

使用控制访问模型的研究^{*}

彭凌西^{1,2}, 杨 频², 彭银桥¹, 孙飞显², 曾金全², 刘才铭²

(1. 广东海洋大学 信息学院, 广东 湛江 524025; 2. 四川大学 计算机学院, 成都 610065)

摘 要: 分析了现今高度动态和分布式环境下, 传统的访问控制模型已不能满足信息访问的需要, 并由此引出了下一代访问控制模型——使用控制模型(usage control, UCON)。将 UCON 与传统访问控制模型进行了比较, 分析了 UCON 的定义及组成成分, 给出了 UCON 的 16 种基本核心模型形式化描述及应用实例, 分析了 UCON 的具体实现框架, 最后指出了目前研究的难点和需要解决的关键问题。

关键词: 访问控制模型; 使用控制; 网络安全

中图分类号: TP309 文献标志码: A 文章编号: 1001-3695(2007) 09-0121-03

Survey of usage access control model

PENG Ling-xi^{1,2}, YANG Pin², PENG Yin-qiao¹, SUN Fei-xian², ZENG Jin-quan², LIU Cai-ming²

(1. School of Information, Guangdong Ocean University, Zhanjiang Guangdong 524025, China; 2. School of Computer, Sichuan University, Chengdu 610065, China;)

Abstract: In today 's highly dynamic and distributed environment, the traditional access control models can not meet the need of the information access. The next generation access control model usage control (UCON) model, was presented. The UCON model was first compared with the traditional access control models, and then the components and definition of the model were discussed. The 16 types of basic core models of UCON was formalized and the application instances were given. The implementation frame of UCON was also presented. Finally, the difficulties of research and the existing problems that must be solved were pointed out.

Key words: access control model; usage control; network security

访问控制是对信息系统资源进行保护的重要措施, 决定了谁能够访问系统、能访问系统的何种资源以及如何使用这些资源。传统访问的控制模型包括自由访问控制(discretionary access control, DAC)、强制访问控制(mandatory access control, MAC)、以及基于角色的访问控制(role-based access control, RBAC)。这些模型建立在访问控制矩阵的基础上, 其授权过程是基于访问的主体属性和被访问的客体属性以及访问的权限^[1~3]。图 1 显示了这种基于属性访问控制授权过程。

尽管这些传统的基于属性的访问控制模型能够解决许多实际问题, 但却不能满足现今数字信息访问的需要。例如, 有时客体需要某种动作才可以进行访问; 系统需要根据当前环境或者系统状态进行访问决策; 在传统访问控制模型中, 授权发生在访问前, 然而在整个访问期间需要根据访问情况变化, 对相对长期的访问或者对访问权限进行立即回收; 在传统访问控制中, 主体客体属性只可以通过系统管理员才可以修改, 然而在许多应用中属性可以变化, 具体可以分为访问前、访问期间、或者是访问后。

鉴于传统访问控制模型存在的上述问题, Sandhu 教授领

导的 LIST 实验室于 2002 年展开了下一代访问控制模型——使用控制访问模型(UCON) 的研究, 并迅速成为访问控制技术的研究热点^[4~11]。本文就 UCON 模型定义和组成、核心模型的形式化描述、应用实例以及实现体系结构进行了全面的研究。

1 模型组成

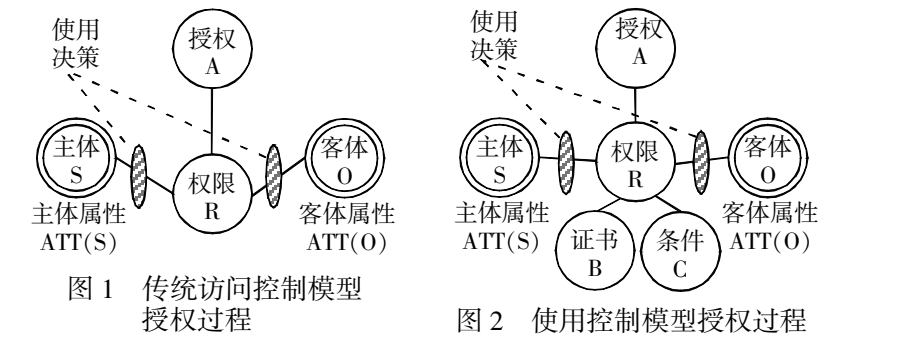
UCON 由主体 S(subject)、主体属性 ATT(S) (subject attributes)、客体 O(object)、客体属性 ATT(O) (object attributes)、权限 R(rights)、授权 A(authorizations)、证书 B(obligations)、条件 C(conditions)^[5] 八个部分组成, 如图 2 所示。与图 1 中的传统访问控制模型不同, 使用控制模型增加了主体属性、客体属性、证书和条件, 这些新增加的成分用于解决传统访问模型中存在的问题。

主体 S 是具有属性的实体, 能够在客体上执行一定的权限; 主体属性 ATT(S) 是主体进行授权控制的主体的属性。客体 O 是主体能够控制权限, 并能够访问和控制的实体, 客体具有本身的属性或者跟权限相关的属性; 客体属性 ATT(O) 可用于授权过程, 并用于访问决策。权限 R 是主体能够对客体进

收稿日期: 2006-07-30; 修返日期: 2006-09-25 基金项目: 国家自然科学基金资助项目(60371046); 广东海洋大学自然科学基金资助项目(0512135, 0412128)

作者简介: 彭凌西(1978-), 男, 湖南岳阳人, 讲师, 博士研究生, 主要研究方向为人工免疫、网络安全(manplx@ 163. com); 杨频(1967-), 男, 四川岳池人, 副教授, 博士研究生, 主要研究方向为网络安全; 彭银桥(1969-), 男, 湖南祁东人, 讲师, 主要研究方向为薄膜传感技术及信息安全研究; 孙飞显(1970-), 男, 河南宝丰人, 博士研究生, 主要研究方向为人工免疫、网络安全; 曾金全(1973-), 男, 四川广安人, 讲师, 博士研究生, 主要研究方向为人工免疫、网络安全; 刘才铭(1977-), 男, 四川广安人, 博士研究生, 主要研究方向为人工免疫、网络安全。

行控制和执行的特权,由主体对客体进行访问的功能集组成,权限之间可具有层次关系。一般来说,权限包括对客体的访问、权限授予和权限管理。权限按功能可分为很多种,最常见的是读和写,另外包括创建和删除。授权 A 是一个用于访问决策,返回主体能否在客体上执行需要权限的断言。授权根据客体属性、主体属性、请求的权限以及授权规则来进行访问决策。授权可以是预先授权或者是进行授权。一般说来,传统访问控制模型如 MAC、DAC 和 RBAC 使用了预先授权的某种形式,证书 B 是在访问前或者访问过程中需要强制验证的断言。证书可以分为预先证书 preB 或者是进行证书 onB。预先证书是根据历史功能来检查某些活动是不是已经完成,并返回“真”或者“假”的断言。条件 C 是对当前环境或者系统状态进行检查,并返回“真”或者“假”的断言。与授权和证书不同,条件不能变化,因为条件不能被单个主体所直接控制,条件的设置也不因主体或者客体属性更新而改变。



2 模型定义和分析

基于上述三个决定因素,授权、证书和条件以及主体和客体属性在访问期间的延续性和可变性,假设主体对目标客体有访问请求,对访问权限的访问决策发生在请求前或者在请求期间。主体/客体属性可变性允许在使用期间作为访问的结果对主体或者客体的属性进行更新,如果没有更新,在访问决策过程中没有属性更新要求表示为“0”,使用前(pre)、使用期间(ongoing)、使用后(post)分别标记为“1、2、3”,只考虑纯粹的有授权、证书和条件以及访问前和访问期间的决策。基于以上标准,得到表 1 使用控制的 16 基本模型^[6]。表 1 显示了基于三个决定因素的各种可能模型,现实中不可能的情况标记为“N”,否则为“Y”。如果决策因素是“pre”,属性更新可能发生在权限执行前或执行后,但不可能在执行期间进行更新;进行更新“ongoing”可以只影响将来的请求决策,因此更新可以在访问后进行。然而,如果决策因素是“进行”,属性更新可能发生在访问前后或者访问期间。对于条件模型,因为它只与环境或者系统状态相关,不能对主体和客体属性进行更新。

表 1 使用控制的 16 种基本核心模型

决策因素	0	1	2	3
preA	Y	Y	N	Y
onA	Y	Y	Y	Y
preB	Y	Y	N	Y
onB	Y	Y	Y	Y
preC	Y	N	N	N
onC	Y	N	N	N

2.1 预先授权模型 UCON_{preA}

在 UCON_{preA} 模型中,决策过程发生在授权允许访问前,用 UCON_{pre0} 表示不对主体客体属性进行更新,形式化描述如下:

定义 1 UCON_{preA0} 模型具有以下成分:

- a) S、O、R、ATT(S)、ATT(O)、preA 分别代表主体、客体、权限、主体属性、客体属性、预先授权;
- b) $allowed(s, o, r) = preA(ATT(s), ATT(o), r)$ 。

实例 1 强制访问控制通过 UCON_{preA0} 表达如下:

- (a) L 是用 $\geq$ 域表示的安全标志分级。许可: $S \rightarrow L$, 分级: $O \rightarrow L$ 。
- (b) $ATT(S) = \{ 许可 \}$, $ATT(O) = \{ 分级 \}$ 。
- (c) $allowed(s, o, read) = > 许可(s) \geq 分级(o)$ 。
- (d) $allowed(s, o, write) = > 许可(s) \leq 分级(o)$ 。

实例 2 自由访问控制通过 UCON_{preA0} 表达如下:

- (a) N 表示验证实体集。
- (b) $d: S \rightarrow N$, $ACL: O \rightarrow 2^{N \times R}$ (n 表示授权对 o 进行 r 操作)。
- (c) $ATT(S) = \{ id \}$, $ATT(O) = \{ ACL \}$ 。
- (d) $allowed(s, o, r) = > (id(s), r) \in ACL(o)$ 。

定义 2 UCON_{preA1} 模型除了下面增加授权访问前更新过程外, UCON_{preA3} 除了增加访问后属性更新外,它们与 UCON_{preA0} 模型等价。

- (a) UCON_{preA1} 增加了 $preUpdate(ATT(s))$ 和 $preUpdate(ATT(o))$ 。
- (b) UCON_{preA3} 增加了 $postUpdate(ATT(s))$ 和 $postUpdate(ATT(o))$ 。

2.2 访问时进行授权模型 UCON_{onA}

UCON_{onA0} 定义为访问授权时没有对主体/客体属性进行更新,形式化描述如下:

定义 3 进行授权模型具有以下成分:

- (a) S、O、R、ATT(S)、ATT(O) 以及进行授权 onA;
- (b) $allowed(s, o, r) = > true$;
- (c) $stopped(s, o, r) < = - onA(ATT(s), ATT(o), r)$ 。

如果没有预先授权,请求访问也是允许的。然而授权在对请求权限的使用过程中是激活的, onA 断言在访问过程对维持访问不断进行检查。从技术实现上讲,这些检查是基于时间或者是事件的。在有些情况中,某些属性被改变,条件不再得到满足则执行 stopped 过程。Stopped(s, o, r) 表明主体 s 对客体 o 的访问权限在访问过程中终止。

UCON_{onA1}、UCON_{onA2} 和 UCON_{onA3} 分别增加了预先更新 $preupdate(ATT(s))$ 、 $preupdate(ATT(o))$ 、访问时更新 $onUpdate(ATT(s))$ 、 $onUpdate(ATT(o))$ 以及访问后更新 $postUpdate(ATT(s))$ 和 $postUpdate(ATT(o))$ 过程。

2.3 预先访问证书模型 UCON_{preB}

UCON_{preB} 包括了在允许访问前必须完成相关的证书,称为 preB 断言。例如,需要用户在访问一个公司的白皮书之前需要提供名字和 e-mail 地址,或者是用户点击注册协议上的确定按钮来进行 Web 访问入口。预先证书不是在用户需要进行访问的对象(如白皮书、Web 入口)上进行的,而是需要另外的主体来完成。模型的形式化描述如下:

定义 4 UCON_{preB0} 模型具有以下成分:

- (a) S、O、R、ATT(S)、ATT(O) 与前面的模型相同。
- (b) OBS、OBO、OB 分别表示主体证书、客体证书、证书动作。
- (c) preB、preOBL 分别表示预先授权断言和预先授权成分。
- (d) $preOBLA OBS \times OBO \times OB$, 表示预先完成的证书组成。
- (e) $preFullfilled: OBS \times OBO \times OB \rightarrow \{ true, false \}$ 。
- (f) $getPreOBL: S \times O \times R \rightarrow 2^{preOBL}$, 表示用于访问请求使用选择的预先证书的函数。
- (g) $preB(s, o, r) = \bigwedge_{(obs_i, obo_i, ob_i) \in getPreOBL(s, o, r)} preFullfilled(obs_i, obo_i, ob_i)$ 。
- 如果 $getPreOBL(s, o, r) =$, $preB(s, o, r)$ 必须为 true, 其意义是如访问需要的证书为空,那么就可进行访问。

(h) $\text{allowed}(s, o, r) = > \text{preB}(s, o, r)$ 。

getPreOBL 功能表示对主体 s 和客体 o 的授权进行预先证书的功能。preFullfilled 断言表示每个需要的证书为真; pre-Fullfilled 断言表明每个请求的证书必须为真。 $\text{UCON}_{\text{preB1}}$ 和 $\text{UCON}_{\text{preB3}}$ 分别增加了 $\text{preUpdate}(\text{ATT}(s))$ 、 $\text{preUpdate}(\text{ATT}(o))$ 和 $\text{postUpdate}(\text{ATT}(s))$ 、 $\text{postUpdate}(\text{ATT}(o))$ 过程。

2.4 进行访问证书模型 UCON_{onB}

进行证书访问模型需要完成证书才可以访问, 包括间断性完成或者持续性完成。例如用户每隔至少 30 min, 或者每访问 20 个 Web 页面就需要点击广告, 形式化描述如下:

定义 5 进行访问证书 $\text{UCON}_{\text{onB0}}$ 模型定义如下:

(a) $S, O, R, \text{ATT}(S), \text{ATT}(O), \text{OBS}, \text{OBO}, \text{OB}$ 与 $\text{UCON}_{\text{preB}}$ 相同;

(b) $\text{getONOBL}: S \times O \times R \rightarrow 2^{\text{onOBL}}$ 表示对请求访问选择进行证书;

(c) T 表示事件或者时间集;

(d) onB 和 onOBL 分别表示进行证书断言和进行证书成分;

(e) $\text{onOBLA} \text{ OBS} \times \text{OBO} \times \text{OB} \times T$;

(f) $\text{getOnOBL}: S \times O \times R \rightarrow 2^{\text{onOBL}}$ 用于选择请求使用的进行证书;

(g) $\text{onFullfilled}: \text{OBS} \times \text{OBO} \times \text{OB} \times T \rightarrow \{ \text{true}, \text{false} \}$;

(h) $\text{onB}(s, o, r) = \bigwedge_{\text{obs}_i, \text{obo}_i, \text{ob}_i, \text{t}} \text{tigetOnOBL}(s, o, r) \text{ onFullfilled}(\text{obs}_i, \text{obo}_i, \text{ob}_i, \text{t}_i)$;

(i) 当 $\text{getONOBL}(s, o, r)$ 为空时, $\text{onB}(s, o, r) = \text{true}$;

(j) $\text{allowed}(s, o, r) = > \text{true}$;

(k) $\text{stopped}(s, o, r) < = \neg \text{onB}(s, o, r)$ 。

$\text{UCON}_{\text{onB1}}$ 、 $\text{UCON}_{\text{onB2}}$ 和 $\text{UCON}_{\text{onB3}}$ 分别增加了相应的主体和客体属性更新过程。

2.5 预先访问条件模型 $\text{UCON}_{\text{preC}}$

条件由环境和系统状态决定, 与主体/客体属性无关。与其他模型相比, 条件模型没有属性更新过程, 在访问前进行检查的模型是 $\text{UCON}_{\text{preC0}}$, 如对客户端 IP 地址的检查。形式化描述如下:

定义 6 $\text{UCON}_{\text{preC0}}$ 模型具有如下成分:

(a) $S, O, R, \text{ATT}(S), \text{ATT}(O)$ 与 $\text{UCON}_{\text{preA}}$ 相同;

(b) preCON 是预先条件成分的集合;

(c) $\text{preConChecked}: \text{preCON} \rightarrow \{ \text{true}, \text{false} \}$;

(d) $\text{getPreCON}: S \times O \times R \rightarrow 2^{\text{preCON}}$;

(e) $\text{pre}(s, o, r) = \bigwedge_{\text{preCon}_i \in \text{getPreCON}(s, o, r)} \text{preConChecked}(\text{preCon}_i)$;

(f) $\text{allowed}(s, o, r) = > \text{preC}(s, o, r)$ 。

2.6 进行访问条件模型 UCON_{onC}

定义 7 进行访问条件模型 $\text{UCON}_{\text{onC0}}$ 定义如下:

(a) $S, O, R, \text{ATT}(S), \text{ATT}(O)$ 与 $\text{UCON}_{\text{preA}}$ 相同;

(b) onCON(进行条件成分集);

(c) $\text{getOnCON}: S \times O \times R \rightarrow 2^{\text{onCON}}$;

(d) $\text{onCONchecked}: \text{onCON} \rightarrow \{ \text{true}, \text{false} \}$;

(e) $\text{onC}(s, o, r) = \bigwedge_{\text{onCon}_i \in \text{getOnCON}(s, o, r)} \text{onConChecked}(\text{onCon}_i)$;

(f) $\text{allowed}(s, o, r) = > \text{true}$;

(g) $\text{stopped}(s, o, r) < = \neg \text{onC}(s, o, r)$ 。

在访问过程中, 通过 UCON_{onC} 模型的 onC 断言对访问进行检查并决定是否可以继续访问。例如系统状态切换到“紧急模式”, 则访问终止; 另外系统超过了一定的访问负载数量也可以终止某些用户的访问。

3 实现体系结构

从实现体系结构来看, UCON 最关键的是参考监视器。参考监视器与访问策略和对数字资源的访问控制相关, ISO 提出了参考监视器和可信计算机的访问控制框架的标准。根据该

标准, 参考监视器由两个部分组成, 即访问控制实施部件(AEF) 和访问控制决策部件(ADF) 。每个请求由 AEF 来接收, 然后由 ADF 进行请求访问的决策。UCON 的参考监视器与 ISO 传统访问监视器相似, 由使用决策部件(UDF) 和使用执行部件(UEF) 组成。基于参考监视器的位置, 可以分为服务器端参考监视器(SRM)、客户端监视器(CRM) 以及 SRM 与 CRM 混合结构^[5]。

4 应用实例

UCON 模型包含并超越传统访问控制模型 MAC、DAC 以及 RBAC。其中 RBAC 可以通过配置实现 DAC 和 MAC^[21]。因此, 本文只研究如何通过 UCON 实现 RBAC。在 $\text{UCON}_{\text{preA0}}$ 中, 用户角色分配可以看成是客体属性, 权限角色分配可以看成是对象和权限属性。例 3 显示了在 UCON_{ABC} 模型如何实现包含层次关系的 RBAC_1 , 只有激活的角色可用于授权决策。因此, 如果只有存在激活角色 $\text{actRole}(s)$ 决定授权角色 $\text{PRole}(o, s)$, 访问请求才允许。例 4 是一个在 DRM 系统访问使用期间进行预先更新($\text{UCON}_{\text{preA1}}$) 信用卡例子。

实例 3 $\text{UCON}_{\text{preA0}}$ 实现 RBAC_1

(a) $P = \{ (o, r) \}$, ROLE 表示用 $\geq$ 符号表示的有序角色集;

(b) $\text{ActRole}: S \rightarrow 2^{\text{ROLE}}$;

(c) $\text{Prole}: P \rightarrow 2^{\text{ROLE}}$;

(d) $\text{ATT}(S) = \{ \text{actRole} \}$;

(e) $\text{ATT}(O) = \{ \text{PRole} \}$;

(f) $\text{allowed}(s, o, r) = > \forall \text{role} \in \text{actRole}(s), \forall \text{role} \in \text{Prole}(o, r) \geq \text{role}$ 。

实例 4 DRM 系统使用期间付款的 $\text{UCON}_{\text{preA1}}$ 实例:

(a) M 代表账户集;

(b) $\text{Credit}: S \rightarrow M$;

(c) $\text{Value}: O \times R \rightarrow M$;

(d) $\text{ATT}(s) : \{ \text{credit} \}$;

(e) $\text{ATT}(o, r) : \{ \text{value} \}$;

(f) $\text{Allowed}(s, o, r) = > \text{credit}(s) \geq \text{value}(o, r)$;

(g) $\text{PreUpdate}(\text{credit}(s)) : \text{credit}(s) - \text{value}(o, r)$ 。

在访问期间, 用户的账户根据访问的时间进行减小。

5 结束语

理论研究表明, 使用控制访问模型包含并超越了传统的访问控制模型, 能满足现今数字资源访问的需要, 必将为下一代访问控制技术发展奠定坚实的基础。然而, 目前对使用控制模型的研究仅局限于理论模型, 模型描述有待进一步丰富和完善, 并需要一个属性和管理策略的管理模型。另外在分布式和多用户环境下, UCON 系统涉及对属性进行的并发更新问题也有待进一步研究。

参考文献:

[1] ANDHU R, COYNE E, FEINSTEIN H, *et al.* Role-based access control models[J] . IEEE Computer, 1996, 29(2) : 38-47.

[2] OSBORN S, SANDHU R, MUNAWER Q. Configuring role-based access control to enforce mandatory and discretionary access control policies[J] . ACM Transactions on Information and Systems Security, 2000, 3(2) : 85-106.

[3] 赵宝献, 秦小麟. 数据库访问控制研究综述[J] . 计算机科学, 2005, 32(1) : 88-91. (下转第 126 页)

生器 PBG, 其作用是产生为用户和攻击者可见的内部指纹。指纹的嵌入是软件制造商通过下列步骤实现:

(a) 准备阶段。给定软件 $S \in C_S$, 其初始拷贝为 S_M , 并定义一个安全参数来提供分割 S 的块的大小。软件 S 被分割为 $2un + 1$ 块, $S = (S_1, \cdots, S_{2un+1})$, 参数 u 是依据软件的长度和安全要求来选取的。

(b) 产生签名。计算 $h = H(S_M, id_p, id_v, date)$, 同时产生签名 $SG_p = D_{k_p}(h)$ 。

(c) 构造秘密共享。构造 u 对 (t, n) 门限方案(u 的大小是由安全要求决定)。每对中第一个秘密共享是 h , 第二个是数字签名 SG_p , 最终结果有 $2un$ 个共享(块)。令所有共享组成的序列为 $\alpha = (\alpha_1, \cdots, \alpha_{2un})$ 。

(d) 屏蔽共享。版权持有者选取一个秘密 β 使用伪随机序列发生器产生一个足够长的位串, 表示为 $\gamma = (\gamma_1, \cdots, \gamma_{2un}) = \alpha \text{ PBG}(\beta)$ 。

(e) 嵌入内部指纹。对每一个软件分块 $S_i \quad i = 1, \cdots, 2un$, 其变量 S_i^* 的标志记为 $id_{S_i^*} = \gamma_i$ 。

(f) 嵌入外部指纹。 $h^* = H(S_1^*, \cdots, S_{2un}^*, id_p, id_v, date)$ 最后一个块被选取为标志, $id_{S_{2un+1}^* = SG_p(h^*)}$ 。注意软件的每一个分块 S_i 必须足够长, 以符合嵌入量的要求。

b) 用户检测标志
任何用户可以通过检查最后一个块是否与签名 $SG_p(h^*)$ 相匹配来检测指纹。例如, 检查从最开始的 $2un$ 个块中获取的哈希值是否与从嵌入 S_{2un+1}^* 中签名中提取的哈希值一致。

c) 版权持有者检测标志
版权持有者首先通过计算 $\text{PBG}(\beta)$, 去掉屏蔽并提取 $2n$ 个共享 $\alpha = \gamma \text{ PBG}(\beta)$, 并从每一个 $2n$ 块中检测签名。只要在每 n 个块中存在 t 个未受干扰的块就足够通过检测。

其安全性显然与数字签名方案、哈希函数以及 PBG 相关。软件代码指纹提供一个外部公开可见的指纹, 而内部指纹是屏蔽的。事实上, 串 γ 是从一个真的随机强加密 PBG 中产生的, 是多项式和不可辨别的, 一个攻击者甚至无法意识到内部指纹的存在。并且, 版权持有者无法通过其私有秘密 β 来获得一个

期望的 α , 即版权持有者不可能获取其他软件版权的相关信息, 原因是在 $\alpha = \gamma \text{ PBG}(\beta) = \gamma' \text{ PBG}(\beta)$ 中由 α 和 γ 来获得 β 几乎是不可能实现的。

大部分计算机软件是通过目标代码销售和分发的, 指纹可以在源代码编译器中被嵌入, 但是往往有些编译器会作代码优化来删除代码冗余。而软件代码指纹却向程序代码中添加了冗余代码, 因此软件代码指纹很可能被某些编译器移除。在这种情况下, 提供软件版权依据的可以是嵌入了代码指纹的软件源代码。目标代码则无法提供合适的版权依据。

4 结束语

本文提出的基于代码的软件指纹技术理论和实现方案是应对计算机网络环境和软件不断发展的服务模式而提出的, 对软件的合法版权提出了新的保护手段。随着信息隐藏技术的发展, 使得软件版权信息从简单的静态隐藏向动态技术发展, 类似于传统软件水印技术, 软件指纹从最初的静态代码指纹发展到针对软件行为而设计的动态指纹。软件代码指纹针对软件源代码或者目标代码嵌入指纹信息实现不同软件拷贝的静态标志, 其实现比较简单, 但安全性相对较低; 软件行为指纹是对软件运行过程和状态演变的动态标志, 其理论和实现比较复杂, 但是可以达到较高的安全性, 这方面将会是今后课题研究的重点。

参考文献:

[1] UCSMITH D. Tamper resistant software: an implementation[C] // ANDERSON R. Proc of the 1st International Workshop on Information Hinding. Berlin: Springer-Verlag, 1996: 317-334.

[2] BONEH D, SHAW J. Collusion-secure fingerprinting for digital data [C] // COPPERSMITH D. Proc of the Advances in Cryptology. Berlin: Springer-Verlag, 2000: 452-465.

[3] PIEPRZYK J. Fingerprints for copyright software protection [C] // Proc of the 2nd International Workshop on Information Security. London: Springer-Verlag, 2000: 178-190.

[4] 张立和, 杨义先, 钮心忻, 等. 软件水印综述 [J]. 软件学报, 2003, 14(2): 268-277.

Technology. Berlin: IEEE Computer Society Press, 2005: 512-517.

[8] BERTINO E, KHAN L R, SANDHU R, *et al.* Secure knowledge management: confidentiality, trust, and privacy[J]. IEEE Transactions on Systems, Man and Cybernetics: Part A, 2006, 36(3): 429-438.

[9] ZHANG Xin-wen, CHEN Song-qing, SANDHU R. Enhancing data authenticity and integrity in P2P systems[J]. IEEE Internet Computing, 2005, 9(6): 42-49.

[10] BERTINO E, SANDHU R. Database security-concepts, approaches, and challenges [J]. IEEE Transactions on Dependable Secure Computing, 2005, 2(1): 2-19.

[11] ZHANG Xin-wen, SANDHU R, PARISI-PRESICCE F. Safety analysis of usage control authorization models[C] // Proc of ACM Symposium on Information, Computer, and Communication Security. Taipei: ACM Press, 2006: 243-254.

(上接 123 页)

[4] PARK J, SANDHU R. Towards usage control models: beyond traditional access control[C] // Proc of the 7th ACM Symposium on Access Control Models and Technologies. Monterey, California: ACM Press, 2002: 57-64.

[5] PARK J, SANDHU R. The UCON_ABC usage control model[J]. ACM Transactions on Information and System Security, 2004, 7(1): 128-174.

[6] ZHANG Xin-wen, PARISI-PRESICCE F, SANDHU R, *et al.* Formal model and policy specification of usage control [J]. ACM Transactions on Information and System Security, 2005, 8(4): 351-387.

[7] UNLU V, HESS T. The access-usage-control-matrix: a heuristic tool for omplementing a selected level of technical content protection [C] // Proc of the 7th IEEE International Conference on E-Commerce