RBAC模型的非法信息流控制*)

葛方斌[1] 杨 林[2] 王建新[2] 丁 利[3]
(解放军理工大学指挥自动化学院 南京 210007)[1] (中国电子系统工程研究所 北京 100039)[2]
(解放军理工大学通信工程学院 南京 210007)[3]

摘 要 由于缺乏信息流控制机制,RBAC模型的授权访问可能导致不安全的信息流.为了保护RBAC模型系统中信息的机密性,定义了RBAC模型的非法信息流概念,给出了非法信息流的检测、更新以及控制算法。将这些算法用于RBAC模型的授权管理可有效防止信息的非授权泄漏,实现安全的访问。

关键词 RBAC,非法信息流,引发用户,威胁用户

Illegal Information Flow Control on RBAC Model

GE Fang-bin[1] YANG Lin[2] WANG Jian-xin[2] DING Li[3]
(College of Command Automation, PLA University of Science and Technology, Nanjing 210007, China)[1]
(Institute of China Electronic System Engineering, Beijing 100039, China)[2]
(College of Communication Engineering, PLAUniversity of Science and Technology, Nanjing 210007, China)[3]

Abstract Due to lock the mechanism of controlling information flow, the authorized accesses of RBAC model may lead to insecure information flow. To protect confidentiality of information in RBAC model systems, the concept of illegal information flow is defined, algorithms of detecting, renewing and controlling illegal information flow are proposed. These algorithms can prevent effectively unauthorized information leakage, realize secure accesses.

Keywords RBAC, Illegal information flow, Users of inducing, Users of threatening

基于角色的访问控制[1-3](RBAC, Role-based Access Control)是目前广泛使用的一种访问控制技术。其基本思想是通过在用户和权限之间引入角色中介,用户通过分配适当的角色获得访问授权。实现了用户和权限的逻辑分离,大大简化了权限的管理,也方便了访问授权中最小权限原则的实施。

任何访问控制都体现了一定的安全策略,主体对客体的访问请求只有满足安全策略的约束才可获得请求的权限。然而,在一个缺乏信息流约束的访问控制模型系统中,根据策略授权的访问不一定总是安全的。原因在于授权访问引发的信息流可能导致不安全的系统状态。如果系统中主体对客体的访问导致系统中一个客体的信息流入了另一个客体,此时的系统状态可能就是不安全的。因为策略禁止获得前一个客体信息的主体却可能通过对后一个客体的授权访问获得前一个客体的信息。由此可见,安全的访问控制离不开信息流的约束机制,缺乏信息流约束的访问控制是危险的。

在强制访问控制中,经典的BLP模型[4]是一种基于信息流约束的模型。模型中每个主客体都分配有一个安全级,访问授权由安全级支配关系进行约束,这样的约束使信息只能从高安全级客体流向低安全级客体,保证了低安全级主体不能获得高安全级信息。RBAC模型实施的同样是强制访问控制,但它的访问授权基于的是用户角色承担的职责,而不是对信息流的约束,访问控制中没有针对信息流的约束机制,因此,RBAC模型在信息的机密性保护方面存在严重缺陷,在高安全要求的系统中使用RBAC模型会带来很大的安全隐患。

1 RBAC中的非法信息流

随着RBAC模型应用的不断拓展,它的上述安全缺陷受到越来越多的关注。围绕此问题,学术界展开了一系列的研究。这些研究主要可分为两类。一类是将RBAC和MAC相结合[5],给RBAC系统的主客体分配安全标记,依据标记控制信息流。但是,标记的合理分配是困难的甚至是不可能的,因为安全标记只适用于主客体存在安全级划分的系统。另外,MAC本身是灵活性欠佳的访问控制方式,在RBAC中绑定MAC会导致过多的访问制约,合法访问常会因此遭到拒绝,严重影响系统的可用性;另一类是直接分析RBAC中授权操作产生的信息流情况,对可能导致信息泄漏的信息流进行必要的约束。文献[6]的研究属于此类。在文献[6]中,对信息流的分析是从角色的角度进行的,定义了针对角色而言的非法信息流。如图1所示,角色r_1可以"读"客体o_1,"写"客体o_2;角色r_2可以"读"客体o_2,但不能"读"客体o_1。显然,角色r_1可引发从o_1到o_2的信息流,这样的信息流使r_2可通

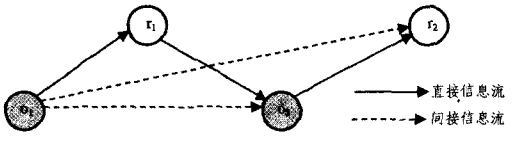

图1 RBAC中的非法信息流

*)项目资助:国家"863"高技术研究发展计划基金项目(2005AA147050)。**葛方斌** 博士研究生,主要研究方向为计算机网络、信息安全;**杨 林** 博导,研究员,主要研究方向为计算机网络系统、信息安全;**王建新** 博导,研究员,主要研究方向为指挥自动化理论、信息安全;**丁 利** 硕士研究生,主要研究方向为信息安全。

过对 o_2 的访问获得策略禁止其获得的 o_1 的信息。此时，o_1 到 o_2 的信息流被定义为非法信息流。为保证系统的安全，文献中提出了针对非法的角色信息流的制约机制。

在 RBAC 系统中，信息安全的威胁来自于用户，对信息流约束的根本目的在于阻止用户获得非授权的信息。基于此，下面换个角度来重新定义非法信息流，希望借此实现对 RBAC 模型系统中信息流的更合理控制。

定义新概念之前首先给出基本 RBAC 模型框架的描述。

基本 RBAC 模型包含下列元素：

U：用户集；R：角色集；S：会话集；OP：操作集；OB：客体集

P：$P\subseteq OP\times OB$，权限集。

UA：$UA\subseteq U\times R$，用户角色指派，用户与角色间的多对多映射。

PA：$PA\subseteq R\times P$，角色权限指派，角色与权限间的多对多映射。

RH：$RH\subseteq R\times R$，角色继承关系，R 上的偏序关系。$(r_1, r_2)\in RH$ 记为 $r_1\succ r_2$，表示 r_1, r_2 是一对父子角色（规定：每个角色都和自身具有继承关系），它们的权限满足关系：$\forall p\in P((r_2,p)\in PA\rightarrow(r_1,p)\in PA)$，即父角色 r_1 拥有子角色 r_2 的全部权限。

$user$：$S\rightarrow U$，会话创建者函数，将会话映射为创建会话的用户。

图 1 给出了 RBAC 基本模型的结构。

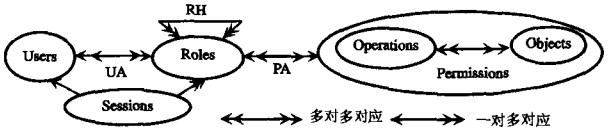

图 2　RBAC 基本模型

约定：RBAC 中，任意 $r\in R$，都存在 $u\in U$，满足 $(u,r)\in UA$。

定义 1　六元组 $Syst=(U,R,P,UA,PA,RH)$ 称为一个 RBAC 系统。$Syst$ 由 RBAC 的基本要素构成，唯一确定了每个用户的权限。

定义 2　$UP\subseteq U\times P$ 称为系统 $Syst$ 的授权状态，当且仅当

$$\forall u\in U\,\forall(op,o)\in P[(u,(op,o))\in UP\leftrightarrow\exists r,r'\in R((r,(op,o))\in PA\wedge r'\succ r\wedge(u,r')\in UA)]$$

定义 3　序列集合 $SEQ=\{seq\mid seq:N\rightarrow UP\}$ 称为系统 $Syst$ 的状态轨迹集合，每个 $seq\in SEQ$ 称为 $Syst$ 的一个状态轨迹。其中，N 是自然数集合。

系统状态轨迹是对系统行为的刻画，反映了系统中访问的有序变迁。

定义 4（权限的信息流特征函数 λ）　$P\rightarrow\{-1,0,1\}$

$$\lambda((op,o))=\begin{cases}-1 & (op,o)\text{使信息流出 }o\\ 0 & (op,o)\text{不产生信息流动}\\ 1 & (op,o)\text{使信息流入 }o\end{cases}$$

定义 5（客体间信息流）

设 $seq_k\in SEQ$，$seq_k=(up_1^k,up_2^k,\cdots,up_i^k,\cdots)$，其中，$up_i^k=(u_i^k,(op_i^k,o_i^k))$，$i=0,1,2,\cdots$。

对于 $o,o'\in OB$，若存在 seq_k 中的元素 $up_{k_1}^k,up_{k_2}^k,\cdots,up_{k_{2n-1}}^k,up_{k_{2n}}^k$，满足：

(1) $o_{k_1}^k=o,o_{k_{2n}}^k=o'$

(2) $o_{k_{2i}}^k=o_{k_{2i+1}}^k(i=1,2,\cdots,n-1)$

(3) $u_{k_{2i-1}}^k=u_{k_{2i}}^k(i=1,2,\cdots,n)$

(4) $\lambda(op_{2i-1}^k,o_{2i-1}^k))=-1(i=1,2,\cdots,n)$

(5) $\lambda(op_{2i}^k,o_{2i}^k))=1(i=1,2,\cdots,n)$

(6) $i<j\Rightarrow k_i<k_j(1\leqslant i,j\leqslant 2n)$

则称 $u_{k_1}^k,u_{k_3}^k,\cdots,u_{k_{2n-1}}^k$ 在状态轨迹 seq_k 下引发了 o 到 o' 的一个信息流，该信息流记为：$o\xrightarrow[o_{k_2}^k\rightarrow o_{k_4}^k\rightarrow\cdots\rightarrow o_{k_{2n-2}}^k]{u_{k_1}^k,u_{k_3}^k,\cdots,u_{k_{2n-1}}^k}o'$，$u_{k_1}^k,u_{k_3}^k,\cdots,u_{k_{2n-1}}^k$ 称为此信息流的引发用户序列，$o_{k_1}^k\rightarrow o_{k_2}^k\rightarrow o_{k_4}^k\rightarrow o_{k_6}^k\rightarrow\cdots\rightarrow o_{k_{2n}}^k$ 称为此信息流的传播路径。若不关心信息流的引发用户，则上述信息流可记为：$o_{k_1}^k\rightarrow o_{k_2}^k\rightarrow o_{k_4}^k\rightarrow o_{k_6}^k\rightarrow\cdots\rightarrow o_{k_{2n}}^k$；若既不关心信息流引发用户也不关心传播路径，则 o 到 o' 的信息流可统一简记为：$o\rightarrow o'$。

若定义中 $n=1$，则称定义中信息流为单步信息流，否则称为跨步信息流。跨步信息流是单步信息流的有序连接。

定义 6（非法信息流）

设 $o,o'\in OB$，$seq\in SEQ$，seq 中存在 o 到 o' 的信息流。

若存在 $u\in U$，满足：

(1) $\exists(op,o')\in P[(u,(op,o'))\in UP\wedge\lambda((op,o'))=-1]$

(2) $\forall(op,o)\in P[(u,(op,o))\in UP\rightarrow\neg(\lambda((op,o'))=-1)]$

则称 seq 中 o 到 o' 的每个信息流是非法信息流，用户 u 称为这些信息流的威胁用户，否则，称这些信息流为合法信息流。

定义 7（信息流安全的状态轨迹）

设 $seq\in SEQ$，若 seq 中所有的信息流都是合法信息流，则称 seq 是信息流安全的状态轨迹。

定义 8（信息流安全的系统）

在系统 $Syst=(U,R,P,UA,PA,RH)$ 中，若任意 $seq\in SEQ$，seq 是信息流安全的状态轨迹，则称 $Syst$ 是信息流安全的系统。

定理 1　设 $o_1\xrightarrow{u_1,u_2,\cdots,u_{n-1}}o_n$ 是 seq 中的信息流，$o_1\rightarrow o_2\rightarrow\cdots\rightarrow o_n$ 是其传播路径。若每个 $o_i\xrightarrow{u_i}o_{i+1}(i=1,2,\cdots,n-1)$ 都是合法的单步信息流，则 $o_1\xrightarrow{u_1,u_2,\cdots,u_{n-1}}o_n$ 是合法信息流。

证明：根据定义 6，只需证，任意 $u\in U$，若存在 $(op,o_n)\in P$，使 $(u,(op,o_n))\in UP$，且 $\lambda((op,o_n))=-1$，都有，存在 $(op',o_1)\in P$，使 $(u,(op',o_1))\in UP$，且 $\lambda((op',o_1))=-1$。

设 u 是 U 中任一用户，存在 $(op_n,o_n)\in P$，使 $(u,(op_n,o_n))\in UP$，且 $\lambda((op_n,o_n))=-1$，由于 $o_{n-1}\rightarrow o_n$ 是合法信息流，由定义 6，存在 $(op_{n-1},o_{n-1})\in P$，使 $(u,(op_{n-1},o_{n-1}))\in UP$，且 $\lambda((op_{n-1},o_{n-1}))=-1$，又由于 $o_{n-2}\rightarrow o_{n-1}$ 是合法信息流，仍由定义 6，存在 $(op_{n-2},o_{n-2})\in P$，使 $(u,(op_{n-2},o_{n-2}))\in UP$，且 $\lambda((op_{n-2},o_{n-2}))=-1$。依次类推，对用户 u，最终必有，存在 $(op_1,o_1)\in P$，使 $(u,(op_1,o_1))\in UP$，且 $\lambda((op_1,o_1))=-1$。　证毕

推论　若 o_1 到 o_n 的一个信息流是非法的，则在该信息流传播路径上，至少存在一个非法的单步信息流。

定理 2　$seq\in SEQ$ 是信息流安全的状态轨迹，当且仅当 seq 中每个单步信息流都是合法的。

证明：必要性是显然的。

充分性

设 $o \xrightarrow{S_U} o'$(S_U 是引发用户序列)是 seq 中任一信息流，若 $o \xrightarrow{S_U} o'$是单步信息流，由条件直接知，$o \xrightarrow{S_U} o'$是合法信息流。若 $o \xrightarrow{S_U} o'$是跨步信息流，假设该信息流是非法的，则由定理 1 的推论知，在 $o \xrightarrow{S_U} o'$的传播路径上必存在非法的单步信息流，这与条件矛盾。因此，seq 中所有信息流都是合法的，故 seq 是信息流安全的状态轨迹。 证毕

2 两种非法信息流的关系

定义 6 从用户的角度定义了非法信息流，而文献[6]从角色的角度给出了非法信息流的定义。由于信息流的控制目的在于阻止用户获得访问控制策略禁止其获得的信息。从这一点来看，于用户角度定义非法信息流是合理的。那么，于角色角度定义非法信息流合理性如何呢？如果不合理，它可能导致怎样的问题呢？这些问题可以从两种不同定义的非法信息流之间的关系中找到答案。

定义 6(非法的角色信息流)

设 $o,o' \in OB, seq \in SEQ, seq$ 中存在 o 到 o'的信息流。

若存在 $r \in R$，满足：

(1) $\exists (op,o') \in P[(r,(op,o')) \in PA \wedge \lambda((op,o')) = -1]$

(2) $\forall (op,o) \in P[(r,(op,o)) \in PA \rightarrow \neg(\lambda((op,o)) = -1)]$

则称，o 到 o'的信息流为非法的角色威胁信息流，简称非法角色信息流，r 称为该信息流的威胁角色。

该定义和文献[6]的非法信息流定义实质是一致的。为区别起见，将定义 6 定义的非法信息流称为非法的用户威胁信息流，简称非法的用户信息流。

下面的定理反映了两种非法信息流之间的关系。

定理 3 在 RBAC 系统的状态轨迹中，非法的用户信息流一定也是非法的角色信息流。

证明：设 $o \xrightarrow{S_U} o'$是系统状态轨迹 seq 中非法的用户信息流，则存在 $u \in U$，满足

①存在$(op,o') \in P$，使$(u,(op,o')) \in UP$，且 $\lambda((op,o')) = -1$；

②任意$(op,o) \in P$，若$(u,(op,o)) \in UP$，都有 $\lambda((op,o)) = -1$ 不成立。

由①中的存在$(op,o') \in P$，使$(u,(op,o')) \in UP$ 知，存在角色 $r_0 \in R$，$(u,r_0) \in UA$，且$(r_0,(op,o')) \in PA$，加之 $\lambda((op,o')) = -1$，因此，对角色 r_0，定义 6 的(1)成立。

对任意$(op,o) \in P$，若$(r_0,(op,o)) \in PA$，由于$(u,r_0) \in UA$，从而$(u,(op,o)) \in UP$ 根据②有，$\lambda((op,o)) = -1$ 不成立，所以，对角色 r_0，定义 6 的(2)成立。

由此可见，$o \xrightarrow{S_U} o'$也是非法的角色信息流。 证毕

定理 3 的逆命题不成立，下面的例子可说明这一点。

例 1 如图所示，角色 r_0 引发了从客体 o 到 o'的非法的角色信息流 $o \rightarrow o'$，$R_o^{o'} = \{r_1, r_2, \cdots, r_k\}$是该信息流的威胁角色集。

若对任意 $r_i \in R_o^{o'}$，拥有角色 $r_i (1 \leqslant i \leqslant k)$的用户同时拥有角色 r_0，则信息流 $o \rightarrow o'$对用户而言是合法的，而不是非法的。因为对任意一个用户 u，若 u 通过角色 r 拥有一个权限(op, o')，且 $\lambda((op,o')) = -1$。当 $r \notin R_o^{o'}$ 时，由于 r 是非威胁角色，所以 r 至少拥有一个权限(op',o)，使 $\lambda((op',o)) = -1$，(op',o)当然也是用户 u 拥有的权限；当 $r \in R_o^{o'}$ 时，根据条件，u 拥有角色 r_0，而 r_0 显然具有某个权限 (op_0,o)，满足 $\lambda((op_0,o)) = -1$，这也是用户 u 的权限。可见，$o \rightarrow o'$是合法的用户信息流。

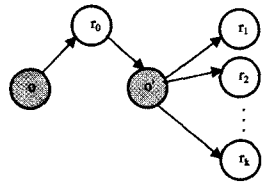

图 3 o 到 o'的非法的角色信息流

例 1 和定理 3 说明，非法的角色信息流比非法的用户信息流外延更广。对非法的角色信息流进行控制虽然可防止信息的非授权泄漏，但它也可能使用户的合理请求遭到拒绝，给系统灵活性造成不良影响。

3 非法的用户信息流的检测与更新

一个系统，如果其中所有的信息流都是合法的，那么该系统就是信息流安全的。因此，保证系统的信息流安全性只需确定系统中所有可能的非法信息流，并在访问授权时对它们加以限制。而定理 2 进一步指出，只需控制非法的单步信息流就可控制全部的非法信息流。下面首先来给出一个非法的单步用户信息流检测算法。若无特殊说明，后面出现的信息流都是指单步用户信息流。

输入：集合 U,R,OB,P,UA,PA，其中，$OB = \{o_1, o_2, \cdots, o_m\}$

输出：所有非法的单步信息流(包括引发及威胁用户)和所有合法的单步信息流(包括引发用户)

算法 1

(1)对每个客体 o_i，计算 A_i, B_i

$A_i = \{(op,o_i) \in P \mid \lambda((op,o_i)) = 1\}$，$B_i = \{(op,o_i) \in P \mid \lambda((op,o_i)) = -1\}$

(2)对每个客体 o_i，计算 C_i, D_i

$$C_i = \bigcup_{(op.o_i) \in A_i, (r,(op.o_i)) \in PA} \{r\}, D_i = \bigcup_{(op.o_i) \in B_i, (r,(op.o_i)) \in PA} \{r\}$$

(3)对每个客体 o_i，计算 E_i, F_i

$$E_i = \bigcup_{r \in C_i, (u,r) \in UA} \{u\}, F_i = \bigcup_{r \in D_i, (u,r) \in UA} \{u\}$$

(4)对任意 $o_i, o_j \in OB (i \neq j)$，计算 $F_i \cap E_j$ 和 $F_j - F_i$，若 $F_i \cap E_j \neq \phi, F_j - F_i \neq \phi$，则输出四元组$(i,j,F_i \cap E_j, F_j - F_i)$，若 $F_i \cap E_j \neq \phi, F_j - F_i = \phi$，则输出三元组$(i,j,F_i \cap E_j)$，结束。

算法中，A_i, B_i 分别代表能使信息流入和流出客体 o_i 的权限集合；C_i, D_i 分别代表具有权限能使信息流入和流出客体 o_i 的角色集合；E_i, F_i 分别代表具有权限能使信息流入和流出客体 o_i 的用户集合；四元组$(i,j,F_i \cap E_j, F_j - F_i)$表示 o_i 到 o_j 的非法信息流，$F_i \cap E_j$ 和 $F_j - F_i$ 分别表示该信息流的引发和威胁用户集；三元组$(i,j,F_i \cap E_j)$表示 o_i 到 o_j 的合法信息流，$F_i \cap E_j$ 是该信息流引发用户集。

例 2 设在一个 RBAC 系统中，$U = \{u_1, u_2, u_3\}$，$R = \{r_1, r_2, r_3\}$，$OB = \{o_1, o_2, o_3, o_4\}$，$UA = \{(u_1, r_1), (u_2, r_2), (u_3, r_3)\}$，角色的权限特征如下表 1($(o_1, 1)$ $(o_1, -1)$分别代表对应角色具有权限能使信息流入和流出客体 o_1，其它依此类推)。

表 1　角色的权限特征

角　色	角色权限特征
r_1	$(o_1,1),(o_3,-1),(o_4,-1)$
r_2	$(o_1,-1),(o_2,1)$
r_3	$(o_2,1),(o_3,-1),(o_4,1)$

对每个客体，算法首先分步计算得到能使信息向其流入和从其流出的用户集，结果如表 2。

表 2　使信息流入和流出客体的用户集

角　色	使信息流入用户集	使信息流出用户集
o_1	$E_1=\{u_1\}$	$F_1=\{u_2\}$
o_2	$E_2=\{u_2,u_3\}$	$F_2=\succ$
o_3	$E_3=\succ$	$F_3=\{u_1,u_3\}$
o_4	$E_4=\{u_3\}$	$F_4=\{u_1\}$

根据表 2 中的结果，算法中的条件 $F_i\cap E_j\neq\phi$ 确定了所有的信息流，其中，同时满足 $F_j-F_i\neq\phi$ 是非法信息流，否则是合法信息流。表 3 给出了所有的信息流。

表 3　合法与非法信息流

合法信息流	非法信息流
$(1,2,\{u_2\})$	$(3,1,\{u_1\},\{u_2\})$
$(3,2,\{u_3\})$	$(4,1,\{u_1\},\{u_2\})$
$(3,4,\{u_3\})$	

图 4 显示了系统中的信息流情况。

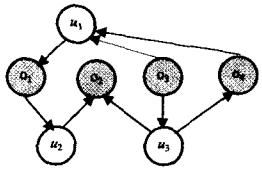

图 4　系统中用户与客体间信息流

在某些 RBAC 系统中，经常需要增加或删除用户。用户的变化会导致怎样的信息流变化呢？添加用户时，由于原有非法信息流的引发和威胁用户依然存在，因此，原有的非法信息流在添加用户后仍是非法信息流。即，非法信息流具有随用户添加不减性。进一步，若添加的用户权限和系统中某个已存在用户权限完全相同，则非法信息流还具有随用户添加不增性。

定理 4　在 RBAC 系统中添加一个用户，若该用户的权限与系统中某个已存在用户的权限完全相同，则系统中信息流状况保持不变。

证明：设新增用户为 u，u 与系统中已存在用户 u_0 权限完全相同。

首先，u 不会增加系统中的信息流，因为每个可由 u 引发的信息流都可由 u_0 引发。其次，对系统中每个原来合法的信息流，由于 u_0 不是该信息流的威胁用户，因而 u 也不是该信息流的威胁用户，所以该信息流在添加 u 后仍是合法的信息流。对系统中每个原来非法的信息流，在添加 u 后显然仍是非法信息流。可见，添加 u 后系统的信息流状况没有发生变化。证毕。

应注意的是，定理中信息流状况指的是信息流的存在性与合法性，不包括信息流的引发和威胁用户情况。实际上，添加定理中的用户可能导致系统中某些信息流的引发或威胁用户的增加。

下面的定理 5 揭示了一个更一般的结论。

定理 5　在 RBAC 系统中添加一个用户，若该用户的权限是系统中若干已存在用户的权限的并集，则系统中原有信息流仍存在且合法性不变。

证明：设新增用户 u 的权限是系统中原有用户 $u_1,u_2,\cdots,u_m$ 的权限的并集。系统中原有的每一个信息流 $o_i\rightarrow o_j$ 在添加 u 后显然仍存在。若 $o_i\rightarrow o_j$ 是合法信息流，假设添加 u 后变为非法信息流，则 u 是该信息流的唯一威胁用户。于是，存在权限 (op,o_j)，使 $\lambda((op,o_j))=-1$，且 u 拥有该权限。由于 u 的权限是 $u_1,u_2,\cdots,u_m$ 的权限之并，故存在 $u_k\in\{u_1,u_2,\cdots,u_m\}$，$u_k$ 拥有权限 (op,o_j)。由于 u_k 不是信息流 $o_i\rightarrow o_j$ 的威胁用户，因而，存在权限 (op',o_i)，使 $\lambda((op',o_i))=-1$，且 u_k 拥有该权限，(op',o_i) 显然也是 u 拥有的权限，这与 u 是威胁用户矛盾。若 $o_i\rightarrow o_j$ 是非法信息流，添加 u 后显然还是非法信息流。证毕。

需要指出的是，添加定理 5 中的用户可能增加新的信息流，新增的信息流既可能是合法信息流，也可能是非法信息流。这与定理 4 的情况不同。

推论　在一个 RBAC 系统中添加用户 u，若存在已有用户 $u_0,u_1,u_2,\cdots,u_m$，满足 u 的权限包含于 u_0 的权限，且 u 的权限是 $u_1,u_2,\cdots,u_m$ 权限的并集，则添加 u 后系统中信息流状况保持不变。即系统中无信息流的增减，且原有信息流的合法性保持不变。

对系统的信息流进行控制首先需明确系统中存在的非法信息流。由于非法信息流随用户的添加可能发生变化，因此，系统中添加用户时非法信息流的更新是重要的。下面给出一个添加用户时非法信息流的增量更新算法。该算法以算法 1 的结果为基础。

输入：角色权限指派 PA，新增用户 u，u 的角色指派集合 R_u，原有的合法信息流集合 G，原有的非法信息流集合 H，所有客体原有的 $E_i,F_i(i=1,2,\cdots,m)$

输出：新的非法信息流集合

算法 2

(1)计算用户 u 拥有的权限集 P_u，$P_u=\bigcup\limits_{r\in R_u,(r,(op,o))\in PA}\{(op,o)\}$。

(2)计算 u 能使信息流入和流出的客体集 $can\_w(u)$ 和 $can\_r(u)$，$can\_w(u)=\bigcup\limits_{(op,o)\in P_u,\lambda((op,o))=1}\{o\}$，$can\_r(u)=\bigcup\limits_{(op,o)\in P_u,\lambda((op,o))=-1}\{o\}$。

(3)对每个 $(i,j,F_i\cap E_j)\in G$，若 $o_j\in can\_r(u)$，$o_i\notin can\_r(u)$，则输出 $(i,j,F_i\cap E_j,\{u\})$ //输出原有合法信息流转化而成的非法信息流。

(4)对每个 $o_i\in can\_r(u)$，$o_j\in can\_w(u)$，若 $F_j-F_i\neq\phi$，则输出 $(i,j,(F_i\cap E_j)\cup\{u\},F_j-F_i)$//输出引发用户集包含 u 的非法信息流。

(5)对每个 $(i,j,F_i\cap E_j,F_j-F_i)\in H$，若 $o_i\in can\_r(u)\rightarrow\neg o_j\in can\_w(u)$ 与 $o_j\in can\_r(u)\wedge\neg o_i\in can\_r(u)$ 都为真，则输出 $(i,j,F_i\cap E_j,(F_j-F_i)\cup\{u\})$，若 $o_i\in can\_r(u)\rightarrow\neg o_j\in can\_w(u)$ 与 $o_j\in can\_r(u)\rightarrow o_i\in can\_r(u)$ 都为真，则输出 $(i,j,F_i\cap E_j,F_j-F_i)$ //输出原有非法信息流转化而成且转化后 u 不在引发用户集中的非法信息流。

算法中输出的所有四元组构成了新的非法信息流集合。

新的非法信息流可分为两个部分：引发用户集包含 u 的和引发用户集不包含 u 的。引发用户集包含 u 的由算法的步骤(4)生成。引发用户集不含 u 的非法信息流在添加 u 之前

(下转第 89 页)

结束语 本文从性能的角度，度量 BitTorrent 的行为，证明 BitTorrent 不是高效的，给出并分析 Choking/Unchoking 机制对 BitTorrent 效率的影响。在证明 BitTorrent 不是高效的过程中，我们设计了一种接近于互联网行为的、专门用于度量 P2P 内容分发协议的新方法，且全新设计了一个称为 ShareStorm 的协议，作为参照对象。通过我们提出的度量方法，我们证明 Share Storm 比 BITTorrent 具有更好的效率。

参 考 文 献

[1] ISP's battle against encrypted BitTorrent downloading. http://www.afterdawn.com/news/archive/7896.cfm

[2] Cohen B. Incentives build robustness in BitTorrent // Proceedings of the First Workshop on the Economics of Peer-to-Peer Systems. Berkeley, CA, June 2003

[3] Akan O B. On the Throughput Analysis of Rate-based and Window-based Congestion Control Schemes. Computer Networks Journal (Elsevier Science), 2003

[4] Vahdat A, Yocum K, Walsh K, et al. Scalability and Accuracy in a Large-scale Network Emulator // Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI). December 2002

[5] BitComet. http://www.bitcomet.com/

[6] Adar E, Huberman B A. Freeriding on Gnutella. First Monday, 2000, 5(10)

[7] Anagnostakis K G, Greenwald M B. Exchange-based incentive m-echanismsfor peer-to-peer file sharing // Proc. of IEEE Icdcs (March 2004)

[8] Bharambe A, Herley C, Padmanabhan V. Analyzing and improvingbittorrent performance // Proc. of IEEE Infocom. 2006

[9] Qiu D, Srikant R. Modeling and performance analysis of bittorrentlike peer-to-peer networks // Proc. ACM SIGCOMM'04. Portland, Oregon, USA, Aug. 30—Sept. 3, 2004

[10] Gkantsidis C, Rodriguez P. Network Coding for Large Scale ContentDistribution // IEEE INFOCOM 2005, Miami. March 2005

[11] Cherkasova L, Lee J. Fastreplica: Efficient large file distribution within content delivery networks // USENIX Symposium on Internet Technologies and Systems. 2003

[12] Levin D, Sherwood R, Bhattacharjee B. Fair File Swarming with FOX // International Workshop on Peer-to-Peer Systems (IPTPS). 2006

[13] Chun B G, Wu P, Weatherspoon H, et al. ChunkCast: An AnycastService for Large Scale Content Distribution. In IPTPS, 2006

[14] Park K, Pai V S. Scale and performance in the coblitz large-file distribution service. In NSDI, 2006

(上接第 68 页)

显然就存在(原来可能是合法信息流)，所以，这一类非法信息流只可能由原来的合法信息流或非法信息流转化而来，由合法信息流转化来的由步骤(3)生成，由非法信息流转化来的由步骤(5)生成。

注意：算法 2 省略了对合法信息流和每个 E_i，F_i 的更新计算，它们是非法信息流进一步更新的前提。

如果系统中用户的变化是删除用户，此时系统中信息流显然不可能增加，且原来合法的信息流不会转化为非法的信息流。非法信息流的更新相对简单。只需将被删除用户从原来非法的用户信息流的引发用户集和威胁用户集中剔除，剔除后仍非法的信息流和没有经过剔除的非法信息流构成了新的非法信息流集合。

如果在系统中添加新角色并赋予已有的用户，这相当于给系统中某些用户增加权限，系统中信息流一般会增加，并且，原来合法与非法的信息流都可能变为相反的信息流。此时，最简单也是最直接的信息流更新方法是，首先更新每个客体 o_i 的 E_i 和 F_i，然后由所有的 E_i 和 F_i 直接生成新的非法与合法信息流。

4 非法信息流的控制

RBAC 系统中的非法信息流可能导致信息的非授权泄漏，为保证客体信息的安全性，需要对这些非法信息流进行控制。控制非法信息流实际是控制非法信息流引发用户的"写"操作，这与用户的访问历史有关，是动态的。下面给出一个非法信息流控制算法。算法中假定用户的访问请求已满足 RBAC 系统基于角色的授权约束，即用户 u 与其请求权限 (op,o) 满足：$(u,(op,o))\in UP$。

输入：用户 u，u 请求的权限 (op,o)，系统非法信息流集合 H；

输出：授权访问或拒绝访问。

算法 3

(1)对每个非法信息流 $(i,j,X,Y)\in H$，设置访问历史记载项 $\tau(o_i,o_j)$，初始化 $\tau(o_i,o_j)=\phi$。

(2)若 (op,o) 满足 $\lambda((op,o))=-1$，且 $o=o_i$，则将 u 添加到每个 $\tau(o_i,o_j)$ 中，并对 u 的访问授权。

(3)若 (op,o) 满足 $\lambda((op,o))=1$，则检查每个 $\tau(o_i,o_j)$，如果存在某个 $\tau(o_i,o_j)$ 满足 $o_j=o$ 且 $u\in\tau(o_i,o_j)$，则拒绝 u 的访问请求。否则对 u 的访问授权。

(4)若 (op,o) 是其它类型的权限，都对 u 的访问授权。

算法为每个非法信息流 (i,j,X,Y) 生成一个历史记载项 $\tau(o_i,o_j)$。用于记载 X 中已执行过使信息流出 o_i 的操作的用户。这些用户被记载后请求的权限若能使信息流入 o_j 时，将拒绝请求，其它情况的访问请求都会获得授权。

结束语 RBAC 模型的访问授权基于的是用户角色所承担的职责，而不是对信息流的约束。因此，用户的授权访问可能引发不安全的信息流，从而导致信息的非授权泄漏。为保证 RBAC 模型系统的信息流安全性，在定义了非法的用户信息流概念的基础上，重点研究了非法信息流的检测、更新以及控制问题，给出了具体的检测、更新和控制算法。将这些算法纳入到 RBAC 模型系统的授权管理中可有效杜绝系统中非法信息流的发生，实现访问控制对信息机密性的保护。

参 考 文 献

[1] Sandhu R, Coyne E, Feinstein H et al. Role-based access control model [J]. IEEE comprter, 1996, 29(2): 38-47

[2] Ferraiolo D F, Sandhu R S, Gavrila S, et al. Proposed NIST standardfor role-based access control [J]. ACM transactions on information and system security, 2001, 3(4): 182-186

[3] Al kahtani M, Sandhu R. Induced role hierarchies with attribute-based RBAC [C] // proceedings of the 8th ACM symposium on access control models and technologies. Villa Gallia, 2003: 142-148

[4] Bell D E, LaPadula L J. Secure computer system: Unified eposition and MUL TICS interpretation. The MITRE Corporation, Technical Report: MTR-2997 Revision1. 1976

[5] 李斓，冯登国，徐震. RBAC 与 MAC 在多级关系数据库中的综合模型. 电子学报，2004，32(10)：1635-1639

[6] Izaki k, Tanaka K, Tskizawa M. Information flow control in role-based model for distributed objects // Proc of the 8th Int'l Conf. on Parallel and Distributed Systems. Los Alamitos, CA: IEEE Computer Society Press, 2001: 363-370