第 38 卷第 10 期
2010 年 10 月

同济大学学报(自然科学版)
JOURNAL OF TONGJI UNIVERSITY(NATURAL SCIENCE)

Vol. 38 No. 10
Oct. 2010

文章编号：0253-374X(2010)10-1540-06

DOI:10.3969/j.issn.0253-374x.2010.10.026

基于属性委托模型中的自动委托撤销

叶春晓[1]，符云清[2]，李　响[3]

(1.重庆大学 计算机学院，重庆 400044；2.重庆大学 软件学院，重庆 400044；3.重庆市勘测院，重庆 400020)

摘要：基于属性的扩展委托模型中($ABDM_A$)，受托者必须同时满足委托先决条件和委托属性表达式才能被委托权限或角色.为提高委托的安全性，方便撤销操作，对该模型进行了扩展.首先定义了用户委托撤销类型，委托者可由此自行撤销委托出去的角色.然后为了支持自动撤销，在模型中引入了自动撤销机制，提出了 4 种自动撤销类型：由委托有效时间区间、用户先决角色变化、用户属性表达式变化和委托角色属性表达式变化引起的自动委托撤销.接着讨论了自动撤消的系统开销和自动撤销引起的多步委托安全性问题.最后给出了自动撤销算法和系统实现框架.

关键词：信息安全；访问控制；委托；自动撤销；属性

中图分类号：TP 390.2　　**文献标识码**：A

Auto-revocation in Attribute-based Delegation Model

YE Chunxiao[1], *FU Yunqing*[2], *LI Xiang*[3]

(1. College of Computer Science, Chongqing University, Chongqing 400044, China; 2. College of Software Engineering, Chongqing University, Chongqing 400044, China; 3. Chongqing Survey Institute, Chongqing 400020, China)

Abstract: In attribute-based extended delegation model ($ABDM_A$), delegatee must satisfy both delegation prerequisite condition(CR) and delegation attribute expression(DAE) when assigned to a delegation role. We extended $ABDM_A$ to make delegation more safe and easy to revoke delegation role form delegateee. This paper first defines user revocations, by which delegator can revoke delegate role from delegate. Then an auto-revocation mechanism is introduced and four types of auto-revocations are defined as revocation caused by delegation duration, the change of delegatee's CR, the change of user's DAE and the changed of delegation role's DAE. System cost and security in multi-step delegation caused by auto revocation are also discussed. An auto revocation algorithm and a system architecture are proposed in this paper.

Key words: information security; access control; delegation; auto revocation; attribute

委托可实现将委托者的全部或部分权限或角色指派给受托者.撤销是委托的逆过程，其作用是将受托者获得的委托权限或角色收回.

在 RBAC(基于角色的访问控制)[1]基础上，RBDM(基于角色的委托模型)[2]将角色概念引入到委托模型中，详细讨论了委托撤销，将其分为由系统条件自动触发的系统撤销和用户撤销 2 种.RDM2000(基于规则的委托模型 2000)[3]模型支持层次角色和多步委托，详细讨论了不同的委托和撤销方式：将委托分为初始用户委托和受托者委托 2 种；将撤销分为由委托期限约束条件引起的撤销和用户撤销，并讨论了级联撤销问题.PBDM(基于权限的委托模型)[4]支持部分委托模式，该模型主要针对委托操作，对委托撤销未做深入研究.RPRDM(基于重复和部分权限的转授权模型)[5]支持重复和部分权限委托，定义了 4 种委托撤销：只有委托者才能完成撤销、具有委托角色的用户均可完成撤销、系统自动撤销和系统管理员撤销.

在 QBCDM(基于量化角色的可控委托模型)[6]中对委托撤销进行了阐述，支持了 8 种撤销模式，但不涉及自动撤销问题.TRDM(具有时限的基于角色的转授权模型)[7]中提出了 2 种委托撤销方式：基于时限限制的委托撤销和用户主动委托撤销，但对基于时限限制的委托撤销没有详细说明.文献[8]中给出了可控、多粒度的用户-用户的多步委托，并讨论了多步撤销问题，给出了其扩展方法以支持基于时间约束的撤销，但未说明相应的撤销方式.RBCD(基

收稿日期：2009-07-09

基金项目：国家自然科学基金资助项(60803027)；国家“八六三”高技术研究发展计划资助项目(2007AA01Z445)；重庆市自然科学基金资助项目(CSTC,2008BB2320)

作者简介：叶春晓(1973—)，男，副教授，工学博士，主要研究方向为访问控制、本体等. E-mail:yecx@cqu.edu.cn

符云清(1969—)，男，教授，工学博士，主要研究方向为信息安全、网络教育. E-mail:yqfu@cqu.edu.cn

于角色的级联委托模型)[9]讨论了在信任系统中基于角色的多步委托问题,提到基于时限的委托撤销,但未提到自动委托撤销问题.文献[10-11]均讨论了委托和撤销问题,但未详细讨论自动撤销.

为提高委托过程的安全性和灵活性,文献[12]中提出了一个基于属性的扩展委托模型($ABDM_A$).在委托过程中,只有同时满足委托先决条件和委托属性表达式的受托者才能获得委托角色或权限.$ABDM_A$ 就基于属性的委托进行了讨论,其中并没有讨论委托撤销的问题,特别是自动撤销.本文在此基础上对委托撤销,特别是自动撤销进行了详细讨论.

1　$ABDM_A$ 模型扩展

此处只给出在文献[12]基础上修改和增加的定义及元素.U,R,CR,P,P_M,P_N,U_{ee},U_{de},R_{TDM},R_{TDN},R_{TD},T_c,T_s,T_e 分别为用户、角色、授权先决角色、权限、单调权限、非单调权限、不确定用户、确定用户、单调委托角色、非单调委托角色、临时委托角色、系统当前时间点、可作为起始时间和可作为终止时间的系统时间点集合.URA,PRA,PDA 分别为用户-角色指派、普通角色-权限指派和委托角色-权限指派关系.

定义 1　$U_{EAM}\subseteq U_{ee}\times R_{TDM}$:不确定受托者-单调委托角色之间的指派关系;$U_{EAN}\subseteq U_{ee}\times R_{TDN}$:不确定受托者-非单调委托角色之间的指派关系.

用户 u 是否激活角色 r 的判断函数 $c_{anActive}(u,r):=\{T$:如果 u 可以激活 r;F:如果 u 不能激活 $r\}$.该函数是为了委托的安全性引入,详细说明见第 3 节.相应地下面 3 个函数均进行了修改.

$U(r_{td})=\{u\mid u\in U,u$ 包含 r_{td} 要求的所有先决角色 $\wedge E_{u.DA}\triangleright E_{r_{td}.DA}\wedge c_{anActive}(u,r_{td})=T\}$,返回满足委托在 CR 和属性表达式上要求的用户集合;$U_{de}(r_{td})\{u\mid u\in U_D(r_{td})\wedge u\in U(r_{td}),\forall p\in p_{er\_d}(r_{td}),\wedge p\notin p_{er\_u}(u)\}$,返回符合委托要求的确定受托者集合,其中 U_D 为返回拥有某个委托角色的用户集合的函数,p_{er_d} 为返回某个用户所拥有的权限集合函数;$U_{ee}(r_{td})=\{u\mid u\in U(r_{td}),\forall p\in p_{er\_d}(r_{td}),p\notin p_{er\_u}(u)\}$,返回符合委托要求的非确定受托者集合,其中 p_{er_u} 为返回某个角色所拥有权限集合函数.

用户 u 拥有委托角色 r_{td} 的有效时间函数 $e_{time}(u,r_{td})=\{t\mid t\in T_E\wedge((u,r_{td})\in U_{DAM}\vee(u,r_{td})\in U_{DAN}\vee(u,r_{td})\in U_{EAM}\vee(u,r_{td})\in U_{EAN})\}$,$t$ 缺省为系统允许的最大系统时间;系统的当前时间函数 $c_{urtime}()=\{t\mid t\in T_c\}$.上面 2 个函数为新增的函数,与基于时间的自动撤销有关,可见后面的分析.

作用于 U_{DAM} 的委托约束关系为 $c_{an\text{-}delegateDP}\subseteq R\times U_{de}\times[T_s,T_e]\times R_{TDM}$;作用于 U_{DAN} 的委托约束关系为 $c_{an\text{-}delegateDT}\subseteq R\times U_{de}\times[T_s,T_e]\times R_{TDN}$;作用于 U_{EAM} 的委托约束关系为 $c_{an\text{-}delegateUP}\subseteq R\times U_{ee}\times[T_s,T_e]\times R_{TDM}$;作用于 U_{EAN} 的委托约束关系为 $c_{an\text{-}delegateUT}\subseteq R\times U_{ee}\times[T_s,T_e]\times R_{TDN}$.

对原模型委托关系进行了扩展,加入了委托有效时间区间概念以支持自动撤销.

$ABDM_A$ 同样支持手工撤销操作:U_{DAM} 委托关系的撤销操作可扩展为:$c_{an\text{-}revokeDP}\subseteq U\times R_{TD}\times U$;$(u,r_{td},u')\in c_{an\text{-}revokeDP}\Leftrightarrow u\in U_{RA}(r_{td})\wedge(u',r_{td})\in U_{DAM}$;$U_{DAN}$ 委托关系的撤销操作可扩展为 $c_{an\text{-}revokeDT}\subseteq U\times R_{TD}\times U$;$(u,r_{td},u')\in c_{an\text{-}revokeDT}\Leftrightarrow u\in U_{RA}(r_{td})\wedge(u',r_{td})\in U_{DAN}$;$U_{EAM}$ 委托关系的撤销操作可扩展为 $c_{an\text{-}revokeUP}\subseteq U\times R_{TD}\times U$;$(u,r_{td},u')\in c_{an\text{-}revokeUP}\Leftrightarrow u\in U_{RA}(r_{td})\wedge(u',r_{td})\in U_{EAM}$;$U_{EAN}$ 委托关系的撤销操作可扩展为 $c_{an\text{-}revokeUT}\subseteq U\times R_{TD}\times U$;$(u,r_{td},u')\in c_{an\text{-}revokeUT}\Leftrightarrow u\in U_{RA}(r_{td})\wedge(u',r_{td})\in U_{EAN}$.

上面定义的扩展后的 $ABDM_A$ 模型主要元素之间的关系如图 1 所示.

2　自动撤销

2.1　委托有效时间区间引起的自动撤销

由时间引起的自动撤销关系 c_{an_ARbyT} 定义如下:

定义 2　$c_{an\text{-}ARbyT}\subseteq U\times R_{TD}$,$(u,r_{td})\in c_{an\text{-}ARbyT}\Leftrightarrow r_{td}\in r(u)\wedge e_{time}(u,r_{td})<c_{urtime}()$.其中 $r(u)$ 返回用户 u 所拥有的角色.

该定义表明被委托者 u 拥有委托角色 r_{td},当委托有效时间区间失效时,系统将自动撤销该委托.

2.2　委托先决角色引起的自动撤销

委托先决角色的改变将引起受托者可能因为不具有执行委托角色所需要的其他角色,而使其无法执行委托角色.委托先决角色引起的自动撤销关系 c_{an_ARbyCR} 定义如下:

定义 3　$c_{an\text{-}ARbyCR}\subseteq U\times R_{TD}$,$(u,r_{td})\in c_{an\text{-}ARbyCR}\Leftrightarrow r_{td}\in r(u)\wedge c_r(u,r_{td})\not\subset r(u)$.其中 $c_r(u,r_{td})$ 返回用户 u 拥有 r_{td} 所必须拥有的先决角色.

该定义表明当受托者 u 拥有的作为得到委托角色 r_{td} 先决角色发生变化后,将自动进行撤销操作.

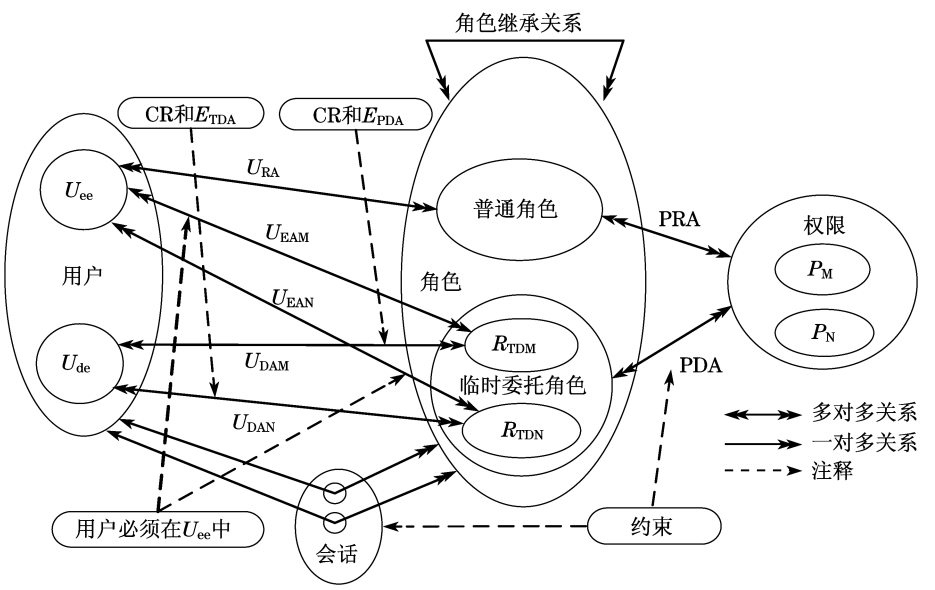

图1 扩展后的 $ABDM_A$ 模型

Fig.1 Extended $ABDM_A$ model

2.3 由属性表达式的变化引起的自动撤销

2.3.1 由用户属性表达式变化引起撤销

用户属性表达式的变化可能导致对 E_{TDA} 和 E_{PDA} 的优先级判定结果的不同.

(1) 临时委托

在这种委托类型中,需要判断用户的属性表达与委托角色的 E_{TDA} 之间的优先级关系.

定义4 由用户的 u 的属性表达式(记为 $E_{u.DA}$)变化引起的临时委托自动撤销需满足下面关系:

$c_{an\text{-}ARbyDT} \subseteq U \times R_{TD}, (u, r_{td}) \in c_{an\text{-}ARbyDT} \Leftrightarrow r_{td} \in r(u) \wedge M_N(u, r_{td}) = T \wedge \neg E_{u.DA} \rhd E_{r_{td}.DA}$. 其中 M_N 为判断用户 u 是否将角色 r 临时委托出去的函数,若是,则函数返回 T,若否,则返回 F.

当 $E_{u.DA}$ 发生变化后且不再满足其拥有的委托角色的委托属性表达式(记为 $E_{r_{td}.DA}$),该 r_{td} 将自动从 u 中撤销.

(2) 永久委托

同样可以类似定义永久委托自动撤销需满足的关系。

定义5 由用户的 $E_{u.DA}$ 变化引起的永久委托自动撤销关系 c_{an_ARbyDP} 可定义为

$c_{an\text{-}ARbyDP} \subseteq U \times R_{TD}, (u, r_{td}) \in c_{an\text{-}ARbyDP} \Leftrightarrow r_{td} \in r(u) \wedge M_N(u, r_{td}) = F \wedge \neg E_{u.DA} \rhd E_{r_{td}.DA}$.

当 $E_{u.DA}$ 发生变化后且不再满足其拥有的委托角色的 $E_{r_{td}.PDA}$,该 r_{td} 将自动从 u 中撤销.

2.3.2 由权限属性表达式变化引起的撤销

结合权限属性表达式的类型和委托类型进行讨论。

(1) 临时委托

定义6 由权限的 p 的临时委托属性表达式(记为 $E_{p.TDA}$)变化引起的临时委托自动撤销需关系 c_{an_ARbyPT} 可定义为

$c_{an\text{-}ARbyPT} \subseteq U \times P \times R_{TD}, (u, p, r_{td}) \in c_{an\text{-}ARbyPT} \Leftrightarrow r_{td} \in r(u) \wedge \neg E_{u.DA} \rhd E_{r_{td}.TDA} \wedge p \in p_{er\_d}(r_{td}) \wedge M_N(u, r_{td}) = F$.

在一个临时委托中,当 $E_{r_{td}.DA}$ 中的某个 $E_{p.TDA}$ 发生变化后,如果用户 u 的 $E_{u.DA}$ 不再满足 $E_{r_{td}.DA}$,该 r_{td} 将自动从 u 中撤销.

(2) 永久委托

同样地,对于永久委托的自动撤销需要满足的关系可定义如下:

定义7 由权限 p 的永久委托属性表式(记为 $E_{p.PDA}$)变化引起的永久委托自动撤销关系 c_{an_ARbyPP} 可定义为

$c_{an\text{-}ARbyPP} \subseteq U \times P \times R_{TD}, (u, p, r_{td}) \in c_{an\text{-}ARbyPP} \Leftrightarrow r_{td} \in r(u) \wedge \neg E_{u.DA} \rhd E_{r_{td}.PDA} \wedge p \in p_{er\_d}(r_{td}) \wedge M_N(u, r_{td}) = T$.

在一个永久委托中,当 $E_{r_{td}.DA}$ 中的某个 $E_{p.PDA}$ 发生变化后,如果用户 u 的 $E_{u.DA}$ 不再满足 $E_{r_{td}.PDA}$,该 r_{td} 将自动从 u 中撤销.

2.4 自动撤销优先级、系统开销及安全性分析

2.4.1 自动撤销优先级

由于4种自动撤销类型触发条件出现的频率不同,因而其优先级也不同:对一个具体的角色而言,其

CR 很少改变,即由 CR 引起的自动撤销发生的频率相对较低.对于用户的属性表达式而言,部分用户的属性表达式可能会随着时间的改变而发生变化,其发生的可能性相对 CR 变化的可能性要大.对一个权限的 E_{PDA}来说,一旦产生后几乎不会发生改变,其发生频率最低.对于 E_{TDA}而言,由于对用户资格和能力的变化可以通过其来实现,其发生的频率比 E_{PDA}高,因而需优先判断该类属性表达式的变化.对于委托有效时间区间而言,因为每个委托关系都将存在委托时间,因为其发生的频率最高,将最先判断.因此在判断自动撤销类型时,先判断由委托有效时间区间引起的自动撤销,然后判断由权限的 E_{TDA}引起的自动撤销、由用户属性表达式引起的自动撤销、由 CR 引起的自动撤销,最后判断由 E_{PDA}引起的自动撤销.

2.4.2　自动撤销系统开销

按通常的方式,由委托有效时间区间引起的自动撤销会在每个系统最小时间点到来时对当前系统是否有需要撤销的委托进行判断并自动撤销.这在系统最小时间点间隔小、委托数量多的情况下,会带来很大的系统开销.在后面几种自动撤销中,如果系统在每个用户的委托先决角色改变、用户和权限的属性表达式改变后立即对涉及到的委托进行自动撤销触发条件判断,以决定和执行自动撤销,同样需要较大的系统开销.考虑到实际上用户在激活委托角色的会话结束之前是不能立即撤销的,因而可将自动撤销推迟到会话结束后进行.这样可大大减轻自动撤销判断和执行所需的系统负担,特别是由委托有效时间区间引起的自动撤销.

2.4.3　自动撤销安全性分析

将自动撤销推迟到会话结束后进行会带来一定的安全问题,特别是在多步委托的情况下.以图 2 为例说明.

图 2 为 1 个用户 u 与 1 个委托角色 r_{td}相关的 4 种状态:r_{td}所在会话的结束、r_{td}的重新激活、r_{td}的再次委托和 r_{td}作为先决角色以获得新委托角色.从图中可以看出,当 r_{td}在 t_2 时刻结束会话时,由于此时还没到委托有效区间结束时间 t_3,因而 r_{td}并不会被立即撤销.如果当 u 在 t_4 时刻重新激活 r_{td},则该操作是非法操作,因为此时刻 u 并不能拥有 r_{td}.同样,如果由于 $E_{\text{u.DA}}$变化、u 的 CR 变化和 $E_{r_{\text{td}}.\text{DA}}$的变化引起在 t_3 时刻用户不能再拥有 r_{td}.则 u 重新激活 r_{td}和将其再次委托出去就是非法操作.

从上面的分析可以看到,仅仅在会话结束时进行自动撤销判断及操作,会带来多步委托的安全问题.因而系统自动撤销时机可增加 2 个:激活委托角色的会话开始时刻和对委托角色多步委托操作时刻.即系统将在会话开始、会话结束和委托时刻进行是否自动撤销的判断和操作.

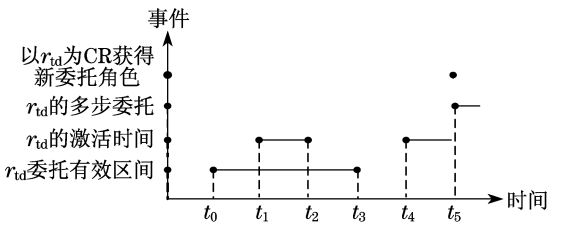

图 2　委托角色失活、激活、做为 CR 获得新的委托角色及多步委托示例

Fig.2　Example of r_{td}'s deactivation, activation, as a CR to get a new tdr and further delegation

但是也会出现另一个问题,图中当 u 以 r_{td}为先决角色在 t_5 时刻去获得新的委托角色时,该操作就是非法的,因为 u 在 t_3 时刻就不能拥有 r_{td}了.因而用户在获得委托角色时,对 CR 要判断该用户是否可以激活,只有可以激活的 CR 才能作为获得新委托角色的先决角色.定义 1 中 $U(r_{\text{td}})$函数中的 $c_{\text{anActive}}()$函数的作用就是判断用户是否可以激活 CR,这样也就可以避免出现上面所说的问题.

3　自动撤销算法及系统实现框架

3.1　自动撤销算法

表 1 实际上保存了系统中所有委托信息,包括委托角色、受托者、受托时限、受托先决角色和委托类型.其中 P 表示永久委托,T 表示临时委托.当某项委托撤销后,相应的委托数据项将从该列表中删除.

表 1　委托关系列表示例

Tab.1　Example of delegation relation list

委托角色	受托者	受托时限	受托先决角色	委托类型
dtr11	u21	[2009-01-01 10:00:00, 2009-01-10 12:00:00]	R2	P
dtr11	u10	[2009-01-03 08:30:00, 2009-01-03 12:00:00]	R10	T
dtr35	u3	[2009-08-15 20:00:00, 2009-09-14 09:00:00]	R12,R4	P

下面给出自动撤销(auto-revocation)算法.

输入:用户 u,用户当前角色集合 R,改变了 E_{DA}用户列表 $L_{\text{U.DAE}}$,改变了委托先决角色的用户列表 $L_{\text{U.CR}}$,改变了 E_{TDA}的委托角色列 $L_{\text{R.TDAE}}$,改变了

E_{PDA}的委托角色列表 $L_{R.PDAE}$,委托关系列表 L_d.

输出:修改后的 $L_{U.DAE}$,$L_{U.CR}$,$L_{R.TDAE}$,$L_{R.PDAE}$,L_d.

算法步骤如下:

(1) 对 R 中的每个角色执行如下循环;

(2) 在 L_d 中查找 u,r_i,找到转(3),否转(15).

(3) 如果(u,r_i)$\in c_{an\text{-}ARbyT}$成立转(12),否转(4).

(4) 在 $L_{R.TDAE}$中搜索 r_i.找到转(5),否转(6).

(5) 判断(u,p,r_i)$\in c_{an\text{-}ARbyPT}$是否成立,其中 $p\in p_{er\_d}(r_i)$.成立转(12),否转(6).

(6) 在 $L_{U.DAE}$中搜索 u.找到转(7),否转(8).

(7) 如果 $M_N(u,r_i)=T$,判断(u,r_i)$\in c_{an\text{-}ARbyDT}$是否成立,成立转(12),否转(8).如果 $M_N(u,r_i)=F$,判断(u,r_i)$\in c_{an\text{-}ARbyDP}$是否成立,成立转(12),否转(8).

(8) 在 $L_{U.CR}$中搜索 u,找到转(9),否转(10).

(9) 判断(u,r_{td})$\in c_{an\text{-}ARbyCR}$是否成立.成立转(12),否转(10).

(10) 在 $L_{R.PDAE}$中搜索 r_i.找到转(11),否转(15).

(11) 判断(u,p,r_i)$\in c_{an\text{-}ARbyPP}$是否成立,其中 $p\in p_{er\_d}(r_i)$.成立转(12),否转(15).

(12) 撤销 u 在 r_i 上的当前操作,删除 L_d 中该数据行,转(13).

(13) 如果 L_d 中不存在 u,则在 $L_{U.DAE}$和 $L_{U.CR}$中删除 u.转(14).

(14) 如果在 L_d 中不存在 r_i 且 $M_N(u,r_i)=T$,则在 $L_{R.PDAE}$中删除 r_i,如果在 L_d 中不存在 r_i 且 $M_N(u,r_i)=F$,则在 $L_{R.TDAE}$中删除 r_i.转(15).

(15) 然后取 R 中下一个 r_i;

(16) 返回 $L_{U.DAE}$,$L_{U.CR}$,$L_{R.TDAE}$,$L_{R.PDAE}$,L_d.

其中 $M_N(u,r_i)$函数的判断可通过查询委托关系列表的委托类型字段获得.上面算法中的判断顺序体现了前面讨论的自动撤销优先级.步骤(12)~(14)是将不存在委托关系的用户和角色从相应监控列表中删除,以减小监控列表的空间复杂性和查找的事件复杂性.设系统当前委托关系为 n,角色总数为 m,则该算法的时间复杂度为 $O(|n\times m|)$.

3.2 系统实现

图3给出了本模型实现的主体框架.图中分别列出了实现委托、用户撤销和系统自动撤销所必须的主要构件,构件说明见表2.其中委托和系统自动撤销共用部分控件.其中实箭头线表示控件与外部之间的数据交换,虚箭头线表示控件之间的数据交换.

图3的系统架构中,自动撤销以自动撤销部件(AR)为核心,它一方面获得事件监控(EM)和属性表达和委托先决角色监控部件(DAE&CRM)发送的消息,将其转发到属性表达和委托先决角色判断部件(DAE&CRJ),由其中的3个部件对由于 $E_{u.DA}$,CR,$E_{p.TDA}$和 $E_{p.PDA}$触发的自动撤销进行判断.另一方面,它通过EM传来的消息,将其转发到TJ进行委托有效区间触发的自动撤销判断.AR还负责将撤销后的结果写回用户-角色指派关系(URA)中.

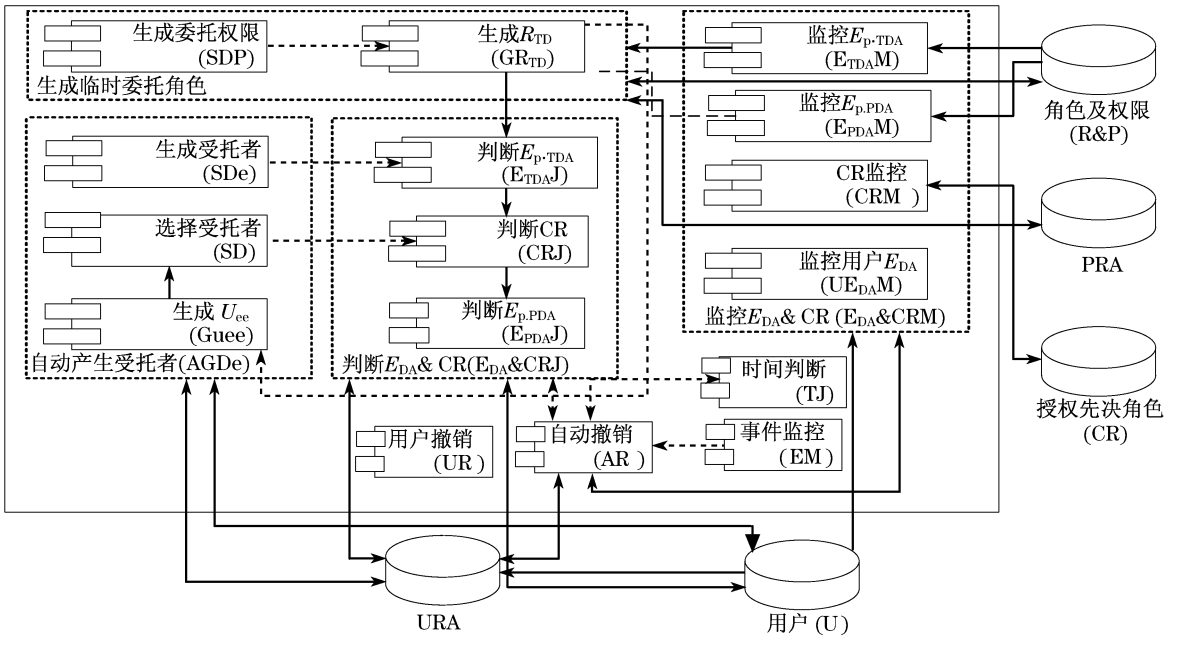

图3 系统实现框架

Fig.3 System architecture

4　结语

作为文献[12]的重要补充和进一步研究的成果,本文着眼于自动委托撤销操作,以期补上 $ABDM_A$ 模型在委托安全性上缺失的部分,达到一个完整安全的委托模型.在几种自动撤销中,由属性表达式的变化所引起的自动撤销最为重要,否则当委托者未能及时撤销不合格的受托者时,必然违背了系统的安全策略.本模型将委托过程的安全性作为优先考虑的因素,兼顾了系统效率.另外,本模型也为用户提供了手工撤销机制,以进一步提高系统效率.

表 2　系统架构组件功能说明

Tab.2　Functions of system components

组件	功能
R&P,PRA,CR,U,URA	保存用户、角色、权限基础数据和相应的 E_{DA},URA,PRA 和 CR,L_d 保存于 URA 中.
生成委托权限(SDP)	委托者确定确定受托者集合.
产生临时委托角色(GTDR)	产生一个委托角色 r_{td},生成相应的 E_{DA},CR 数据.
生成受托者(SD)	在确定受托者委托(DDD)类型的委托中选定受托者.
生成 U_{ee}($G_{U_{ee}}$)	系统自动产生符合委托要求的不确定受托者委托(UDD)类型的受托者集合.
选择受托者(SDE)	委托者在 U_{ee}集合中选定 1 个受托者.
$E_{P.TDA}$判断(E_{TDA}J)	临时委托自动撤销时对 $E_{p.TDA}$的判断
CR 判断(CRJ)	由 CR 引起的自动撤销判断
$E_{p.PDA}$判断(E_{PDA}J)	永久委托自动撤销时对 $E_{p.PDA}$的判断
用户 E_{DA}监控(UE_{DA}M)	监控用户的 E_{DA}变化情况,产生 $L_{U.DAE}$.
监控 $E_{p.TDA}$(E_{TDA}M)	监控权限的 E_{TDA}变化情况,产生 $L_{R.TDAE}$.
监控 $E_{p.PDA}$(E_{PDA}M)	监控权限的 E_{PDA}变化情况,产生 $L_{R.PDAE}$.
CR 监控(CRM)	监控用户 CR 变化情况,产生 $L_{U.CR}$.
事件监控(EM)	监控 URA 中产生的会话激活、失活和再委托事件,获得 L_d 列表.
时间判断(TJ)	判断委托有效时间区间是否失效.
自动撤销(AR)	执行自动撤销算法,结果写回 URA 中.
用户撤销(UR)	用户手工撤销操作.

参考文献:

[1] Ravi Sandhu, Edward Coyne, Hal Feinstein, et al. Role-based access control models[J]. IEEE Computer,1996,29(2):38.

[2] Barka Ezedin S. Framework for role-based delegation models [D]. Virginia:George Mason University. School of Information Technology and Engineering,2002

[3] Zhang Longhua, Gail Joon Ahn, Chu Beitseng . A rule-based framework for role-based delegation and revocation[J]. ACM Transactions on Information and System Security, 2003, 6 (3):404.

[4] Zhang Xinwen, Sejong Oh, Ravi Sandhu. PBDM: a flexible delegation model in RBAC[C]// Proceedings of the SACMAT'03. Como:ACM Press,2003:149 - 157.

[5] 赵青松,孙玉芳,孙波. RPRDM:基于重复和部分角色的转授权模型[J].计算机研究与发展,2003,40(2):221.
ZHAO Qingsong, SUN Yufang, SUN Bo. RPRDM: a repeated-and-part-role-based delegation model [J]. Chinese Journal of Computer Research and Development,2003,40(2):221.

[6] 翟征德.基于量化角色的可控委托模型[J].计算机学报,2006,29(8):1401.
ZHAI Zhengde. Quantified-role based controllable delegation model[J]. Chinese Journal of Computer,2006,29(8):1401.

[7] 孙波,赵庆松,孙玉芳. TRDM——具有时限的基于角色的转授权模型[J].计算机研究与发展,2004,41(7):1104.
SUN Bo, ZHAO Qingsong, SUN Yufang. TRDM-temporal role-based delegation model [J]. Chinese Journal of Computer Research and Development,2004,41(7):1104.

[8] Jacques Wainer, Akhil Kumar. A fine grained, controllable, user to user delegation method in RBAC [C] // Proceedings of SACMAT'05. Stockholm:ACM Press,2005:59 - 66.

[9] Roberto Tamassia, Yao Danfeng, William H Winsborough. Role-based cascaded delegation[C]// Proceedings of the SACMAT'04. New York:ACM Press,2004:146 - 155.

[10] Barka E, Sandhu R. Framework for agent-based role delegation [C] // Proceedings of IEEE International Conference on Communications. Glasgow:IEEE Press,2007:1361 - 1367.

[11] Abdallah A E, Takabi H. Integrating delegation with the formal core RBAC model[C]// Proceedings of Information Assurance and Security. Napoli:IEEE Press,2008:33 - 36.

[12] 叶春晓,吴中福,符云清,等.基于属性的扩展委托模型[J].计算机研究与发展,2006,43(6):1050.
YE Chunxiao, WU Zhongfu, FU Yunqing, et al. An attribute-based extended delegation model [J]. Chinese Journal of Computer Research and Development,2006,43(6):1050.