基于属性的委托撤销研究*)

叶春晓[1] 符云清[2] 钟 将[1] 冯 永[1]
(重庆大学计算机学院 重庆 400044)[1] (重庆大学网络教育学院 重庆 400044)[2]

摘 要 基于属性的委托模型中,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。在该模型中,委托撤销完成将委托出去的权限收回到委托者处的工作。与常见委托撤销不同,本文针对基于属性委托模型中委托过程的特点,提出了两种新的撤销模式:用于用户属性表达式变化引起的委托撤销;由于角色或权限属性表达式变化引起的撤销。这两种撤销模式能够确保当用户属性表达式不再满足委托权限或角色属性表达式时,系统能够自动地完成相应的撤销操作,保证了委托过程的安全性。

关键词 信息安全,访问控制,委托,撤销,属性

Study on Attribute-based Revocation in Delegation

YE Chun-Xiao[1] FU Yun-Qing[2] ZHONG Jiang[1] Feng Yong[1]
(College of Computer Science, Chongqing University, Chongqing 400044)[1]
(College of Network Education, Chongqing University, Chongqing 400044)[2]

Abstract In Attribute-based Delegation Model (ABDM), delegatee must satisfy both delegation prerequisite condition (CR) and delegation attribute expression (DAE) when assigned to a delegation role. In ABDM, revocation focus on how to revoke those delegated roles or permissions. Considering the characteristics of delegation operation in ABDM, this paper proposes two different revocations: revocation caused by the change of user' delegation attribute expression and revocation caused the change of delegation role's or permission's delegation attribute expression, which is differed from usual revocation. In ABDM, these two revocations can automatically revoke delegation roles or permissions from delegatee when their DAEs are no longer satisfy delegation roles' or permissions' DAEs either, thus guarantee the security of delegation.

Keywords Information security, Access control, Delegation, Revocation, Attribute

1 引言

委托(delegation)可实现将委托者(delegator)的全部或部分权限指派给受托者(delegatee)。撤销是委托的逆过程,其作用是将受托者获得的委托权限收回[1]。

在现有的委托模型中,RBDM[2,3]提出了基于角色的委托模型,首次将角色概念引入到委托模型中。RBDM 详细讨论了委托撤销,将其分为系统撤销和用户撤销两种。前者指撤销操作由系统条件自动触发,后者指撤销操作由用户手工完成。RBDM 对后者又进行了进一步分类。

RDM2000[4]模型支持层次角色和多步委托,该模型对委托撤销进行了详细分类。PBDM[6]是为解决权限的部分委托而提出的模型,支持部分委托模式。但该模型主要针对委托操作,对委托撤销未做深入研究。RPRDM[7]主要就 RBDM 和 RDM2000 中不能进行重复和部分权限委托进行改进和扩展,定义了四种委托撤销。

$ABDM_X$[8]和 $ABDM_A$[9]主要从两个不同的方面就基于属性的委托进行讨论,其中并没有详细讨论委托撤销的问题。本文在文[9,10]基础上,详细讨论了将属性引入到委托过程后,委托撤销所具有的特点并引入新的委托撤销类型。

2 基于属性的委托撤销

2.1 撤销触发机制

系统自动撤销有多种机制,本文不详细讨论常见的自动撤销机制,将重点讨论由属性表达式的变化引起的撤销。

$ABDM_A$ 模型中,当用户的属性表达式不满足临时委托角色的属性表达式后,该用户不应当继续拥有该临时委托角色。引起用户的属性表达式(记为 u. DAE)不再满足临时委托角色属性表达式(记为 tdr. DAE)的原因主要有两个:用户的属性表达式发生了变化和临时委托角色中的委托权限的属性表达式(记为 p. DAE,包括 TDAE 和 PDAE)发生变化。由于表明了用户的资格和能力,因而 u. DAE 一旦变化,不再满足临时委托角色的 tdr. DAE 的要求后,用户就丧失了执行委托权限的资格和能力。同样,委托权限的 p. DAE 发生变化后,表明其对用户的 u. DAE 有新的要求,同样会使得不再满足新要求的用户不再拥有该权限。

用户的 u. DAE 和权限的 p. DAE 的变化频度不一样,一般情况下 u. DAE 的变化频度要高于 p. DAE 的变化频度。实际应用中,用户的状态、资格和能力会随着时间的推移不断变化,这使得 u. DAE 也发生相应的变化。而权限和角色在角色工程阶段产生后相对比较固定,因而使得 p. DAE 变化的可能性相对较小。显然,为保证系统安全,u. DAE 和 p. DAE 的变化均需要系统管理员来控制。

以下讨论不同属性表达式变化引起的委托撤销。

2.2 由用户属性表达式变化引起撤销

当用户的 u. DAE 发生变化后,会产生两种可能:变化后

*)本文受教育部博士点基金资助(基金号:20040611002)。

的 u. DAE 还满足原来临时委托角色的 tdr. DAE；变化后的 u. DAE 不再满足临时委托角色的 tdr. DAE。前一种情况不会引起委托撤销操作，但后一种情况肯定会引起委托操作。例如，当项目经理 *QE* 将权限 inspect-java-code(p. DAE: language=java AND programming experience≥2)委托给用户 *u*(u. DAE: language=java AND programming experience=2)。若 QE 的出差时间很长，以致于 u. DAE 发生了变化：language=java AND programming experience=3。显然，此时 u. DAE 还是满足权限的 p. DAE。这种情况下不会引起委托撤销操作。

再考虑一个例子：设 design-java-code(p. DAE: language=java AND programming experience≥2 AND code quality=high)。该权限除了要求用户熟悉 java 语言、具有两年以上的编程经验外，还要求其设计的代码质量高。若用户在某段期间内其代码质量降低，则不具备继续设计代码的资格，自然也就不能再拥有该权限了。

由于临时委托权限的 p. DAE 最终是通过临时委托角色的 tdr. DAE 表现出来，因而当用户的属性表达式 u. DAE 发生变化后是否会引起委托撤销操作，主要是看变化后的 u. DAE 是否还满足临时委托角色的 tdr. DAE，即 u. DAE tdr. DAE 是否成立。

为了定义由用户 u. DAE 变化引起的自动委托撤销，此处先定义一个判断用户属性表达式是否改变的函数。

定义 1

$$\text{DAEModified}(u)=\begin{cases}\text{true}, & \text{u. DAE 发生变化}\\ \text{false}, & \text{u. DAE 未发生变化}\end{cases}$$

同样，自动委托撤销也存在时机问题。当该临时委托角色还处于活动时，用户的 u. DAE 或权限的 p. DAE 发生变化后，使得用户的 u. DAE 不再满足临时委托角色的 tdr. DAE。若此时将该临时委托角色撤销掉，这就是立即撤销。反之，当需要撤销的临时委托角色所处的会话结束后，将该临时委托角色自动撤销掉，此时的撤销为延迟撤销。

定义 2 由用户的 u. DAE 变化引起的自动延迟委托撤销需满足下面关系：

$$\text{can-revoke-caused-by-delegatee}\subseteq U\times TDR,$$

$$(u,tdr)\in \text{can-revoke-caused-by-delegatee}\Leftrightarrow (tdr\in \text{roles}(u))\wedge\neg(\text{u. DAE}\geqslant \text{tdr. DAE})\wedge(\text{DAEModified}(u)=\text{true})。$$

其中 U 为用户，TDR 为临时委托角色。

上面定义表明，当用户 u. DAE 发生变化后且不再满足其拥有的委托角色的 tdr. DAE，该 tdr 将自动从 *u* 中撤销。

2.3 由权限属性表达式变化引起的撤销

当 p. DAE 发生变化后也会产生两种可能：受托者的 u. DAE 满足变化后的 p. DAE 和受托者的 u. DAE 不满足变化后的 p. DAE。例如，当权限 inpsect-java-code 的 p. DAE 变为：language=java AND programming experience≥3 时，若用户 *u* 的 u. DAE 仍然为 language=java AND programming experience=2 时，此时应当将该权限从用户 *u* 中撤销。而若该权限也同时委托给另一个用户 u'，该用户的 u'. DAE 为 language=java AND programming experience=4，则该 u'. DAE 仍然满足变化后的权限的 p. DAE，u' 继续拥有该权限。

由于权限的 p. DAE 实际上是通过临时委托角色的 tdr. DAE 体现出来的，因而主要是判断受托者的 u. DAE 是否满足变化后临时委托角色的 tdr. DAE。

同理，此处定义一个判断权限属性表达式是否改变的函数。

定义 3

$$\text{DAEModified}(p)=\begin{cases}\text{true, p. DAE 发生变化}\\ \text{false, p. DAE 未发生变化}\end{cases}$$

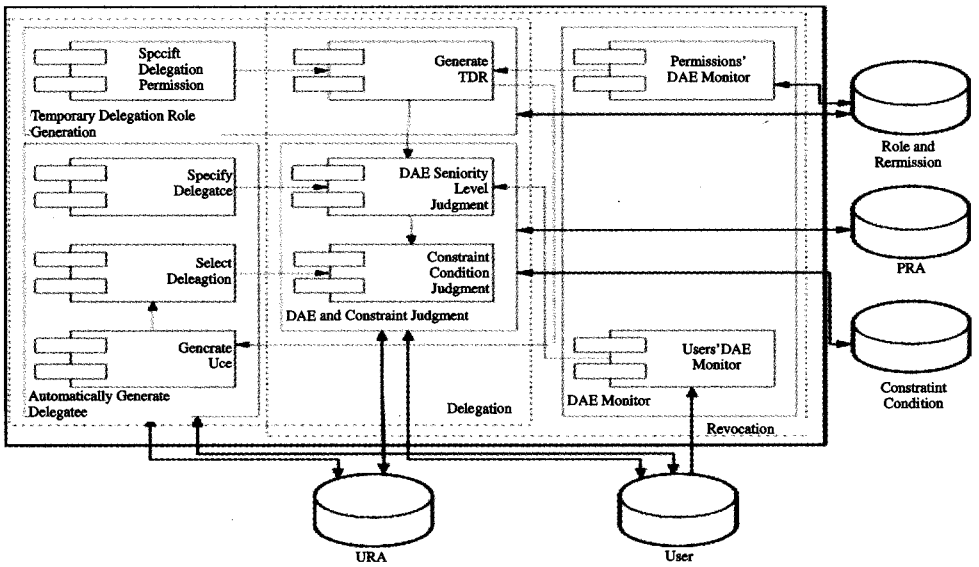

图 1 委托和撤销部件

定义 4 由权限的 p. DAE 变化引起的自动延迟委托撤销需满足下面关系：

$$\text{can--revoke-caused-by-permission}\subseteq U\times P\times TDR,$$

$$(u,p,tdr)\in \text{can--revoke-caused-by-permission}\Leftrightarrow(tdr\in \text{roles}(u))\wedge\neg(\text{u. DAE}\geqslant \text{tdr. DAE})\wedge(p\in \text{per\_d}(tdr))\wedge(\text{DAEModified}(p)=\text{true})$$

其中 U 为用户，P 为权限，TDR 为临时委托角色。

上面定义表明，当 *tdr* 中的权限 p. DAE 发生变化后，如果用户 *u* 的 u. DAE 不再满足临时委托角色 *tdr* 的属性表达式 tdr. DAE，且拥有该权限的临时委托角色处于失活状态，该 *tdr* 将自动从 *u* 中撤销。

根据 ABDM_A 定义，一个权限 *p* 可以被委托给多个用户。这样，权限 *p* 可能存在于不同的临时委托角色中。当 *p* 的属性表达式发生改变后，并不是所有包含该权限的临时委托角

色都被撤销。只有不再满足变化后的临时委托角色的用户才不再具有该角色，这一特性在定义中体现出来了。

对于单调权限，由于其 PDAE 和 TDAE 相同，因而其中任意一个属性表达式发生变化均可能影响到受托者属性表达式不再满足该权限属性表达式，会引起撤销。对于非单调权限，同样无论 PDAE 或 TDAE 发生变化，也可能造成受托者属性表达式不满足权限的属性表达式，引起撤销操作。因而，本文并没有详细区分权限属性表达式发生变化的类型。

需要说明的是，无论受托者是通过确定性委托 DDD 或是通过非确定性委托 UDD 获得委托权限，当发生上面两种类型的属性表达式改变后，同样可能引起撤销操作。因而，can-revoke-caused-by-delegatee 和 can-revoke-caused-by-permission 可适用于 $ABDM_A$ 中 can-delegateDP、can-delegateDT、can-delegateU 三种委托关系。

3 系统实现

此处给出包含委托和撤销所需部件的主要构件，其中的委托部件在文[9]中已经给出，此处再次列出主要是说明撤销与委托之间会共用一些部件。图 1 中用虚线围出委托和撤销所涉及到的部件。

用户的属性表达式变化引起的撤销各个部件之间的关系为：

1)Users' DAE monitor 部件检测到用户 u 的属性表达式变化；

2)Automatically Generate Delegatee 部件将 URA 中用户 u 与临时委托角色 tdr 之间的指派关系传送到 DAE Seniority Level Judgment 部件；

3)User's DAE monitor 部件将变化后 u 的属性表达式传送到 DAE Seniority Level Judgment 部件；

4)由 Seniority Level Judgment 部件判断变化后 u 的属性表达式是否满足临时委托角色 tdr 属性表达式；

5)如果 u 的属性表达式不满足临时委托角色 tdr 属性表达式，则撤销 u 和 tdr 之间的指派关系，结果写回 URA 中。

权限属性表达式变化引起的撤销各个部件之间的关系为：

1)Permissions' DAE monitor 部件检测到权限 p 的属性表达式变化；

2)Permissions' DAE monitor 部件将 p 变化后的属性表达式传送到 Generate TDR 部件；

3)如果 p 为临时委托角色 tdr 中的权限，则 Generate TDR 部件将产生新的临时委托角色属性表达式；

4)DAE Seniority Level Judgment 部件将重新比较指派该临时委托角色 tdr 的属性表达式和所有指派该 tdr 角色的用户的属性表达式；

)DAE and Constraint Judgment 将撤销不满足的用户和 tdr 之间的指派关系，结果写回 URA 中。

表 1 委托和撤销服务部件的功能

Component	Function
User's DAE monitor	Monitor which user's DAE has changed
Permission's DAE monitor	Monitor which permission's DAE has changed

表 2 用户及其 DAE

User	DAE
Cxy	familiar_test_tool≥1 AND testing_experience≥3 AND language=JAVA AND database=ORACLE AND familiar_with_test_theory=yes AND current_program _module=A
Xmw	language=VB AND database=ORACLE AND familiar_with_test_theory=yes AND current_program _module =none
Yqf	familiar_test_tool≥1 AND testing_experience≥3 AND language=JAVA AND database=ORACLE AND current_program _module =A AND familiar_with_test_theory=yes
Jz	language=JAVA AND database=ORACLE AND current_program _module =none
Kgw	familiar_test_tool≥1 AND language=JAVA AND database =ORACLE AND current_program _module =B

1. 撤销实例：在某段时间内，由于测试人员较少，使得大量测试工作无法进行下去。采取的措施是临时将 $p1$、$p2$、$p3$ 指派给其他人员，如程序员 P、数据库程序员 DP 等，由他们和测试员 T 一起来完成对程序的测试工作。按照 ABMDA 规定的委托操作，只有 Cxy、Yqf 获得这三个权限。委托过程在文[9]中详细介绍，此处不再赘述。将详细说明当用户或权限表达式发生变化后系统如何进行撤销操作。

·用户属性表达式的改变

设想当前进行测试的为 B 模块，显然按照测试的一般原则，测试人员是不能参与 B 模块的开发的。当测试过程中发现模块有错误后，需要修改模块，现假设用户 Cxy 现在从事 B 模块的修改工作。显然，此时 Cxy 的 DAE 应改为：familiar_test_tool≥1 AND testing_experience≥3 AND language=JAVA AND database=ORACLE AND familiar_with_test_theory=yes AND current_program _module=B。显然，此时用户 Cxy 的属性表达式不能满足权限 $p1$、$p2$、$p3$ 属性表达式，此时应当撤销用户所拥有的上述三个权限。

·权限属性表达式的改变

随着测试工作的进行，需要用户对 SQL SERVER 数据库非常熟悉。这样，$p1$、$p2$、$p3$ 的属性表达式将分别改变为：language=JAVA AND testing_experience ≥2 AND database =SQL SERVER AND familiar_with_test_theory=yes AND current_program _module≠B、familiar_test_tool≥1 AND language=JAVA AND database=SQL SERVER AND current_program _module≠B、familiar_test_tool≥1 AND testing_experience≥2 AND language=JAVA AND database=SQL SERVER AND familiar_with_test_theory=yes AND current_program _module≠B。显然，如果用户 Cxy 和 Yqf 对 SQL SERVER 数据库不熟悉的话，其属性表达式不能满足 $p1$、$p2$、$p3$ 的属性表达式，需要将上述三个权限从用户处撤销掉。

总结与进一步的工作 撤销是委托模型必不可少的功能，作为 $ABDM_A$ 模型的重要组成部分，本文和文[9]分别从撤销和委托两个方面说明了基于属性的委托模型中委托与撤销操作。基于属性的撤销主要考虑了当用户和权限属性表达式发生变化后，$ABDM_A$ 如何进行撤销操作。文中给出了与撤销相关的部件，并给出了实际例子。本文只给出了最基本的撤销操作，进一步的工作将包括如何进行基于属性的多步和多重撤销。

（下转第 291 页）

pHost = gethostbyname((char *)LocalName)); // 获取本地 IP 地址

addr_in.sin_addr = *(in_addr *)pHost->h_addr_list[0]; //IP

addr_in.sin_family = AF_INET;

addr_in.sin_port = htons(57274);

bind(sock, (PSOCKADDR)&addr_in, sizeof(addr_in)); // 把原始套接字 sock 绑定到本地网卡地址上[4,5]。

此时可通过 recv()函数从网卡接收数据。接收到的原始数据包存放在缓存 RecvBuf[]中,缓冲区长度 BUFFER_SIZE 定义为 65535。根据前面对 IP 数据包进行分析,获取源地址与目标地址:

int ret = recv(sock, RecvBuf, BUFFER_SIZE, 0); // 接收原始数据包信息

if (ret > 0) {

ip = *(IP*)RecvBuf; // 对数据包进行分析,并输出分析结果

if (inet_ntoa(*(in_addr*)&ip.SrcAddr))<>addr_in.sin_addr)

or (inet_ntoa(*(in_addr*)&ip.DstAddr))<>addr_in.sin_addr)

or (inet_ntoa(*(in_addr*)&ip.SrcAddr))<>server.sin_addr)

or (inet_ntoa(*(in_addr*)&ip.DstAddr))<>server.sin_addr) //判断源地址、目标地址是否为本机或服务器的 IP,这里若为了程序的灵活性,可采用 IP 地址的黑白名单的办法。

{close(sock);//禁止通信。}}

结束语 计算机辅助测验(CAT)具有广泛的应用前景,在考核过程中利用计算机自身的监控能力来进行监考是人力所无法达到的。在实际应用中,要注意及时卸载钩子,以免占用资源。采集数据时不要把网卡设置为混杂模式,否则收到网络上的其他数据,达不到监视目的,同时使网速变慢。凡有违规行为,均要记录并及时警告考生,达到某种程度,应中止该考生的考试。以上方法在 100M 局域网内(61 台计算机),Windows2000 下测试通过。

参 考 文 献

1 Richter J(美). Windows 核心编程[M]. 王建华,译. 北京:机械工业出版社,2000

2 赵新宇,林作铨. 具有监控能力的 Agent 模型[J]. 计算机科学,2006(3):11~17

3 陈少辉,张艳宁,刘艳玲. 基于封包截获技术的个人防火墙核心驱动技术[J]. 计算机工程,2007(6):123~125

4 胡金初. 数据包时延及控制策略的研究[J]. 计算机科学,2006(11):52~53

5 刘翔,席守卿,吴昕怡,等. Windows2000 系统中网络数据包截获方法[J]. 武汉理工大学学报,2002(24):88~89

(上接第 276 页)

表 3 权限及其 DAE

Permission	DAE
P1:design test case and data	Requirements: professional in JAVA, two years' test experience, professional in ORACLE, familiar with test theory, not a member of the team of module B. language=JAVA AND testing_experience ≥2 AND database=ORACLE AND familiar_with_test_theory=yes AND current_program _module≠B
P2:config test environment	Requirements: familiar with at least one tool, professional in JAVA and ORACLE, not a member of the team of module B. familiar_test_tool≥1 AND language=JAVA AND database=ORACLEANDcurrent_program _module≠B
P3:perform test	Requirements: familiar with at least one tool, professional in JAVA and ORACLE, familiar with test theory, two years' test experice, not a member of the team of module B. familiar_test_tool≥1 AND testing_experience≥2 AND language=JAVA AND database=ORACLE AND familiar_with_test_theory=yesANDcurrent_program _module≠B

参 考 文 献

1 Sandhu R, Coyne E, Feinstein H, et al. Role-based Access Control Models. IEEE Computer, 29(2): 38~47

2 Barka E, Sandhu R. Framework for Role-based Delegation Models. In: Proc. of 16th Annual Computer Security Application Conference (ACSAC2000). New Orleans, USA: IEEE Computer Society Press, 2000

3 Barka E, Sandhu R. A role-based delegation model and some extensions. In: Proc. of 23rd National Information Systems Security Conference (NISSC 2000). Baltimore, USA, 2000

4 Zhang Longhua, Ahn Gail-Joon, Chu Bei-Tseng. A rule-based framework for role-based delegation. In: Proc. of SACMAT'01. Chantilly, VA, USA: ACM Press, 2001

5 Tamassia R, Yao Danfeng, Winsborough W H. Role-based cascaded delegation. In: Proc. of the SACMAT'04. Yorktown Heights, New York, USA: ACM Press, 2004

6 Zhang Xinwen, Oh Sejong, Sandhu R. PBDM: A Flexible Delegation Model in RBAC. In: Proc. of the SACMAT'03. Como, Italy: ACM Press, 2003

7 赵青松,孙玉芳,孙波. RPRDM:基于重复和部分角色的转授权模型. 计算机研究与发展,2003,40(2):221~227

8 Ye Chun-xiao, Fu Yun-qing, Wu Zhong-fu. An Attribute-based Delegation Model and Its Extension. Journal of Research and Practice in Information Technology, 2006,38(1):3~17

9 叶春晓,吴中福,符云清,等. 基于属性的扩展委托模型. 计算机研究与发展, 2006,43(6):1050~1057

10 叶春晓. 基于角色访问控制(RBAC)中属性约束委托模型研究:[博士论文]. 重庆:重庆大学,2005