事件而形成的可观察事件序列，记为 $\vec{t}_L$。在策略的研究中，我们更关注转移关系中的可观察事件序列，常用转移关系 $\langle m,c\rangle\overset{*}{\rightarrow_{\vec{\ell}}}$ 表示 $\langle m,c\rangle\overset{*}{\rightarrow_{\vec{t}}}\wedge\vec{t}_L=\vec{\ell}$。

攻击者根据程序运行中可观察事件序列而推导出可能产生这些事件的初始存储的集合，称该集合为攻击者知识。攻击者知识包含的元素越多，攻击者对机密信息的不确定性就越大；攻击者根据程序执行中新产生的可观察事件，将不断排除不满足条件的存储，攻击者知识中包含的元素个数将不断减少，该性质称为知识的单调性。

定义 1 假设程序 c 从初始存储 m 开始运行，产生可观察事件序列 $\vec{\ell}$，则攻击者知识 $k(m_L,c,\vec{\ell})$ 为：

$$k(m_L,c,\vec{\ell})=\{m' \mid m_L'=m_L\wedge(\langle m',c\rangle\overset{*}{\rightarrow_{\vec{t}}}\langle m'',stop\rangle\vee\langle m',c\rangle\overset{*}{\rightarrow_{\vec{t}}}\cdot_{\ell})\}$$

在定义 1 中，程序 c 运行产生的可观察事件序列 $\vec{t}$ 分两种情况：①可观察事件序列 $\vec{\ell}$ 后面不再产生可观察事件，程序运行最终能到达终止状态；②可观察事件序列 $\vec{\ell}$ 后紧跟着其他可观察的事件。由此可见，定义 1 的攻击者知识是终止不敏感的，即忽略了是否进入死循环所引发的攻击者知识的变化。例如，程序 c：

$l:=0$；while $h=3$ do skip；$l:=6$

其中 $\Gamma(l)=L,\Gamma(h)=H$。假设程序 c 的一次运行产生了可观察事件序列$(l,0)(l,6)$,根据定义 1 可得:

$k(m_L,c,(l,0))=k(m_L,c,(l,0)(l,6))$

说明攻击者知识 $k(m_L,c,(l,0))$在事件$(l,6)$发生后并没有发生变化,为 $h\neq 3$ 且与初始存储具有相同 L 投影的存储的集合。Askarov 等人对终止不敏感的安全性给出了下面的论断[8]:攻击者不可能在机密信息取值空间大小的多项式时间内获取到机密信息;并且,如果机密信息的取值均匀分布,那么在多项式时间内猜测出机密信息的概率是可以忽略的。基于此论断,本文仅考虑终止不敏感的情况。

另外,攻击者初始知识 $k(m_L,c)$表示能导致程序运行终止且与初始存储 m 有相同 L 投影的存储的集合,形式上有:

$$k(m_L,c)=\{m' \mid m_L'=m_L \wedge \exists m'',\vec{\ell}.\langle m',c\rangle \xrightarrow{*}_{\vec{\ell}} \langle m'',stop\rangle\}$$

3 降级策略

任何一种安全策略都与攻击者的能力假设有关。本文假定攻击者除了具有观察程序运行状态的能力外,还具有注入攻击代码而改变程序执行过程的能力,但限制攻击者注入的代码不能包含降级语句。如果可以注入降级语句,那么攻击者就可以直接获取任意的高机密性信息。该限制可以通过代码的完整性约束来实现[10],本文仅考虑机密性问题。

3.1 "渐进定界"策略

定义 2 可观察事件序列$\vec{\ell}_n=\ell_1\cdots\ell_n,n\geqslant 1$,其中 $\ell_{r_1}\cdots\ell_{r_k}$是$\vec{\ell}_n$ 中的降级事件序列,该序列中每个降级事件 ℓ_{r_i} 对应的降级表达式为 e_{r_i},其中 $1\leqslant i\leqslant k$,初始存储为 m,m_1,m_2,令 $m_1^{r_i}$ 和 $m_2^{r_i}$ 表示分别从初始存储 m_1,m_2 开始运行程序 c 且降级事件 l_{r_i} 发生时各自的存储状态,其中 $1\leqslant i\leqslant n$。程序 c 满足"渐进定界"策略,当且仅当同时满足下列两个安全条件:

Ⅰ)$\forall i,1\leqslant i\leqslant n,(\forall j,1\leqslant j\leqslant k,i\neq r_j)\Rightarrow k(m_L,c,\vec{\ell}_{i-1})=k(m_L,c,\vec{\ell}_i)$,其中 $k(m_L,c,\vec{\ell}_0)=k(m_L,c)$;

Ⅱ)$m_1=_L m_2\wedge\forall i,1\leqslant i\leqslant k,(\forall j,1\leqslant j\leqslant i,m_1(e_{r_j})=m_2(e_{r_j}))\Rightarrow m_1^{r_i}=_L m_2^{r_i}$。

安全条件Ⅰ)表示只有降级事件发生时才可能改变攻击者知识,从而限定了高机密性信息的降级只能在程序的降级语句中完成;安全条件Ⅱ)表示程序 c 分别在两个 L 投影相等的初始存储 m_1 和 m_2 上开始运行,如果程序中每个降级语句点之前的所有降级点的降级表达式值都是相等的,那么程序 c 执行到该降级语句点时存储的 L 投影依然是相等的,这个条件可防止在每个降级语句点泄漏超过期望降级内容的额外机密信息。举例说明"渐进定界"策略提供的安全性。考虑程序 c_1:

$l_1:=6;v:=\text{declassify}(h_1+h_2);l_2:=8;v:=h_1+h_2$

设变量 h_1 和 h_2 是高机密性变量,且 $m(h_1)=7,m(h_2)=9$,变量 l_1,l_2 和 v 是低机密性变量。程序 c_1 运行将产生可观察事件序列$(l_1,6)(v,16)(l_2,8)(v,16)$。当第一个可观察事件$(l_1,6)$产生时,攻击者知识 $k(m_L,c,(l_1,6))$是与初始存储 m 具有相同的 L 投影的存储集合;当第二个可观察事件$(v,16)$产生时,攻击者知识 $k(m_L,c,(l_1,6)(v,16))$是 $h_1+h_2=16$ 且与初始存储 m 具有相同的 L 投影的存储的集合,显然有 $k(m_L,c,(l_1,6))\supseteq k(m_L,c,(l_1,6)(v,16))$;当第三、第四个可观察事件$(l_2,8)(v,16)$产生时,攻击者没有得到新的高机密性信息,攻击者知识没有发生变化。由此可知,程序 c_1 满足定义 2 的安全条件Ⅰ),另外,通过分析不难得出定义 2 的安全条件Ⅱ)也是满足的。综上,程序 c_1 是满足"渐进定界"策略的。下面考虑程序 c_1 的变体 c_1':

$l_1:=6;v:=h_1+h_2;l_2:=8;v:=\text{declassify}(h_1+h_2)$

该程序中,h_1+h_2 的值在降级语句 $v:=\text{declassify}(h_1+h_2)$执行之前就通过赋值语句 $v:=h_1+h_2$ 泄漏(非法降级)到了低机密性变量 v 中。当赋值语句 $v:=h_1+h_2$ 执行时产生的可观察事件$(v,16)$将引起攻击者知识的更新,但是 $v:=h_1+h_2$ 不是合法的降级语句,根据安全条件Ⅰ),它的执行是不允许引起攻击者知识更新的,从而可得程序 c_1'不满足"渐进定界"策略的条件Ⅰ),因此,"渐进定界"策略认为程序 c_1'是不安全的。但是"定界释放"策略[4]判定该程序是安全的。在实际应用中,如果不考虑高机密信息的降级地点是非常危险的,比如在信息购买协议中,用户可能会通过信息清洗攻击不需要付费而提前获得所需要的信息。

考虑另外一个例子。假设公司中有 n 个员工,每个员工的工资用高机密性变量 h_i 表示,其中 $1\leqslant i\leqslant n$。现在要求把 n 个员工的平均工资释放到低机密性变量 avg 中,除此之外,不能泄漏更多的关于员工工资的信息。假设 $n=4$,则求平均工资的语句 c_2:

$avg:=\text{declassify}((h_1+h_2+h_3+h_4)/4)$

一种针对求平均工资的信息清洗攻击 c_2-*attack*:

$h_2:=h_1;h_3:=h_1;h_4:=h_1;$

$avg:=\text{declassify}((h_1+h_2+h_3+h_4)/4)$

显然该程序将员工工资 h_1 完全泄漏给了 avg,用类似的方法可获取其他员工的工资信息。假设初始存储 m_1 和 m_2 满足 $m_1=_L m_2,m_1(h_1)=m_2(h_2)=2,m_1(h_2)=m_2(h_1)=3$,并且 $\forall i\in\{3,4\},m_1(h_i)=m_2(h_i)=0$,则有:

$$m_1((h_1+h_2+h_3+h_4)/4)=m_2((h_1+h_2+h_3+h_4)/4)=5/4$$

程序 c_2-*attack* 分别从初始存储 m_1 和 m_2 开始执行,当降级语句执行完时,存储状态分别是 m_1'和 m_2',则 $m_1'(avg)=2$和 $m_2'(avg)=3$,不满足定义 2 的安全条件Ⅱ),从而"渐进定界"策略判定程序 c_2-*attack* 是不安全的。但是"渐进释放"策略[5]判定该程序是安全的。考虑 c_2-*attack* 的变体 c_2-*attack*$'$:

$h_2:=h_1;h_3:=h_1;h_4:=h_1;$

$avg:=\text{declassify}((h_1+h_2+h_3+h_4)/4);avg:=0$

采用与程序 c_2-*attack* 同样的分析可知,"渐进定界"策略判定程序 c_2-*attack*$'$是不安全的,但是"渐进释放"和"定界释放"都判定该程序是安全的。由上述例子分析可见,"渐进定界"策略在结合"渐进释放"策略的地点维度和"定界释放"策略的内容维度的同时,对策略的控制粒度进行了细化,能判别出"渐进释放"和"定界释放"策略都不能识别的攻击。图 1—图 3 是这 3 个策略的示意图。

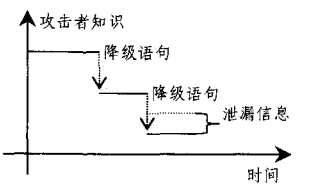

图 1 渐进释放策略

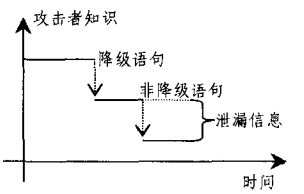

图 2 定界释放策略

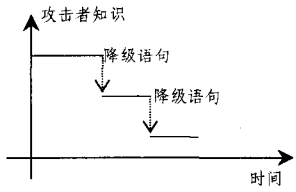

图 3　渐进定界释放策略

3.2　类型规则

本节给出了一个实施“渐进定界”策略的类型规则，如图 4 所示。表达式类型规则的一般形式为 $\Gamma \vdash e:\ell$，它的含义是在机密性环境 Γ 下，e 是机密性级别为 ℓ 的良类型(well-typed)表达式。命令类型规则的一般形式为 $\Gamma, pc \vdash c:U, D$，它表示在机密性环境和程序计数器标签 pc 下命令 c 是良类型，c 的执行产生了效果 U 和 D，分别表示在 c 执行到当前程序点时所有被更新的变量的集合和所有降级表达式中的变量的集合。为了跟踪程序中的隐式流，引入了程序计数器标签[9](program counter label)，简称 pc，它的取值是一个机密性等级，表示所有影响当前程序点的隐式流机密性级别的最小上界。比如程序：

…;if h then $x:=1$ else $x:=0$;…

其中 if 语句条件表达式 h 的信息隐式地流入了各个分支命令($x:=1$ 或 $x:=0$)，所以各个分支命令的 pc 值至少为 $\Gamma(h)$，但是可能存在 if 语句的嵌套或循环语句中嵌套 if 语句。那么流入最内层 if 语句各个分支的隐式流的数据源有多个，分别是嵌套的 if 语句或循环语句的各个条件表达式。最内层的 if 语句各个分支命令的 pc 值是这些条件表达式的机密性级别的最小上界。上述程序中，if 语句各个分支的 pc 值是 if 语句的 pc 值与 if 条件表达式 h 的最小上界。如果类型规则能应用到程序的每个表达式和命令上，那么类型规则判定该程序是良类型，即是安全的程序。下面对规则作较详细的阐述。

规则 T-CO 表示常量的机密性级别是低机密性级别 L；规则 T-VA 表示变量 v 的机密性级别在机密性环境下是 $\Gamma(v)$；规则 T-OP 表示表达式 e_1 和 e_2 的 op 运算值的机密性级别是它们各自机密性级别的最小上界。

规则 T-SK 表示 skip 命令是良类型，它的执行不修改效果集合 U 和 D。

规则 T-AS 表示赋值语句右侧表达式 e 的机密性级别和程序计数器标签的最小上界必须不高于赋值语句左侧的变量 v 的机密性级别(目的是防止隐式的和显式的非法信息流)，才能保证赋值语句 $v:=e$ 是良类型；另外，由于变量 v 赋值而更新了效果集合 U。

规则 T-SE 表示当顺序语句 $c_1;c_2$ 中的两个命令都是良类型，并且命令 c_2 中的所有降级表达式的变量在命令 c_1 中没有被更新时，顺序命令 $c_1;c_2$ 是良类型，顺序语句的执行将引起命令 c_1 和 c_2 相应效果集合的叠加。

规则 T-IF 表示当 IF 选择语句的各个分支命令在机密性环境和程序计数器标签值 $\ell \sqcup pc$ 下是良类型(防止由于选择语句的嵌套、循环语句嵌套选择语句所引发的隐式非法信息流)时，该选择语句是良类型。由于类型规则需要考虑所有可能的执行路径，因此选择语句的效果集合是各个分支命令相应效果集合的叠加。

规则 T-WH 表示当 while 循环语句的循环体在机密性环境和程序计数器标签值 $\ell \sqcup pc$ 下是良类型，并且循环体内所有降级表达式中的变量没有在循环体内被更新(循环体 c 的循环执行可看成若干次顺序命令 $c;c$ 的形式)时，该循环语句是良类型。

规则 T-DE 表示降级语句执行时程序计数器标签必须是低机密性 L(防止非法的隐式信息流)，并且降级后赋值给低机密性变量，才能保证降级语句是良类型，该语句的执行将引起效果集合的更新。

$$\Gamma \vdash n:L \qquad \text{(T-CO)}$$

$$\Gamma \vdash v:\Gamma(v) \qquad \text{(T-VA)}$$

$$\frac{\Gamma \vdash e_1:\ell_1 \quad \Gamma \vdash e_2:\ell_2}{\Gamma \vdash e_1 \text{ op } e_2:\ell_1 \sqcup \ell_2} \qquad \text{(T-OP)}$$

$$\Gamma, pc \vdash \text{skip}:\emptyset,\emptyset \qquad \text{(T-SK)}$$

$$\frac{\Gamma \vdash e,\ell \quad \ell \sqcup pc \sqsubseteq \Gamma(v)}{\Gamma, pc \vdash v:=e:\{v\},\emptyset} \qquad \text{(T-AS)}$$

$$\frac{\Gamma, pc \vdash c_i:U_i, D_i \quad i=1,2 \quad U_1 \cap D_2=\emptyset}{\Gamma, pc \vdash c_1;c_2:U_1 \cup U_2,, D_1 \cup D_2} \qquad \text{(T-SE)}$$

$$\frac{\Gamma \vdash e:\ell \quad \Gamma, \ell \sqcup pc \vdash c_i:U_i, D_i \quad i=1,2}{\Gamma, pc \vdash \text{if } e \text{ then } c_1 \text{ else } c_2:U_1 \cup U_2, D_1 \cap D_2} \qquad \text{(T-IF)}$$

$$\frac{\Gamma \vdash e:\ell \quad \Gamma, \ell \sqcup pc \vdash c:U_1, D_1 \quad U_1 \cap D_1=\emptyset}{\Gamma, pc \vdash \text{wihile } e \text{ do } c:U_1, D_1} \qquad \text{(T-WH)}$$

$$\frac{pc=\Gamma(v)=L}{\Gamma, pc \vdash v:=\text{declassify}(e):\{v\}, Vars(e)} \qquad \text{(T-DE)}$$

图 4　While 语言的类型规则

3.3　类型规则的可靠性

定理 1(可靠性定理)　如果根据上述类型规则能推出程序 c 是良类型，即 $\Gamma, pc \vdash C:U, D$，则程序 c 满足“渐进定界”释放策略。

证明：对 $\Gamma, pc \vdash c:U, D$ 导出结构的深度作归纳证明。

1)(T-SK)规则。skip 命令的执行不会引起存储状态和攻击者知识的变化，因此满足“渐进定界”的两个安全条件。

2)(T-AS)规则。分两种情况：

①当 $\Gamma(v)=H$ 时，没有高机密性信息流入到低机密性变量 v 中，此时赋值命令的执行不会引起存储的 L 投影和攻击者知识的变化，因此满足“渐进定界”的两个安全条件。

②当 $\Gamma(v)=L$ 时，通过对规则前提条件 $\Gamma \vdash e:\ell$ 和 $\ell \sqcup pc \sqsubseteq \Gamma(v)$ 反向推导可得 $\ell=pc=L$，从而可推出赋值命令没有引起高机密性信息显式或隐式地流入到低机密性变量 v 中，从而攻击者知识没有变化，满足“渐进定界”的安全条件Ⅰ)。另外，本规则由于不涉及降级语句(降级语句由 T-DE 规则考虑)，因此显然满足“渐进定界”的安全条件Ⅱ)。

3)(T-SE)规则。假设顺序语句 $c_1;c_2$ 从初始存储 m 上执行产生可观察事件序列 $\vec{\ell}_n=\ell_1 \cdots \ell_n, n \geqslant 1$，其中 $\ell_{r_1} \cdots \ell_{r_k}$ 是 $\vec{\ell}_n$ 中的降级事件序列。分两步证明：

第一步，证明其满足安全条件Ⅰ)。

令 $L(m_L, c)=\{\vec{\ell} \mid \exists m' \cdot m_L=m_L' \wedge \langle m', c\rangle \xrightarrow{*}_{\vec{\ell}} \langle m'', stop\rangle\}$，则 $\vec{\ell}_n \in L(m_L, c_1;c_2)$，分两种情况考虑：

①当 $\vec{\ell}_n \in L(m_L, c_1)$ 时，说明 c_2 的执行没有产生可观察事件，那么 $c_1;c_2$ 执行时攻击者知识的变化仅仅与命令 c_1 有关，从而有：

$$k(m_L, c_1;c_2, \vec{\ell}_{i-1})=k(m_L, c_1, \vec{\ell}_{i-1}), 1 \leqslant i \leqslant n \qquad (1)$$

$$k(m_L, c_1;c_2, \vec{\ell}_i)=k(m_L, c_1, \vec{\ell}_i), 1 \leqslant i \leqslant n \qquad (2)$$

根据规则的前提条件 $\Gamma, pc \vdash c_i: U_i, D_i, i=1,2$，对命令 c_1 应用归纳假设可得：

$$\forall i, 1 \leqslant i \leqslant n, (\forall j, 1 \leqslant j \leqslant k, i \neq r_j) \Rightarrow k(m_L, c_1, \vec{\ell}_{i-1}) = k(m_L, c_1, \vec{\ell}_i) \quad (3)$$

综合式(1)－式(3)，可得：

$$\forall i, 1 \leqslant i \leqslant n, (\forall j, 1 \leqslant j \leqslant k, i \neq r_j) \Rightarrow k(m_L, c_1; c_2, \vec{\ell}_{i-1}) = k(m_L, c_1; c_2, \vec{\ell}_i)$$

从而满足“渐进定界”策略的安全条件Ⅰ)。

②当 $\vec{\ell}_n \notin L(m_L, c_1)$ 时，$\vec{\ell}_n$ 可看成两个可观察事件序列 $\vec{\ell}_a$ 和 $\vec{\ell}_b$ 的顺序组合：$\vec{\ell}_n = \vec{\ell}_a; \vec{\ell}_b$，其中 $\vec{\ell}_a = \ell_1 \cdots \ell_a$，$\vec{\ell}_b = \ell_{a+1} \cdots \ell_n$ 并且 $\exists m_1, m_2, m_3, m_1 =_L m, \langle m_1, c_1\rangle \xrightarrow{*}_{\vec{\ell}_a} \langle m_2, \text{stop}\rangle \wedge \langle m_2, c_2\rangle \xrightarrow{*}_{\vec{\ell}_b} \langle m_3, \text{stop}\rangle$。根据规则的前提条件 $\Gamma, pc \vdash c_i: U_i, D_i, i=1,2$，对命令 c_1 应用归纳假设可得，安全条件Ⅰ)在序列 $\vec{\ell}_a$ 部分是成立的。下面仅考虑 $\vec{\ell}_b$ 部分：令 $k^*(m_L, c, \vec{\ell}) = \{(m_1, m_2) \mid m_L = m_{1L} \wedge \langle m_1, c\rangle \xrightarrow{*}_{\vec{\ell}} \langle m_2, \text{stop}\rangle\}$，根据攻击者知识的定义，可推出：

$$\forall i, a+1 \leqslant i \leqslant n, k(m_L, c_1; c_2, \vec{\ell}_i) = \{m_1 \mid (m_1, m_2) \in k^*(m_L, c_1, \vec{\ell}_a) \wedge m_2 \in k(m_{2L}, c_2, \vec{\ell}_b)\} \quad (4)$$

对 c_2 应用归纳假设可得：

$$\forall i, a+1 \leqslant i \leqslant n, (\forall j, a+1 \leqslant j \leqslant k, i \neq r_j) \Rightarrow k(m_{2L}, c_2, \vec{\ell}_i) = k(m_{2L}, c_2, \vec{\ell}_{i-1}) \quad (5)$$

由式(5)，显然可得：

$$k(m_{2L}, c_2, \vec{\ell}_b) = k(m_{2L}, c_2, \vec{\ell}_{b-1}) \quad (6)$$

由式(4)和式(6)得：

$$\forall i, a+1 \leqslant i \leqslant n, (\forall j, a+1 \leqslant j \leqslant k, i \neq r_j) \Rightarrow k(m_L, c_1; c_2, \vec{\ell}_i) = \{m_1 \mid (m_1, m_2) \in k^*(m_L, c_1, \vec{\ell}_a) \wedge m_2 \in k(m_{2L}, c_2, \vec{\ell}_{b-1})\} = k(m_L, c_1; c_2, \vec{\ell}_{i-1})$$

第二步，证明其满足安全条件Ⅱ)。

令 $E(c)$ 表示命令 c 产生的降级事件，分两种情况：

①当 $\ell_{r_k} \in E(c_1)$ 时，顺序语句 $c_1; c_2$ 产生的降级事件 $\ell_{r_1} \cdots \ell_{r_k}$ 都由命令 c_1 产生，根据前提条件对命令 c_1 实施归纳假设即可推出其满足“渐进定界”策略的安全条件Ⅱ)。

②当 $\ell_{r_k} \notin E(c_1)$ 时，在顺序语句 $c_1; c_2$ 产生的降级事件 $\ell_{r_1} \cdots \ell_{r_k}$ 中，假设从 ℓ_{r_1} 到 ℓ_{r_a} 的降级事件序列属于集合 $E(c_1)$，从 $\ell_{r_{a+1}}$ 到 ℓ_{r_k} 的降级事件序列属于集合 $E(c_2)$，根据前提条件对命令 c_1 进行归纳假设，可得：

$$m_1 =_L m_2 \wedge \forall i, 1 \leqslant i \leqslant a (\forall j, 1 \leqslant j \leqslant i, m_1(e_{r_j}) = m_2(e_{r_j})) \Rightarrow m_1^{r_i} =_L m_2^{r_i} \quad (7)$$

令 $\langle m_1, c_1\rangle \Downarrow m_1'$，$\langle m_2, c_1\rangle \Downarrow m_2'$，设 $m_1 =_L m_2 \wedge \forall i, 1 \leqslant i \leqslant a, (\forall j, 1 \leqslant j \leqslant i, m_1(e_{r_j}) = m_2(e_{r_j}))$，根据式(7)可得，$m_1^{r_a} =_L m_2^{r_a}$，在命令 c_1 范围内，降级事件 ℓ_{r_a} 之后不存在降级事件，所以可推出 $m_1' =_L m_2'$。规则的前提条件($U_1 \cap D_2 = \emptyset$)确保了降级表达式 $e_{r_i}(a+1 \leqslant i \leqslant k)$ 中的变量 v 没有在命令 c_1 中被更新，即 $m_1(v) = m_1'(v)$，$m_2(v) = m_2'(v)$，从而可得：

$$\forall i, a+1 \leqslant i \leqslant k, m_1(e_{r_i}) = m_2(e_{r_i}) \Rightarrow m_1'(e_{r_i}) = m_2'(e_{r_i})$$

从而根据规则的前提条件对命令 c_2 实施归纳假设可得：

$$m_1' =_L m_2' \wedge \forall i, a+1 \leqslant i \leqslant k (\forall j, a+1 \leqslant j \leqslant i, m_1'(e_{r_j}) = m_2'(e_{r_j})) \Rightarrow m_1^{r_i} =_L m_2^{r_i} \quad (8)$$

综合式(7)和式(8)可得，其满足“渐进定界”策略的安全条件Ⅱ)。

4)(T-IF)规则。分两种情况：

①当 $\Gamma \vdash e: L$ 时，如果 $m_L = m_L'$，则 $m(e) = m'(e)$，那么 IF 语句在所有 L 投影相同的初始存储上执行时将进入相同的选择分支 $c_i(i=1,2)$。由规则的前提条件 $\Gamma: \ell \sqcup pc \vdash c_i: U_i, D_i(i=1,2)$ 可确保 IF 语句满足“渐进定界”策略的两个安全条件。

②当 $\Gamma \vdash e: H$ 时，$\ell = \Gamma(e) = H$，则 $\ell \sqcup pc = H$，从而规则的前提条件具有形式 $\Gamma: H \vdash c_i: U_i, D_i(i=1,2)$。这确保了在各个分支 $c_i(i=1,2)$ 中没有可观察的事件序列发生，保证了攻击者的知识和存储状态在分支语句的执行过程中是不变的，满足了“渐进定界”策略的两个安全条件。

5)(T-WH)规则。分两种情况：

①当 $\Gamma \vdash e: L$ 时，如果 $m_L = m_L'$，则 $m(e) = m'(e)$。由此，当 while 循环语句在所有 L 投影相同的初始存储上执行时，根据表达式 e 的取值，要么执行相同的有限次数的循环体，要么跳过循环体执行终止(规则仅考虑终止不敏感的攻击者知识，所以忽略死循环的情况)。当循环体 c 被执行多次时，可将其看成有限次数的顺序命令 $c; c$ 的情况，规则的前提条件 $U_1 \cap D_1 = \emptyset$ 确保了循环体中降级表达式中的变量没有在上一轮循环体的执行中被更新，因此可归结为规则(T-SE)的证明。

②当 $\Gamma \vdash e: H$ 时，与规则(T-IF)对应情况类似。

6)(T-DE)规则。释放语句执行时，安全条件Ⅰ)所表示的蕴涵式的左侧为假，因此不要求其蕴涵式右侧为真，所以满足安全条件Ⅰ)。假设 m_1' 和 m_2' 是降级语句分别从初始存储 m_1 和 m_2 执行完时各自的存储状态，如果 $m_1 =_L m_2$ 并且 $m_1(e) = m_2(e)$，则释放语句执行完后 $m_1' =_L m_2'$ 显然成立，从而安全条件Ⅱ)成立。

3.4 类型规则的约束性

上节给出了类型规则的可靠性证明，即满足类型规则的程序一定满足“渐进定界”策略。但反之未必，即满足“渐进定界”策略的程序不一定满足类型规则。这个关系如图 5 所示。

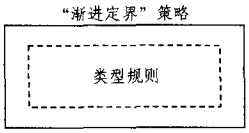

图 5 策略与规则的关系

例如，对于程序：

$t := h_1; h_2 := h_1; h_1 := t; avg := \text{declassify}((h_1 + h_2)/2)$

该程序满足“渐进定界”策略的两个安全条件，但是 3.2 节的类型规则判定该程序是不安全的。可见本文提出的类型规则虽然满足可靠性定理，但是规则的约束性比策略的约束性更强，因此提出更加宽容的类型规则是未来的工作之一。

结束语 机密信息降级策略是基于语言的信息流安全的关键挑战之一。本文利用攻击者知识模型提出了结合内容和地点维度的机密信息降级策略，其克服了单一的内容和地点维度策略的局限性，提高了降级策略抵抗信息清洗攻击的能力；该策略具有更细的控制粒度，能检测出单一的内容或地点

(下转第 185 页)

以看到，最优分数位 minwise 哈希的准确率和召回率在整数位之间，验证了此算法的有效性。

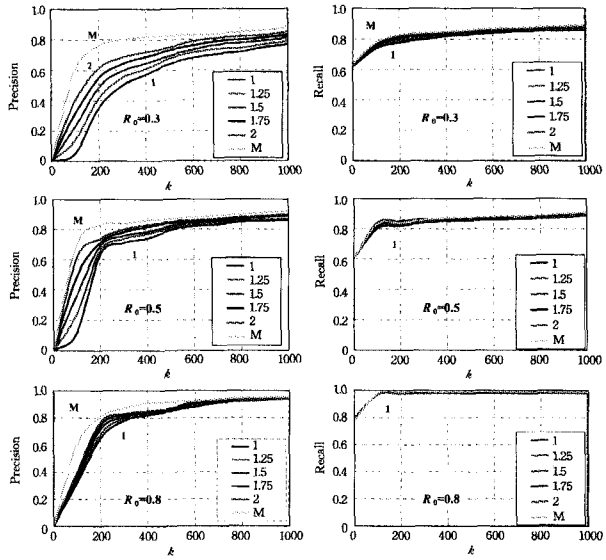

图 4　估值的准确率和召回率

结束语　min-wise 哈希算法广泛地应用于海量数据下的信息检索。b 位哈希算法将 $b=32$ 位缩小到 1 或者 2 位，降低了存储空间和计算时间。分数位 minwise 哈希算法对各种精度和存储空间需求有着更加广泛的可选择性。本文找到了构建分数位的最小方差组合，通过大量的基金项目数据对分数位估值的准确率和召回率进行实验分析，证明了最优分数位 minwise 哈希的准确率和召回率在整数位之间，验证了最优分数位 minwise 哈希算法的有效性。

参 考 文 献

[1] 鲍军鹏，沈钧毅，刘晓东，等. 自然语言文档复制检测研究综述[J]. 软件学报，2003，14(10)：1753-1760

[2] Broder A Z, Charikar M, Frieze A M, et al. Min-wise independent permutations[J]. Journal of Computer Systems and Sciences, 2000, 60(3): 630-659

[3] Broder A Z. On the resemblance and containment of documents [C] // Proceedings of Compression and Complexity of Sequences. Washington, DC, USA: IEEE Computer Society, 1997: 21-29

[4] Kalpakis K, Tang S. Collaborative data gathering in wireless sensor networks using measurement co-occurrence [J]. Computer Communications, 2008, 31(10): 1979-1992

[5] Dourisboure Y, Geraci F, Pellegrini M. Extraction and classification of dense implicit communities in the Web graph [J]. ACM Transactions on the Web(TWEB), 2009, 3(2): 1-36

[6] Bendersky M, Croft W B. Finding text reuse on the Web [C]// WSDM'09 Proceedings of the Second ACM International Conference on Web Search and Data Mining. New York, USA: ACM, 2009: 262-271

[7] Buehrer G, Chellapilla K. A scalable pattern mining approach to Web graph compression with communities [C] // WSDM '08 Proceedings of the international conference on Web search and Web data mining. New York, USA: ACM, 2008: 95-106

[8] Indyk P. A small approximately min-wise independent family of hash functions [J]. Journal of Algorithm, 2001, 38(1): 84-90

[9] Charikar M S. Similarity estimation techniques from rounding algorithms [C]// STOC'02 Proceedings of the thiry-fourth annual ACM symposium on Theory of computing. New York, USA: ACM, 2002: 380-388

[10] Li P, König A C. b-Bit minwise hashing [C]// WWW'10 Proceedings of the 19th international conference on World Wide Web. New York, USA: ACM, 2010: 671-680

[11] Li P, König A C. Theory and Applications of b-Bit Minwise Hashing [J]. Communications of the ACM, 2011, 54(8): 101-109

[12] Yuan X P, Long J, Zhang Z P, et al. f-Fractional Bit Minwise Hashing[J]. Journal of Software, 2012, 7(1): 228-236

(上接第 157 页)

维度的策略无法识别的攻击。另外，提出了策略实施的类型系统并进行了可靠性证明。

参 考 文 献

[1] Denning D E. A lattice model of secure information flow [J]. Communications of the ACM, 1976, 19(5): 236-243

[2] Goguen J A, Meseguer J. Security policies and security models [C]// IEEE Symposium on Security and Privacy. 1982: 11-20

[3] Sabelfeld A, Sands D. Declassification: dimensions and principles [J]. Journal of Computer Security, 2009, 17(5): 517-548

[4] Sabelfeld A, Myers A C. A model for delimited information release[J]. Software Security Theories and Systems, 2004, 3233: 174-191

[5] Askarov A, Sabelfeld A. Gradual Release: unifying declassification, encryption and key release policies[C]// IEEE Symposium on Security and Privacy. 2007: 207-221

[6] Lux A, Mantel H. Declassification with explicit reference points [C]// 14th European Symposium on Research in Computer Security. 2009: 69-85

[7] Lux A, Mantel H. Who can declassify? Formal Aspects in Security and Trust[J]. Lecture Notes in Computer Science, 2009, 5491: 35-49

[8] Askarov A, Hunt S, Sabelfeld A, et al. Termination insensitive noninterference leaks more than just a bit[C]// Computer Security-ESORICS. 2008: 333- 348

[9] Sabelfeld A, Myers A C. Language-based information flow security[J]. Selected Areas in Communications, 2003, 21(1): 5-19

[10] Askarov A, Myers A C. A semantic framework for declassification and endorsement [J]. Programming Languages and Systems, Lecture Notes in Computer Science, 2010, 6012: 64-84

[11] 唐和平，黄曙光，张亮. 动态信息流分析的漏洞利用检测系统[J]. 计算机科学，2010，37(7)：148-151

[12] 李伟楠，李翰超，石文昌. 基于信息流源的访问控制研究[J]. 计算机科学，2011，38(3)：34-39