基于框架的形式化商务安全策略模型

温红子[1,3],周永彬[2,3],卿斯汉[1,3]
(1. 中国科学院软件研究所信息安全技术工程研究中心,北京 100080;2. 中国科学院软件研究所信息安全国家重点实验室,北京 100080;
3. 中国科学院研究生院,北京 100039)

摘 要: 商务信息系统安全的核心目标是维持系统数据的完整性. 虽然研究人员已提出许多完整性安全原则,但至今仍然缺乏一种系统的商务安全策略. 本文所提出的基于框架的形式化商务安全策略模型(FB-FCSM)是一个集成多种完整性原则的系统性商务完整性模型,具有良好的兼容性和扩展性,是 Clark-Wilson 完整性安全策略的精化.

关键词: 信息系统安全; 形式化商务安全策略; 完整性
中图分类号: TP309 **文献标识码**: A **文章编号**: 0372-2112 (2005) 02-0222-05

A Formal Commercial Secure Policy Model Based on Framework

WEN Hong-zi[1,3], ZHOU Yong-bin[2,3], QING Si-han[1,3]
(1. *Engineering Research Center for Information Security Technology, Institute of Software, Chinese Academy of Sciences, Beijing* 100080, *China*;
2. *State Key Laboratory of Information Security, Institute of Software, Chinese Academy of Sciences, Beijing* 100080, *China*;
3. *Graduate School Chinese Academy of Sciences, Beijing* 100039, *China*)

Abstract: The key of security on commercial information systems is to preserve the data integrity within them. Although many integrity principles have been discussed, there is still lack of a systematic commercial secure policy model. A Formal Commercial Secure Policy Model Based on Framework (FB-FCSM) is proposed, which integrates multi-integrity-principle into one systematic commercial secure model. The model has not only good compatibility but also sound scalability, and is a refinement of Clark-Wilson Integrity Secure Policy.

Key words: information system security; formal commercial secure policy model; integrity

1 引言

商务信息系统的安全性需求主要体现在完整性方面[1,2],其目标在于"…无论何时,都要确保数据有一个恰当的物理表示,(数据)是一个恰当的信息语义表示,并且要确保授权用户和信息处理资源对数据执行了正确的处理操作"[3]. 基于上述认识,人们提出了一系列用以实现系统完整性的原则,诸如良构变换过程、授权执行、记账以及职责隔离等[1],这些原则分别反映了系统完整性的一个重要侧面,但是由于缺乏一种把这些原则集成为一体的机制,所以无法对商务系统的安全性进行系统性的评估. 另一方面,商务系统在性质和规模上的多样性也要求作用于其上的完整性安全策略具有依据商务应用的性质和规模进行灵活调整所必需的兼容性和扩展性. 因此,本文首先用一种框架机制来显式地刻画了这些完整性原则之间的内在联系,然后在此基础上提出了一种新颖的基于框架的形式化商务安全策略模型(FB-FCSM),它可以很好地解决商务应用系统安全中存在上述问题.

2 FB-FCSM 原理

文献[3]的完整性目标定义中允许授权用户直接处理数据. 事实上,由于用户的行为具有很大的不确定性,这种不确定性可能会有意无意地损害系统的完整性,因而就规定授权用户不能直接存取修改数据,仅仅只能通过变换过程来操纵数据. 由此可知,变换过程在商务系统完整性中处于核心地位. 完整性目标中的信息处理资源通常表述为变换过程,它是由预定义操作集中的操作依据特定的语义要求顺序执行的操作序列,并要求它具有把系统数据从一个一致状态转换到下一个一致状态的良构特性[1].

变换过程的良构特性是由认证代理通过对变换过程进行认证来确认. 因此,变换过程维持商务系统一致性的信心主要来源于认证代理对于该变换过程的认知程度,即以认证代理所具有的关于该过程的"知识"为前提. 这与现实是相符的:因为对于一个企业组织而言,如果没有人或者机构彻底理解组成企业商务应用的变换过程的安全语义,或者无法从它方获得足够的关于该变换过程的信任并对其负责,那么这个商务过程的安全性显然是不可接受的.

具体来讲,商务完整性安全策略框架由用户完整性、过程完整性、过程执行完整性、职责隔离完整性和日志完整性原则构成(如图 1 所示,图中所示操作编号对应于本文 3.3 部分中的状态转移函数的编号,具体操作语义由函数定义来确定).

收稿日期:2004-03-02;修回日期:2004-05-25
基金项目:国家自然科学基金(No. 60083007/60373039);国家重点基础研究发展规划(973)(No. G1999035802)

其中,用户身份完整性指系统中的用户都是受控生成并且经过身份认证,由用户生成(1)和用户身份认证(2)两个操作阶段体现;过程完整性指变换过程具有良构特性,即具有维持数据一致性状态的能力,对变换过程的认证(3)(6)是确保过程完整性的根本手段;职责隔离(2)(7)指一个变换过程的所有操作可以分为几个不同部分,要求各部分为不同的用户所分别执行.本文把用于实施职责隔离的变换过程称为可分解变换过程,通过对由可分解变换过程所形成的各个变换过程片的授权(7)进行控制来达到职责隔离的目的;过程授权执行完整性则指被执行的变换过程必须是经过认证和授权的,由变换过程授权(4)(7)和用户身份授权(5)两部分来保障,变换过程执行(8)(9)时必须参考该变换过程的授权情况(图中使用虚线来表示);最后,日志完整性由系统日志来体现(10).

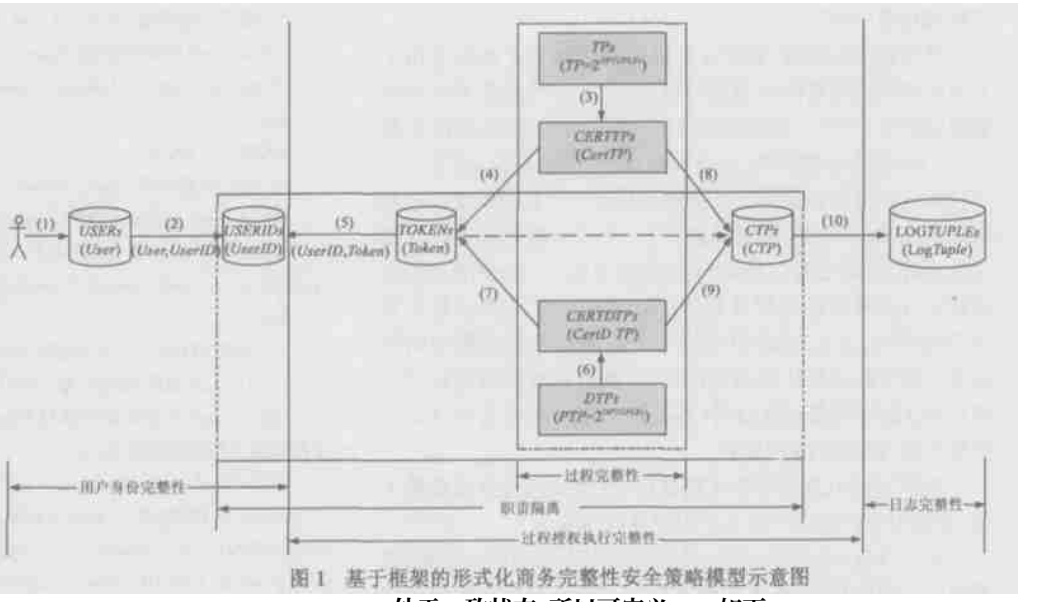

图 1　基于框架的形式化商务完整性安全策略模型示意图

3　FB-FCSM 描述

FB-FCSM 定义了一个状态机系统[2,4]. 为了更清晰地描述模型,这里引入了与安全特性无关的辅助函数. 在模型中,若设某状态变量 S 当前状态为 S,则后继状态为 S^*.

3.1　模型元素

受控数据项(*CDI*):系统中受数据完整性策略所约束的数据项,所有的受控数据项构成了受控数据项集合(*CDIs*).

非受控数据项(*UDI*):系统中不被数据完整性策略所约束的数据项(例如新输入系统中的数据项),所有的非受控数据项构成了非受控数据项集合(*UDIs*).

CDIs 一致状态(V_{CDIs}):本文采用 Ramakrishnan[6]的关系定义来描述 *CDIs* 上的一致性状态条件. $r(A_1:D_1, \cdots, A_n:D_n)$ 是一个数据关系模式,在这里表现为数据完整性策略. 对于每一个 $A_i, 1 \leq i \leq n$, Dom_i 是 D_i 域的值集,一个数据关系 r(满足该模式中的域约束,也称为实例)是一个有 n 个字段的元组集:$r:\{\langle a_1:d_1, \cdots, a_n:d_n\rangle \mid d_1 \in Dom_1, \cdots, d_n \in Dom_n\}$,其中符号 $\langle \ldots \rangle$ 用来表明一个元组的字段. 关系模式是状态独立的,而关系(实例)则是状态相关的. 当 $CDIs = \{D_1, \cdots, D_n\}$ 时,则称 $r:\langle a_1:d_1, \cdots, a_n:d_n\rangle$ 为受控数据关系,记为 r_{CDIs}. 记 R_{CDIs} 为作用在 *CDIs* 上的所有受控数据关系的集合,称之为受控数据关系集. 当 R_{CDIs} 中所有关系都成立时,则称 *CDIs* 一致状态 V_{CDIs} 的值为真.

完整性验证过程(*IVP*):系统中上述受控数据关系是通过完整性验证过程(*IVP*)来体现的. 当系统通过一个 *IVP* 的验证后,也就表明此时系统中与该 *IVP* 相关的 *CDIs* 完整性约束处于一致状态. 所以可定义 *IVP* 如下:

$$IVP: R_{CDIs} \rightarrow 2^{R_{CDIs}}$$

所有的 *IVP* 构成了完整性验证过程集 *IVPs*,只有当系统通过所有的 *IVP* 验证后,才可以确信系统 V_{CDIs} 的值为真.

操作(*OpTuple*):一个操作是一个形如(*OpModel*, *CDI*)的二元结构,其中操作模式 *OpModel* 为正常的读、写等操作模式,系统中的一个操作标明了变换过程是以何种模式来存取某一受控数据项值的. 相应的操作集合和操作模式集合分别称为 *OPTUPLEs* 和 *OPMODELs*.

变换过程(*TP*):系统中变换过程的目的在于把 *CDIs* 从一个完整性状态转换到下一个完整性状态. *TP* 的结构可以形式地定义为 $TP = \{OpTuple \mid OpTuple \in OPTUPLEs\}$,表示一个变换过程由一个 *OPTUPLEs* 的子集构成①. 当一个变换过程通过认证时,便称之为认证变换过程(*CertTP*). 所有认证变换过程构成了认证变换过程集(*CERTTPs*).

变换过程认证知识(*TPKnow*):是形如(*Userj*, 2^{TPs})的二元组,表示认证代理具有足够的用于认证变换过程集 2^{TPs} 中的变换过程的知识. 系统中所有的变换过程认证知识构成了变换过程认证知识集 *TPKNOWs*.

UDI 认证知识(*UDIKnow*):是形如(*TP*, $2^{UDICDIs}$)的二元组,*UDICDIs* 为由非受控数据项值映射集,该集中的元素 *UDICDI* 为 *UDI* 到 *CDI* 上的一个许可映射. 每个 *UDIKnow* 都与一个 *TP* 相关联,标明了与该 *TP* 相关的所允许的所有 *UDI* 到 *CDI* 上的有效转换.

激活变换过程(*CTP*):指当前通过执行而转变为活动的变换过程,系统中某一时刻所有的活动变换过程组成了激活变换过程集(*CTPs*).

可分解变换过程(*DTP*):是指可以分解为子部分以用于

① 严格意义上讲,构成一个变换过程的所有操作之间常有操作之间常常蕴涵着一些结构约束关系,比如操作之间的序关系等. 如果把这些约束标识为一个约束集 S,则这时一个 *TP* 可以严格定义为:$TP = (2^{OPTUPLEs}, S)$. 为了行文简洁起见,本文通常不刻意强调这种约束关系.

实施职责隔离完整性的变换过程,每个子部分称作为可分解变换过程片(*PTP*).相应的集合分别为可分解变换过程集(*DTPs*)和变换过程片集(*PTPs*).显然有,也就是说,任意一个可分解变换过程必定为变换过程.但是,一个普通的变换过程未必可以被用于实施职责隔离.当一个可分解变换过程片通过认证时,便称之为认证可分解变换过程片(*CertPTP*),所有的认证可分解变换过程片构成了认证可分解变换过程片集(*CERTPTPs*).本文约定:只有当一个可分解变换过程中的所有可分解变换过程片都通过认证后,该可分解变换过程才可转化为认证可分解变换过程(*CertDTP*),相应的集合为认证可分解变换过程集(*CERTDTPs*).

用户(*User*):在系统中实施认证、授权和执行等功能的主体,系统中所有的用户形成了用户集(*USERs*).

用户身份(*UserID*):当一个用户通过系统身份认证后,就获得一个用户身份,系统中所有的用户身份形成了用户身份集(*USERIDs*).另外,为了标明用户和用户身份之间的关系,本文引入了用户身份元组(*UserIDTuple*),它是一个形如(*User*, *UserID*)的二元结构.系统中所有的用户身份元组 *UserIDTuple* 构成了用户身份元组集 *USERIDTUPLEs*.

令牌(*Token*):它是一个 *TPs* 的标识符集合,用户执行一个 *TP* 的先决条件是与该用户身份相关的令牌中含有该 *TP* 的标识符.系统中所有的 *Token* 构成了令牌集(*TOKENs*).用户身份和令牌之间的关系使用用户身份令牌元组(*AuthTuple*)来描述,其结构为(*UserID*, *Token*).系统中所有的用户身份令牌元组构成了用户身份令牌元组集(*AUTHTUPLEs*).

3.2 辅助函数

(1)用户分类函数:

$$\text{UserClass}: USERs \rightarrow USERCLASSs$$

用以甄别系统中的用户类型,这里 $USERCLASSs = \{AuthUser, ExecUser, SysUser\}$,只能为 *AuthUser*、*ExecUser* 和 *SysUser* 之一. *AuthUser* 类用户通常为安全官员、等特权用户,或者为允许去认证实体的代理.特权用户可以更改一个实体和其他实体之间的关联列表,但同时要求他不可以具有任何有关那个认证实体的执行权限; *ExecUser* 类用户为执行用户,亦即执行具体变换过程的用户; *SysUser* 为用于执行系统日志行为的系统内置用户,通常不可更改.

(2)变换过程分割函数:

$$\text{DTPPart}: TPs \rightarrow 2^{PTPs}$$

式中 *TPs* 为变换过程集, *PTPs* 为变换过程片集,变换过程分割函数用于把一个变换过程划成几个互不相同的变换过程片,并使得各个变换过程片中具有两两不同的操作集.

(3) *CDIs* 完整性评估函数:

$$\text{Evaluate}: IVPs \rightarrow RESULTs$$

用以评估 *IVP* 运行结果, $RESULT = \{\text{TURE}, \text{FALSE}\}$,TRUE 表示合法状态,FALSE 则为异常状态.TRUE 和 FALSE 是布尔值,要求作用于其上的运算符合布尔运算法则.

3.3 状态转移函数

(1)用户生成函数 *AddUser*(*useri*, *userj*)

(语义: *userj* 请求增加用户 *useri*.)

if $UserClass(userj) = AuthUser \wedge useri \notin USERs$

then

$USERs^{*} = USERs \cup \{useri\}$

(2)用户认证函数 *CertUser*(*useri*, *userid*, *userj*)

(语义: *userj* 请求认证用户 *useri* 的身份 *userid*.)

if $UserClass(userj) = AuthUser \wedge useri \in USERs \wedge userid \in USERIDs \wedge \exists useridtuple = (useri, userid) \wedge useridtuple \notin USERIDTUPLEs$

then

$USERIDTUPLEs^{*} = USERIDTUPLEs \cup \{useridtuple\}$

(3) *TP* 认证函数 *CertTP*(*tp*, *userj*)

(语义: *userj* 请求认证变换过程 *tp*,要求 *tp* 具有对可能出现的非受控数据进行处理的能力.)

if $UserClass(user) = AuthUser \wedge tp \in TPs \wedge tp \notin DTPs \wedge tp \notin CERTTPs \wedge \exists tpknow \in TPKNOWs \wedge userj = tpknow.1 \wedge tp \in tpknow.2 \wedge \exists udiknow \in UDIKNOWs \wedge tp \in udiknow.1 \wedge [[udiknow.2| = 0] \vee [|udiknow.2| > 0 \wedge [\forall udicdi: UDICDI| udicdi \in udiknow.2 \cdot udicdi \in UDICDIs]]]$

then

$CERTTPs^{*} = CERTTPs \cup \{tp\}$

(4) *TP* 授权函数 *AuthTP*(*tp*, *token*, *userj*)

(语义: *userj* 请求把变换过程 *tp* 添加到令牌 *token* 之中.)

if $UserClass(userj) = AuthUser \wedge tp \in TPs \wedge tp \notin DTPs \wedge token \in TOKENs \wedge tp \notin token \wedge tp \in CERTTPs$

then

$TOKENs^{*} = TOKENs \setminus \{token\} \cup \{token \cup \{tp\}\}$

(5)用户身份授权函数 *AuthUserID*(*userid*, *token*, *userj*)

(语义: *userj* 请求对用户身份 *userid* 与令牌 *token* 的关联进行认证,如果令牌中包括有可分解变换过程,则还应确保对于每个可分解变换过程而言仅有一个片在同一令牌中.)

if $UserClass(userj) = AuthUser \wedge userid \notin USERIDs \wedge token \in TOKENs \wedge \exists authuple = (userid, token) \wedge authtuple \notin AUTHTUPLEs \wedge [[tp \notin DTPs] \vee [tp \in DTPs \wedge [\forall ptpi, ptpj: PTP| ptpi, ptpj \in DTPPart\ tp \wedge ptpi \neq ptpj \cdot ptpi \in token \wedge ptpj \notin token]]]$

then

$AUTHTUPLEs^{*} = AUTHTUPLEs \cup \{authtuple\}$

(6) *DTP* 认证函数 *CertDTP*(*dtp*, *userj*)

(语义: *userj* 请求对可分解变换过程 *dtp* 进行认证,要求构成可分解变换过程片的操作元组必定已在 *dtp* 中,同时要求 *dtp* 具有对可能出现的非受控数据项进行处理的能力.)

if $UserClass(userj) = AuthUsers \wedge dtp \notin CERTDTPs \wedge [\forall ptp \in DTPPart\ dtp \cdot [[\exists tpknow \in TPKNOWs; udiknow \in UDIKNOWs \wedge userj = tpknow.1 \wedge ptp \in tpknow.2 \wedge ptp \in udiknow.1 \wedge [[udiknow.2| = 0] \vee [|udiknow.2| > 0 \wedge [\forall udicdi: UDICDI| udicdi \in udiknow.2 \cdot udicdi \in UDICDIs]] \wedge [\forall optuple: Optuple| optuple \in ptp \cdot optuple \in dtp]]] \wedge dtp \in DTPs$

then

$CERTDTPs^{*} = CERTDTPs \cup \{dtp\}$

(7) *DTP* 授权函数 *AuthDTP*(*dtp*, *TOKENsdtp*, *userj*)

(语义: *userj* 请求把可分解变换过程 *dtp* 中的各个可分解变换过程片分别添加到令牌集 *TOKENsdtp* 之中)

if $UserClass(userj) = AuthUser \wedge dtp \in DTPs \wedge dtp \in CERTDTPs \wedge TOKENs_{dtp} \subseteq TOKENs \wedge |TOKENs_{dtp}| = |DTPPart\ dtp| \wedge [\forall ptp: PTP; token: Token| ptp \in DTPPart\ dtp; token \in TOKENs_{dtp} \cdot ptp \notin token]$

then

$[\forall token \in TOKENs_{dtp} \cdot [\forall ptp \in DTPPart\ dtp \cdot [\forall tempptp: PTP|$

$tempptp \in DTPPart\ dtp \wedge tempptp \notin token \cdot token^{*} = token \cup \{ptp\} \wedge TOKENs^{*}_{dtp} = TOKENs_{dtp} \setminus \{token\} \cup \{token^{*}\}] \wedge TOKENs^{*} = TOKENs \setminus TOKENs_{dtp} \cup TOKENs^{*}_{dtp}$

(8) *TP* 执行函数 *ExecTP*(*tp*, *userj*)

(语义:*userj* 请求执行变换过程 *tp*,要求 *userj* 是 *tp* 的授权用户.)

if $UserClass(userj) = ExecUser \wedge tp \in CERTTPs \wedge tp \notin CTPs \wedge tp \notin DTPs [\exists token \in TOKENs;\ authtuple \in AUTHTUPLEs;\ useridtuple \in USERIDTUPLEs \wedge tp \in token \wedge toten = authtuple.2 \wedge userj = userituple.1 \wedge authtuple.1 = useridtuple.2] \wedge [\forall ivp : IVP \mid ivpivp \in IVPs \cdot Evaluate(ivp) = TURE]$

then

$CTPs^{*} = CTPs \cup \{dtp\}$

(9) *DTP* 执行函数 *ExecDTP*(*dtp*, *userj*)

(语义:*userj* 请求执行可分解变换过程 *dtp*,要求 *userj* 是 *dtp* 的授权用户)

if $UserClass(userj) = ExecUser \wedge dtp \in CERTDTPs \wedge dtp \notin CTPs \wedge dtp \in DTPs \wedge [\forall ptp : PTP \mid ptp \in DTPPart\ dtp \cdot [\exists token \in TOKENs;\ authtuple \in AUTHTUPLEs;\ useridtuple \in USERIDTUPLEs \wedge ptp \in token \wedge token = authtuple.2 \wedge userj = userituple.1 \wedge authtuple.1 = useridtuple.2] \wedge [\forall ivp : IVP \mid ivp \in IVPs \cdot Evaluate(ivp) = TURE]$

then

$CTPs^{*} = CTPs \cup \{dtp\}$

(10) 系统日志函数 *SysLog*(*tp*, *userj*)

(语义:*userj* 请求执行系统日志变换过程 *tp*)

if $UserClass(userj) = SysUser \wedge tp \in CTP \wedge [\forall optuple : OpTuple \mid optuple \in tp \cdot optuple \notin LOGOPTUPLEs]$

then

$LOGOPTUPLEs^{*} = LOGOPTUPLES \cup \{optuple \mid optuple \in tp\}$

4 FB-FCSM 分析

4.1 安全性分析

FB-FCSM 的安全可由结论 1 和结论 2 来保证.

结论 1:FB-FCSM 系统地强调了商务系统的完整性需求

论证:由文献[3]有关完整性目标的定义可知,维持系统数据的一致性(完整性)是各个完整性原则的终极目的. FB-FCSM 基于这个目标,以一个商务过程运行的全过程为背景,全面考察了用户身份完整性、过程授权执行完整性、过程完整性、日志完整性及职责隔离等完整性原则保持系统数据完整性的不同方式和与系统完整性目标之间的关系. 这些目标在 FB-FCSM 中被上述完整性原则之间的约束关系联接在一起,并由模型状态转移函数的形式化规范来详细描述,因此 FB-FCSM 是系统的.

结论 2:FB-FCSM 是 Clark-Wilson 完整性策略的精化.

论证:FB-FCSM 是对 Clark-Wilson 完整性安全策略的精化意味着 FB-FCSM 至少完全实现了文献[1]中所有安全策略的完整性要求:

(*C*1) 对于 Clark-Wilson 安全策略中的完整性验证过程,虽然文献[1]把它作为一个安全策略来重点突出,但 *IVP* 仅仅只可用以确信 *CDIs* 在 *IVP* 执行时的完整性状况,其作用并不影响系统的安全状态,所以本文把它认定为辅助函数.

(*C*2) 通过对 *TPs* 进行验证来保证它的良构特性,具体包括 *TPs* 结构和 *TPs* 执行语义两方面的内容. 关系(*TP*, CDI_a, CDI_b, CDI_c, …)蕴涵在 *TPs* 的结构中. 对于 *TPs* 有能力把 *CDIs* 从一个完整性状态转换到下一个完整性状态的语义认证是由认证代理所拥有的对于 *TP* 的认证知识作为认证基础.

(*C*3) 通过对可分解变换过程的认证、授权及执行行为进行控制来达到职责隔离目的.

(*C*4) Clark-Wilson 完整性安全策略中的日志功能要求是由 SysLog 规则(10)来体现.

(*C*5) FB-FCSM 使用认证代理所拥有的有关 *TP* 的 *UDI-Know* 来评估 *TP* 对 *UDI* 数据项的处理能力.

(*E*1) 在 FB-FCSM 中能够执行的 *TP* 必须是经过认证和授权的变换过程,从而间接维持了(*TP*$_i$, CDI_a, CDI_b, CDI_c, …)关系列表,同时认为对任何 *CDI* 的操作都是由 *TP* 来产生的.

(*E*2) 关系(*UserID*, *TP*$_i$, CDI_a, CDI_b, CDI_c, …)已被分解为两部分,即由用户身份元组和变换过程授权元组两者来共同表示.

(*E*3) 由用户身份认证过程来保证每一个试图去执行 *TP* 的用户的身份都是经过认证的.

(*E*4) 对于原 Clark-Wilson 安全策略中的不同用户类型要求,已通过对各策略规则请求者身份的限制达到了目的.

因此 FB-FCSM 不但完全实现了 Clark-Wilson 完整性策略系统,同时又对这些策略要素进行了显式定义和约束分析.

4.2 FB-FCSM 特性分析

框架特性是 FB-FCSM 的核心机制,这种机制具有很强的兼容性和扩展性.

FB-FCSM 的兼容性表现在可以方便地对其中的各种组件属性进行改造,从而使得该模型可以适合于各种不同应用背景. 例如:文中只是简单认为令牌 *Token* 为一个 *TPs* 标识符集合,但事实上,如果在令牌 *Token* 中依据具体应用语义加入有关 *TPs* 之间关系(比如层次关系)的描述特性,则可以很容易使 FB-FCSM 具有对于复杂高层应用的授权进行控制的特性.

FB-FCSM 的扩展性主要表现在可以方便地在控制机制中增加新的控制层,从而增强该策略模型的控制能力. 例如,通过在用户身份和令牌之间增加角色层,就实现了 Matunda NYANCHAMA 在文献[7]中所提出的 Clark-Wilson 完整性策略和 RBAC 存取控制机制之间的结合问题,从而可以大大改善商务系统安全授权的可管理性.

5 结论

开发适应商务系统多样性的系统的、适应性强的安全策略具有迫切的实际需求. 本文围绕维持系统数据一致性的系统完整性目标,用框架机制有效地把用户完整性、过程完整性、过程执行完整性、职责隔离完整性、日志完整性等完整性机制集成到基于框架的形式化商务安全策略模型(BF-FCSM)之中,使该形式化模型系统地强调了商务完整性需求的各个方面,具有良好的兼容性和扩展性,它是 Clark-Wilson 完整性安全策略的精化.

参考文献:

[1] D D Clark, D R Wilson. A comparison of commercial and military computer security policies[A]. IEEE Symposium on Security and Privacy [C]. Oakland, CA: IEEE press, 1987. 184 - 194.

[2] S Fischer-Hübner. IT-Security and Privacy-Design and Use of Privacy-Enhancing Security Mechanisms[M]. NY: Springe, 2001. 201 - 257.

[3] T Mayfield, J E Roskos, S R Welke, J M Boone. Integrity in Automated Information Systems[R]. U. S. National Computer Security Center, 1991. 79 - 91.

[4] U S DoD. A Guide to Understanding Security Modeling in Trusted Systems[R]. NCSC-TG-010, U. S. National Computer Security Center, Oct. 1992.

[5] H Anthony Z. Styles for security properties and modern user interfaces [A]. A Ali E, R Peter, S Steve (Eds.): Formal Aspects of Security, First International Conference[C]. London: Publishing Springer, 2003. 152 - 166.

[6] R Ramakrishnan, J Gehrke. Database Management Systems (2th ed.) [M]. New York: McGraw-Hill, 2000. 51 - 83.

[7] Matunda Nyanchama. Commercial Integrity, Roles and Object Orientation[D]. London, Ontario, Canada: Department of Computer Science, Faculty of Graduate Studies, University of Western Ontario, 1994.

作者简介:

温红子 男,1969 年生于甘肃静宁,博士,主要研究领域为信息安全理论与技术、企业信息系统集成. E-mail: wenhongzi@vip.sina.com.

周永彬 男,1973 生于山东阳信,博士,主要研究领域为应用密码学、网络与信息安全理论与技术.

卿斯汉 男,1939 生于湖南隆回,研究员,教授,博士生导师,主要研究领域为信息安全理论与技术.