基于动态许可证的信任版权安全认证协议*

马兆丰+, 冯博琴, 宋擒豹, 王浩鸣

(西安交通大学 计算机科学与技术系,陕西 西安 710049)

Secure Authentication Protocol for Trusted Copyright Management Based on Dynamic License

MA Zhao-Feng+, FENG Bo-Qin, SONG Qin-Bao, WANG Hao-Ming

(Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China)

+ Corresponding author: Phn: +86-29-2675595, E-mail: supermzf@mail.china.com, http://www.xjtu.edu.cn

Received 2003-01-07; Accepted 2003-09-05

Ma ZF, Feng BQ, Song QB, Wang HM. Secure authentication protocol for trusted copyright management based on dynamic license. *Journal of Software*, 2004,15(1):131~140.
http://www.jos.org.cn/1000-9825/15/131.htm

Abstract: Trusted software copyright protection is one of the most important issues in digital rights management. However, most of the current solutions could not meet the demand of End User License Agreement (EULA) in security and efficiency. In this paper, a new and secure authentication protocol for trusted copyright protection based on dynamic license is proposed to solve the above problem. A third part Certificate Authority (CA) is adopted for an atomic authorization and a forced revocation of software license dynamically according to the software and hardware identity and their usage status. Thus under the control of the dynamic license, the copyright is protected safely and the software entity can be transferred freely without copyright damage and resource leakage. Considering the integrity and security of the dynamic license, symmetric and public key cryptography algorithms are used for data encryption and digital signature respectively, while random verification of coding signature is adopted to resist possible attack and runtime crack. Analysis manifests that the proposed protocol is feasible and secure with a high integrity. It can meet the demand of EULA and provide a new and reliable approach for software copyright management.

Key words: trusted copyright management; dynamic license; data encryption; digital signature

摘 要: 信任软件版权管理是数字版权体系中专门针对软件版权控制的一项重要研究内容.针对当前版权许可软件措施在安全性、有效性方面无法完全满足国际通用的最终用户许可协议(EULA)要求的问题,基于第三方

* Supported by the National High-Tech Research and Development Plan of China under Grant Nos.863-306-QN2000-5, 2003AA1Z2610 (国家高技术研究发展计划(863))

作者简介: **马兆丰**(1974－),男,甘肃镇原人,博士生,主要研究领域为数论和现代密码学,信息安全理论与技术,信任版权管理,机器学习,自适应主动信息检索;**冯博琴**(1942－),男,教授,博士生导师,主要研究领域为计算机网络与信息安全,信息检索,知识工程;**宋擒豹**(1966－),男,博士,副教授,主要研究领域为数据挖掘,知识工程,计算机网络安全;**王浩鸣**(1968－),男,博士,副教授,主要研究领域为数据库应用,计算机网络安全.

可信中心提出了一种动态许可证支持的信任版权动态分布式安全认证协议.该协议将软件实体、软件运行环境以及版权状态联合考虑,通过“特征关联,原子授权,强制收权”的机制有效解决了“软件版权的安全保护,软件资源的任意迁移和软件内容的完整保持”这 3 方面的问题.协议交互中通过加密和数字签名保证分布式环境下数据的安全性和完整性,而实现上以代码随机验证签名实现反跟踪.分析证明,所提出的方案在可行性、安全性以及完备性方面均达到了 ELUA 协议的要求.与已有的方案相比,协议认证机制安全可靠,成本低且易于实施,为软件版权保护提供了一种全新的视野.

关键词: 信任版权管理;动态许可证;数据加密;数字签名

中图法分类号: TP309 **文献标识码**: A

数字版权保护主要涉及多媒体数字商品和应用软件两类形式的数字实体[1,2].长期以来,以数字水印、安全容器、安全协议等方式进行多媒体内容版权保护得到了广泛的研讨:如文献[3~7]通过数字水印实现多媒体内容版权控制;文献[8]针对多媒体数字版权保护的内容、范围、安全性、易用性和公平性等进行了比较系统的研究;而文献[9]探讨了网络商务版权保护问题,其中基于密码技术特别是公钥密码技术的多媒体内容版权保护安全协议表现出了良好的性能[4,5,7,9,10].然而遗憾的是,针对应用软件的数字商品内容版权保护却没有引起足够的重视,致使商用正版软件的非法复制传播等盗版行为十分猖獗,如何有效保护数字商品的知识产权是事关软件供应商和消费者权益的重要议题.在版权意义下,软件可分为商用软件、共享软件、免费软件和公用软件 4 种,其中公用软件是版权已被放弃的一种软件,其他 3 类软件均受软件版权保护,而商业软件版权的保护强度和要求是最高的[11,12].目前,最终用户许可协议 EULA(end user license agreement)是商用软件的国际通行许可协议,该协议内容规定:

(1) 软件的一份副本在授权许可下在单一一台硬件实体上安装、使用.

(2) 运行该软件所在的计算机必须获得一份许可证.同一份“软件产品”许可证不得在不同的计算机共同或同时使用等.

(3) 不允许反向工程(reverse engineering)、反向编译(decompilation)、反汇编(disassembly)等.

遗憾的是,EULA 仅仅是一个文字上的约定而已,它并没有提供有效的版权保护机制和控制手段,从而根本不能保证软件不被非法复制与扩散,无法控制用户的任意安装与使用.为了达到版权保护的目的,人们探索了许多方案和技术,按保护手段可分为硬件保护方式和纯软件保护方式两类.其中,基于硬件载体的保护方式主要有基于软盘的加密方式、基于逻辑门集成电路的加密方式(俗称加密狗)以及基于安全光盘的保护方式等.而以软件程序为载体的保护措施主要有软件产品密钥保护方式、序列号保护方式以及许可证方式等.这些保护措施由于其原理和技术手段的不同所表现出来的性能有很大的差异.

基于物理硬件载体的保护措施在一定程度上能保证版权的有效控制,如不允许复制软件内容(如 SafeDisk[13],Rainbow [14])或者复制内容但无法使用(如朗讯、微软等对大型软件通过许可协议盘来限制安装.瑞星、江民产品将软件内容与许可协议关联防止内容复制等);文献[13,14]的控制手段必须依赖于特定的硬件实体(如光盘驱动器)读取数据;软件内容与许可协议关联的软盘方式无法满足软件容量比较大的软件实体;而基于逻辑门电路的方式对于价值不是很高的软件来讲开销相对较大.这些措施虽然能够防止软件内容的非法复制,但却不能保证软件使用权的惟一性.

在以软件程序为载体的保护措施中,不论是基于产品密钥、序列号或是许可证方式,其本质上都是一种静态的预先授权手段,而用户一旦获得授权,便可任意复制扩散安装;基于软件运行环境相关的版权控制措施,一些有效的措施能够保证软件使用权的原子性,但却无法实现软件运行环境的迁移[15,16],这极大地限制了软件使用的方便性.

虽然上述方法各具特色,在一定程度上能在版权保护的某些方面起到作用,但是,上述各种方法均无法完全达到 EULA 协议的基本要求:(1) 不能有效解决软件内容的原子性使用;(2) 无法实现软件版权意义下的软件资源迁移;(3) 纯软件实现不具有安全性保证等缺陷.事实上,目前已有的诸多保护措施仅仅从软件实体本身进行了考虑,这样,软件内容一旦发布,软件供应商便失去了对软件版权的控制,从而根本无法保证版权不受侵犯.鉴

于此,在综合分析当前软件版权保护机制和原理的基础上,基于第三方可信中心 CA 提出了一种能够有效解决上述问题的新颖的动态安全许可协议 CPSec,将软件内容与软件运行环境联合考虑,通过一种“特征关联,原子授权,强制收权”机制有效地控制版权,从而解决了软件使用权的原子发授和回收、软件资源任意迁移和软件内容完整保持问题.

1 商用软件版权保护系统的基本要求

1.1 威胁版权的潜在因素

版权意义下软件资源分为受限软件和非受限软件两种,针对这两种软件的潜在威胁因素主要包括:

(1) 非受限软件的非法复制与扩散.

(2) 非受限软件的无限制安装使用.

(3) 受限软件内容的跟踪、调试、分析与破解,并非法复制扩散和无限制使用.

1.2 版权保护系统应解决的问题

按照 EULA 许可协议,一种安全、可靠、实用的版权保护系统至少应达到如下几个基本要求:

(1) 使用权的可控性.即对于一套软件必须能够控制其在许可协议申明的数量范围内运行.最简单的情况是,一套软件只能允许在一台硬件环境中运行.而对于作为服务器使用的软件只能允许其支持许可协议声明的客户端数量.即应能保证软件内容的非法复制与扩散无利可图.

(2) 软件使用环境的可迁移性.即必须支持同一软件副本在不同机器之间无差异地迁移,并保持原有软件功能的完备性和内容的完整性.

(3) 软件资源的可控性.软件副本在多台机器之间迁移时,应保证软件资源的完整或部分回收,并至少保证遗留软件资源不足以构成对版权的威胁.

(4) 软件实体的运行安全性.软件实体必须从实现机制上保证其运行安全性以抵御外部的分析和攻击,如进行反跟踪和反汇编以增强程序的自我保护能力.

2 商用软件非法复制和扩散的分布式安全认证协议 CPSec

2.1 CPSec协议思想

针对上述存在的问题,按照 EULA 协议的要求,本文在综合分析商用软件版权控制的诸多机制的基础上,提出了一种新颖的商用软件许可证分布式安全控制新方案 CPSec,其机理在于:

(1) 软件供应商 SP 为每一套要发授的软件赋予一个特征标识 SID,并将该 SID 交给第三方可信中心 CA(模型中包括特征认证服务器、许可证管理服务器以及证书库);

(2) 用户在使用软件时,首先通过许可证申请代理程序提交他从 SP 获得的 SID,以及能惟一标识 SID 运行的硬件特征 HID,申请软件的运行必需的一份许可证以获得服务.

(3) 当用户需要迁移软件的运行环境时,用户首先需要向 CA 申请注销口令,然后注销本机资源,最后提交注销码,当完成全程事务性操作以后,完成本机软件资源的注销,并重新获得用于在另外机器上的许可证.

CPSec 协议模型如图 1 所示.其本质上是以一种“特征关联,原子授权,强制收权”的分布式安全控制机制,通过软件许可证对版权进行有效管理控制,从而起到版权保护的作用.该协议使得同一软件无论复制多少,许可证服务器只能允许一套获得许可证,从而使非法复制和扩散无利可图.这可以有效解决软件授权的原子授权、软件资源的任意迁移和资源的原子保持问题.CPSec 符合 EULA 协议要求,有效解决了软件内容的非法复制与扩散而造成的盗版问题,为软件版权保护提供了一种新思路.

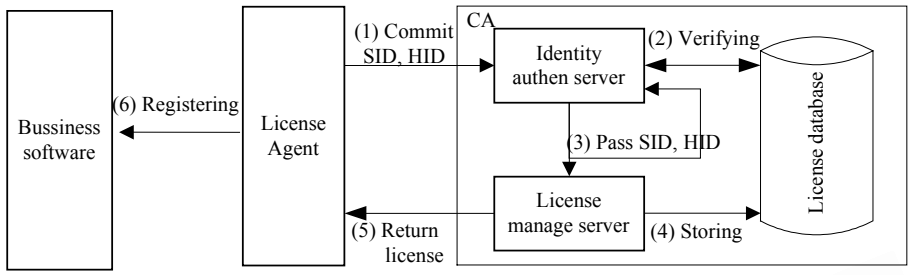

Fig.1 Conceptional model of CPSec

图 1 CPSec 协议概念模型

2.2 数据约定

为了便于准确描述 CPSec 协议,有必要对所需的各种数据对象进行无歧义表述,为此,下面我们先对 CPSec 协议中用到的数据进行准确规约.

定义 1. 软件特征(SID)是指用于区分不同软件提供商生产的软件实体以及同一软件供应商生产的同一软件的不同副本的特征标识.

定义 2. 硬件指纹(HID)是指用于区分软件使用者身份的关联于机器硬件特征的身份标识.

关联特征 AID 是指具有软件特征为 SID 的软件在具有 HID 的硬件运行环境下运行时所表现出的一种动态数据特征.在本文中,将软件的关联特征定义如下:

定义 3. 如果 S,H 分别表示 CPSec 方案中的软件标识集合、硬件标识集合,则对应于 SID 和 HID 的关联特征(AID)定义为,将 SID 和 HID 按照串接方式形成的特定格式化数据结构.事实上,由串接规则 f 是 $S\times H\to A$ 上的一个单值函数(A 为关联特征集合)可知,串接意义下所形成的关联特征具有惟一性.

为了便于表述,将用户向许可证管理服务器提交的序列化数据定义为请求向量.

定义 4. 用户向许可证管理服务器提交的消息是一个请求向量 Req={AID,OP,$R_1,R_2,\ldots,R_n$},其中,AID 为软件的关联特征,且 AID=SID||HID;而 OP 是操作类型,OP={REQ_LIC,REQ_LPSW,COMMIT},其分别代表申请许可证、申请注销口令以及提交注销结果码操作;$R_i(0\le i\le n)$是保留字.

定义 5. 软件许可证(SL)是指用于控制软件版权,由 CA 通过关联特征为用户动态生成的、加密并签名的复合数据结构.证书状态 LS={UNRELEASED,RELEASED,WAITING}.其对应的语义解释分别为:UNRELEASED 表示未为对应的软件发放许可证,包括两种情况,即许可证从未发放和发放后又被最终注销;RELEASED 表示对应的许可证已经发放;WAITING 表示正在等待对应许可证注销结果.

为了有效地控制和保证许可证的原则性,在发放证书时,若软件特征与软件供应商提供的相一致,即客户第一次申请许可证书,或者合法客户多次在同一台机器上请求许可证,则认为是合法的请求,故为其发放证书.否则,即使请求时发送的软件特征与软件供应商提供的相一致,而该软件没有注销,系统也将此请求看做是多复本使用该软件而拒绝发放许可证,其他情况,如软件特征不存在,则自然无法申请到许可证,即:

定义 6. 若 R,S 分别表示请求向量集和证书状态集,对于$\forall r$=Req(SID,HID,REQ_LIC,$R_1,R_2,\ldots,R_n$)$\in R$,如果记认证服务器已保持的特征数据和证书状态分别为 SID′,HID′,s',则证书发放决策函数f_{LRD}是请求向量 r 和证书状态 s 决定的一个多值决策 f_{LRD}: $R\times S\to S$:(R:发放许可证(**R**elease),D:拒绝(**D**eny)).

$$f_{\mathrm{LRD}}=\begin{cases}R, & \text{if } ((SID=SID'\wedge HID=NULL\wedge s=UNRELEASED)\vee(SID=SID'\wedge HID=HID'))\\ D, & \text{otherwise}\end{cases}.$$

在注销证书时,当且仅当该软件已被合法的用户注册后才可以允许其注销.其他情况下(如许可证未发放客户请求注销以及许可证已发放但第三方客户却想注销合法),均拒绝提供注销服务,结合定义 5,证书注销决策可表示如下:

定义 7. 对于$\forall r$=Req(SID,HID,REQ_LPSW,$R_1,R_2,\ldots,R_n$)$\in R$ 以及 SID′,HID′,s',软件供应商发放注销口令的决策可以表示如下:

$$f_{\mathrm{LVD}}=\begin{cases}R, & \text{if } (SID=SID'\wedge HID=HID'\wedge s'=RELEASED)\\ D, & \text{otherwise}\end{cases}.$$

定义 5、定义 6 明确规定了发放和注销许可证的完整决策.上述过程是在广域网络环境下的分布式应用,为了预防潜在的攻击和诱骗,需要对客户服务器交互中的身份以及数据的安全性、完整性等加以保证.为此,借助于密码方法,通过对称加密算法保证数据的安全性和完整性,而以数字签名解决客户服务过程中的数据的不可否认性.下面根据不同决策,结合密码技术,分别精确阐述 CPSec,随后给出其有效性和安全性证明.

2.3 CPSec符号约定

(1) 实体标识

U:用户或代理　　AS:身份认证服务器　　CS:许可证服务器

V:目标服务程序　　DS:数据库服务器

(2) 特征参数

SID,HID:用户提交的软件特征、硬件特征　　SID′,HID′:DS 存储的软件特征、硬件特征

(3) 认证参数

L:软件许可证　　T_{start}:申请证书时间　　T_{cur}:时戳　　Lifetime:许可证有效期限

P_{Lgf}:注销所需口令　　R_{Lgf}:注销结果认证码

(4) 完整性验证参数

$K_{A,B}$:A,B 共享对称密钥　　Ka,Ka':分别表示 a 的公钥和私钥　　E,D:分别表示加密和解密

$\mathrm{Sig}_K(m)$:表示以 K 对 m 的签名　　$\mathrm{Ver}_K(s)$:表示以 K 验证 s 的签名

Nonce:随机量　　N_0:协议交互中 Nonce 的增量值

2.4 CPSec协议描述

根据上述思想,许可证分布式安全控制协议 CPSec 从用户注册和迁移的角度可以划分为软件注册子协议 SRP(software register protocol)以及软件迁移子协议 SMP(software migration protocol)两个子协议.鉴于公平性和责任可证明性等信用考虑,上述方案假定:

(1) 软件提供商是诚实的,它提供给 CA 的 SID 与其发布给用户的 SID 保持一致.

(2) CA 是公平、可信的,在软件实现上是安全的.

(3) 在协议中,通信数据完整性通过安全对称密码算法保证;数据不可否认性通过公钥算法验证;系统时戳由服务器方发放并控制;时戳验证需向服务器方联机.

(A) 软件注册子协议 SRP

阶段Ⅰ:申请许可证

SRP 协议完成许可证申请,U 从 AS 获取许可证 C,SRP 分两个阶段,共 5 步,协议描述如下:

Step1: U 通过网络向 AS 提交许可证申请:U->AS:$R'=E_{K_{U,AS}}$ (SID||HID||Nonce).

Step2: AS 认证 U 所提交信息的合法性:

AS: $D_{K_{U,AS}}(R')$,AS 检索系统数据库,若 SID=SID′不成立,则拒绝并返回,否则转 Step3.

Step3: 通过对客户请求的认证,AS 将请求向量[SID||HID||Nonce]提交至许可证管理服务器 CS,CS 根据许可证发授决策函数 f_{LRD} ,决定是否发放许可证,即:

(1) 若 S_{Lic}=UNRELEASED (此时 HID=NULL),则 CS 动态生成许可证 L 并用 V 的公钥 K_V 加密,然后随机生成 U,V 会话密钥 $K_{U,V}$,加密送至用户 U;

① CS:生成许可证 L,$L=E_{KV}$(SID||HID||P_{run}||T_{start}||Lifetime),其中 $P_{\mathrm{run}}=S_E||R$,S_E 为带版本号的密签,R 为程序正常运行向量.

② AS->U:AS 将 U 的 Nonce 做运算 Nonce′:=Nonce+N_0,并对 L 签名,然后加密传送至 U:

$$L'=E_{K_{U,AS}}(\mathrm{Nonce}'||L||K_{U,V}||\mathrm{Sig}_{Ku'}(L));$$

③ DS:置证书状态 S_{Lic}=RELEASED,将发放许可证的时戳 T_{start},有效期 Lifetime,许可证版本 P_{run} 等写入系

统数据库 DS;

④ 注册代理 U 从 AS 得到许可证 L 的信息,先验证随机向量 Nonce 的复位值,如果 Nonce′−N_0=Nonce 不成立,则拒绝接受,否则说明上述内容确实来自 AS;进而验证 L 的签名 $Ver_{K_V}(Sig_{K_{V'}}(L))$,若通过验证则接受,否则,拒绝接受 L.

(2) 若 S_{Lic}=RELEASED 或 WAITING,则表明该 SID 的许可证已发放或等待注销结果,此时鉴权当前客户请求分量 HID.

① 若 HID=HID′(同一软件在同一机器上重新安装),则 CS 检查 $T_{\text{cur}}-T_{\text{start}}$>Lifetime 是否成立,若不成立,表明在注册有效期内多次申请许可证,拒绝发放许可证.否则按上面(1)为用户发放许可证.

② 若 HID≠HID′,拒绝为发授许可证(即认为一份软件的多份拷贝在不同的机器上安装或者同一软件实体在多台机器上安装).

阶段Ⅱ:注册软件

U 在时间 Lifetime 通过 V 的会话密钥 $K_{U,V}$ 注册目标程序,得到程序服务,然后作废许可证.

Step4: ① U->V:U 向 V 提交申请到的许可证 L,V 用自己私钥 Kv 解密许可证 L,$D_{Kv}(L)=D_{Kv'}(E_{Kv}(\text{SID}||\text{HID}||P_{\text{un}}||T_{\text{start}}||\text{Lifetime}))$,并向许可证服务器 CS 联机,提交 T_{start}||Lifetime,CS 根据自己当前时间 T_{cur} 计算 $T_{\text{cur}}-T_{\text{start}}$<Lifetime 是否成立,并将结果送至 V.如果超时,V 拒绝注册,否则转②;

② U->V:U 向 V 提交认证码 $\text{Authenticator}_u=E_{Ku,v}(\text{HID}_C||\text{SID}_C)$,其中 SID_C,HID_C 分别表示 U 提交的其持有的软件标识和当前机器标识.

Step5:V 验证 U 提交的认证码合法性,并用自己私钥解密的许可证与用户提交的注册信息比较以决定是否允许注册并开放服务.

③ V:$D_{Kv'}(L)=D_{Kv'}(E_{Kv}'(\text{SID}||\text{HID}||P_{\text{run}}||T_{\text{start}}||\text{Lifetime}))$

$D_{Ku,v}(\text{Authenticator}_u)=D_{Ku,v}(E_{Ku,v}(\text{HID}_C||\text{SID}_C)$;

④ 若 HID=HID_C 且 SID=SID_C,则转⑤,否则拒绝注册返回;

⑤ 注册目标软件 V,注入程序正常运行向量 P_{run} 并开放服务,作废许可证并以密文形式日志此次注册内容.

(B) 软件迁移子协议 SMP

考虑可能的攻击和欺诈,用户迁移协议必须遵循基准:

(1) 为了防止用户的欺诈,在许可证失效前的时间段内多次申请针对一套软件的多份许可证,必须在许可证失效后才可以申请注销口令.

(2) 为了确保合法软件不被随意注销(甚至包括恶意抢注,即在注册后 SID 不慎丢失后,第三方恶意试图注销),在注销时需先向许可证管理服务器请求注销口令 P_{Lgf},然后以事务方式(具有 ACID 性质)注销并清理本机资源,事务完成后,再将注销结果码 R_{Lgf} 返回给许可证管理服务器并更新系统数据库,完成注销任务.必要时重新申请许可证.软件迁移协议 SMP 分两个阶段,共 5 步,协议描述如下:

阶段Ⅰ:申请注销口令

Step1: U 向 AS 提交申请注销口令请求: U->AS: $R'=E_{K_{U,AS}}(\text{SID}||\text{HID}||\text{Nonce})$

Step2: 类似于 LAP 验证身份,若不通过则拒绝,否则转 Step3;

Step3: 通过对客户的认证,CS 检查对应于 SID 的证书状态 S_{Lic},并根据证书注销决策函数 f_{LVD} 以及注销时效决定是否为 U 发放注销口令:

(1) 若 S_{Lic}=RELEASED 或 WAITING 且 HID=HID′,则验证是否在许可的注销时效范围内,即检验 $T_{\text{cur}}-T_{\text{start}}$>Lifetime 是否成立,如果不成立,则拒绝注销,否则表明许可证已超过注册时效(在该条件下用户在许可证有效期内不可能多次申请一份软件的多份许可证).此时,AS 动态生成许可证注销口令加密后发送至用户,并将关键数据写入系统数据库 DS:

① AS:生成注销口令 P_{Lgf}:$P_{\text{Lgf}}=E_{Kv}(\text{SID}||\text{HID}||P_{\text{Stop}}||W_{\text{BCK}})$,其中 $P_{\text{Stop}}=S_E||P$,S_E 为带版本号的密签,P_{Stop} 为程序停止运行向量,而 W_{BCK} 为用户用于注销的回执字.

② AS->U:AS 置证书状态 S_{Lic}=WAITING,并置 Nonce′:=Nonce+N_0,然后将注销口令签名后加密传送至 U,

即:$C'=E_{K_{U,AS}}(\text{Nonce}'||P_{\text{Lgf}}||K_{U,V}||\text{Sig}_{Ku'}(P_{\text{Lgf}}))$.

③ U从AS得到注销口令信息,与注册协议一样,先验证Nonce′−N_0=Nonce是否成立,若不成立,则拒绝接受,否则说明上述内容确实来自AS;U进而验证L的签名$Ver_{K_V}(Sig_{K_{V'}}(P_{\text{lg}f}))$,若通过验证,则接受,否则拒绝接受$P_{\text{Lgf}}$.

(2) 若S_{Lic}=UNRELEASED或者S_{Lic}=RELEASED但HID≠HID′,则拒绝发放注销通行字.

阶段Ⅱ:注销软件资源

用户以获得的口令来注销本机资源,注销成功后,提交注销结果给许可证管理服务器AS,AS确认后方可再申请并在另外的机器上安装并运行目标软件.

Step4: 用户解密获得的注销口令,与注册过程类似,先解密并验证签名,若签名为真,则执行如下步骤:

① U->V: P_{Lgf}||Authenticator$_u$,其中,Authenticator$_u$=$E_{Ku,v}$(HID$_C$||SID$_C$);

② V:$D_{Kv'}(P_{\text{Lgf}})=D_{Kv'}(E_{Kv}(\text{SID}||\text{HID}||P_{\text{Stop}}||W_{\text{BCK}}))$,

$D_{Ku,v}(\text{Authenticator}_u)=D_{Ku,v}(E_{Ku,v}(\text{HID}_C||\text{SID}_C))$;

③ 若HID=HID$_C$且SID=SID$_C$,则转Step5,否则拒绝注销并返回;

Step5: ① 用户以此通行字P_{Lgf}注销本机软件资源,注销目标软件(彻底卸载目标软件并清理软件运行所有资源),以事务形式强行检查注销过程是否完成(遍历搜索目标软件的相关资源是否残存,如果存在未卸载的软件资源,一并自动删除),事务完成后,作出注销判决:R=Logoff(SID||HID||P_{Lgf}).

其中R=bLogoffed||R_{Lgf}.若bLogoffed为假,则表明注消失败,此时R_{Lgf}=NULL;否则$R_{\text{Lgf}}=E_{Kv}(\text{SID}||\text{HID}||W'_{\text{BCK}})$,$W'_{\text{BCK}}$为$V$从注销口令中解析出的回执字,$U$将$R_{\text{Lgf}}$返回给AS:$U$->AS: $E_{K_{U,AS}}(\text{SID}_C||\text{HID}_C||R_{\text{Lgf}})$,SID$_C$,HID$_C$为当前软硬件特征.

② AS用其公钥解密R_{Lgf},并检索系统数据库,验证注销结果的合法性,若SID$_C$=SID,HID$_C$=HID,且W'_{BCK}=W_{BCK}同时满足,则接受注销结果并置HID′=NULL,S_{Lic}=UNRELEASED;否则,拒绝接受注销结果.

3 协议分析

定理 1. 用户U提交合法的软件标识SID以及HID,通过SRP子协议能且仅能获得适用于一台机器注册的软件许可证L,并且能够以L通过目标保护软件认证而得到服务.

证明:由SRP知,U向AS提交请求向量,即Req=$E_{K_{U,AS}}$(SID||HID||Nonce).根据f_{LRD}可知,当且仅当$(SID=SID'\wedge HID=NULL\wedge S=UNRELEASED)$或$((SID=SID'\wedge HID=HID')\wedge(T_{\text{cur}}-T_{\text{start}}>\text{Lifetime}))$时,AS才为$U$发放许可证.而当对应于SID的证书一旦发放后,若SID对应的软件没有注销,则在其他机器上无法申请到许可证,这样就保证许可证的原子性发授;而当用户申请到许可证后,在注册有效范围内($T_{\text{cur}}-T_{\text{start}}<$Lifetime)重复申请将无法得到许可证,这使得用户重放攻击不能得逞.从而用户能且仅能获得一份许可证L,即许可证发授具有原子性、惟一性.

SRP各步操作均加密传输,在算法安全的前提下,系统能够安全通信.在用户U申请许可证时,通过判断随机向量Nonce的复位值Nonce′−N_0=Nonce是否成立来确认AS身份的真实性,再根据CA的可信性,U能够得到有效的许可证L;虽然U可以篡改L,但篡改L使其将无法通过V的签名验证$Ver_{K_V}(Sig_{K_{V'}}(L))$,从而许可证的完整性得到保证;再有,根据假设AS是可信的,则L的数据内容是有效的.因此在SRP协议下,U必然能够安全地得到许可证,使得数据完整性和不可否认性得到保证,从而U必然能够通过目标服务程序的认证并得到所需服务.

在许可证申请以及注册过程中,考虑各种可能的攻击,若用户U在未真正注销本机资源就想在许可证有效期内申请一份软件的多个许可证,由于系统时钟由服务器方控制,U无法修改许可证有效期Lifetime,也无法修改服务器方的时戳T_{Start}以及T_{Cur},从而重放攻击无法获得许可证,因此许可证必然在Lifetime时效之外失效;再者,若用户注册完本机资源又欲在另外的机器安装并使用目标软件,则他必须重新申请新的许可证,这时SID的许可证已发放,只有注销本机资源才能获得在另外一台机器上的许可证.

综合上述3方面因素可知,SRP协议中许可证的发放具有原子性,许可证内容具备完整性和不可否认性,注册具有时效性,从而在此协议下用户仅能获得在一台机器上的合法授权并得到相应的服务. □

定理 2. 在SMP协议下,合法用户U请求能够获得注销口令P_{Lgf},在完整注册事务完成之后,必然能够实现

软件资源的迁移.

证明:类似定理 1,在 SMP 协议控制下 U 能安全地获得注销口令 P_{Lgf}.事实上,根据 SMP,只要合法地注销请求,他必然能够得到注销口令,这是迁移软件资源的第 1 阶段;在第 2 阶段,注销本机资源属于事务性操作,一旦注销,要么都彻底完成注销,要么恢复到未注销状态;注销完成后,通过注销口令中的回执字 W_{BCK} 以及用户 U 提交的 SID_C,HID_C 来确认是否真正注销本机资源.不通过注册代理程序,用户若想伪造注册结果码,即意味着猜测 W_{BCK},其难度等价于攻破公钥算法,在算法安全的前提下其成功的概率为 0.这样,AS 能够正确识别来自于 U 的注销结果的合法性和真实性,从而确保软件资源迁移过程中资源的安全性和语义的完整性.从而在 SMP 协议许可的情况下,必然能够完成软件资源的自由迁移,而版权仍然得到有效控制. □

定理 1 和定理 2 说明,CPSec 协议能够确保"不论复制软件内容的多少副本,但仅有一个副本能够获得许可证".从而在算法安全的前提下能够按照 EULA 协议有效地保证版权不受侵犯.需要说明的是,CPSec 协议的有效性、安全性、数据完整性以及内容的不可否认性不仅依赖于密码技术,同时,资源许可和注销的实现建立在基于事务(ACID)的软件实现可行性上.从而协议既有理论上的安全性,同时也具备实现上的可行性.

4 CPSec 协议完整实现

根据 CPSec 协议,我们遵循分层化、模块化和快速重构性原则,采用基于组件的开发方式,以 Microsoft Visual C++6.0 为前端开发工具,而以 Microsoft SQL 2000(Enteriprise Edition)作为后台数据库支持,完整实现了普适于 Internet/Intranet 的软件版权安全管理系统 SmartLicManagment 3.0.该系统由客户端代理 SmartLicMangager——Client、身份认证服务器 AuthenServer、许可证管理服务器 SmartLicManager——Server 这 3 个程序实体(*.exe)和一个可动态集成到被保护软件实体的访问控制库 SmartMonitor.dll(lib)组成.其中 SmartLicManager(Server)与数据库服务器进行数据交互;而业务无关的独立密码算法库(ICL)作为公用模块以动态链接库(*.dll 和*.lib)的形式为提供加密解密支持,CPSec 系统的软件体系结构如图 2 所示.

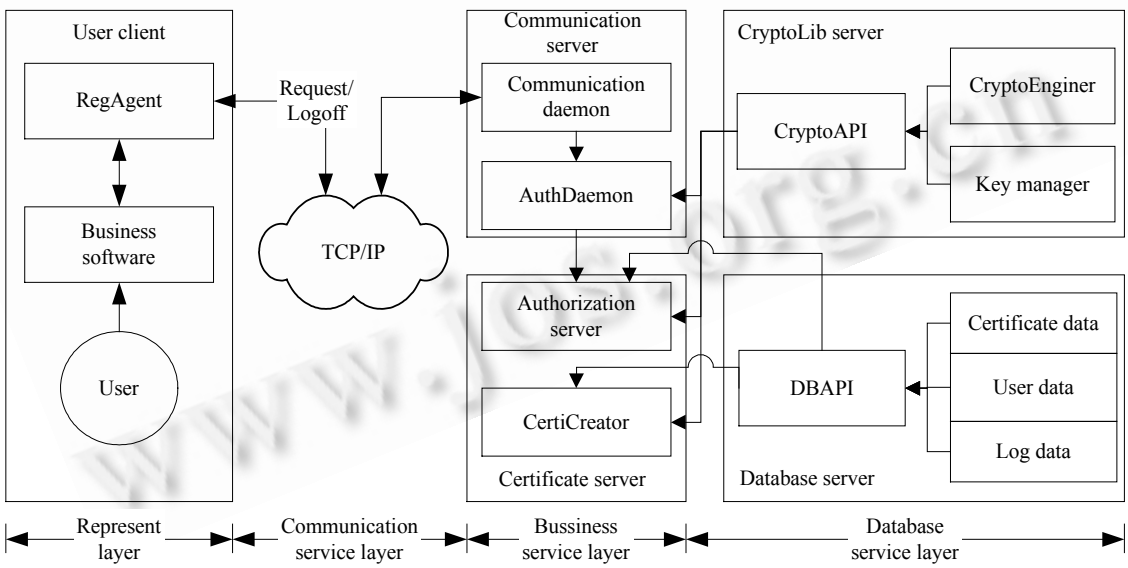

Fig.2 Multi-Layer architecture of CPSec

图 2 CPSec 多层体系结构

在 CPSec 核心模块 ICL 中,签名算法采用 RSA-SHA1,对称密码算法采用 IDEA(避免 DES 诸多弱密钥[17,18]).ICL 在实现上没有采用 Microsoft CryptoAPI 或 Java 提供的密码类库,而是主体程序采用纯 C 语言,关键时运算采用汇编语言自主实现算法.CPSec 系统基于 GUID 生成软件特征;通过提取机器的 3 个关键特征,即中央处理器序列号、硬盘序列号以及网卡序列号,通过复合逻辑运算、散列函数压缩合成机器的特征标识

HID(若无网卡,则设置对应的缺省值).为了防止外部攻击(如 SoftICE)而跳过关键控制代码,实现上采用一种增强型签名验证方法,即“分段校验签名随机决定运行地址”实现目标程序关键代码的完整性和安全性保证.

在实现上,SmartLicManager 系统核心功能,如数据加密和解密、密钥管理以及认证查询 SID,HID 的管理等以组件的方式透明地对外提供高层服务,这使得系统的整体运行性能得到了极大的提高,且系统具有很好的可重构性、可移植性和可扩展性.

5 相关工作比较

与目前已有的商用软件保护措施,如软件加/脱壳、序列号、注册码、基于时间、功能限制、使用次数限制等技术相比,CPSec 系统能够以较弱的限制有效地抵制非法复制与扩散,而且支持许可证的原子性发授、软件资源的迁移和回收等,达到了 EULA 协议的要求.表 1 和表 2 是 CPSec 与其他方案的性能对照表.

Table 1 CPSec contrasts with pure software solutions

表 1 CPSec 与纯软件方案的比较

Pure software solutions	Associate with machine character	Crack and attack supports	License supports	Usage restricts	Software resource revokes	Feasible to use when copied	Examples
Sno, CDKey	No	No	No	No	No	Yes	Acrobat, Office
Key file	No	No	Yes	No	No	Yes	Jbuild7/KeyGen
Time (s)	No	No	No	No	No	Yes	Xfile, LeapFtp
Function	No	No	No	No	No	Yes	WinISO
CPSec*	Yes	Yes	Yes	No	Yes	No	CPSec

Table 2 CPSec contrasts with compound solutions

表 2 CPSec 与综合件方案的比较

Compound solutions	Associate with machine character	Crack and attack supports	License supports	Usage restricts	Software resource revokes	Feasible to use when copied	Examples
Floppy-Type	Yes	No	Yes	Yes	No	No	Rising,KV3000,etc.
Disk-Type	Yes	No	No	No	No	No	SaftDisk[13]
ROM-Type	No	No	No	Yes	No	No	Rainbow[14]
Puresoft-Type	Yes	No	Yes	Yes	Yes	No	Crypkey, Softkey[15,16]
CPSec	Yes	Yes	Yes	No	Yes	No	CPSec

6 小 结

版权保护是数字时代软件产业有序发展的一个重要前提,仅仅通过文字上的约定不可能真正有效地解决问题.本文提出的 CPSec 协议以 EULA 为基准,通过“特征关联,原子授权,强制收权”的机制有效地解决了软件版权的安全管理.本文的主要贡献在于:

(1) 将软件内容本身和软件运行硬件环境以及软件版权状态结合考虑,提出了动态许可证的概念,用于软件版权的灵活控制,使得软件版权的原子性授权、强制收权、软件运行环境的迁移得到了有效控制.

(2) 引入第三方可信中心,以保证版权认证过程的公平性、可信性以及责任可证明性.

(3) 在软件实现过程中,通过加密和数字签名保证分布式环境下数据的安全性和完整性,而以代码随机验证签名确保版权管理软件本身的安全性.

CPSec 达到了 EULA 协议的要求,为软件版权保护提供了一种有效的实现机制.然而,EULA 并没有考虑支持多个客户端的许可证管理问题,支持多个客户端运行的信任版权管理是一个值得进一步研究的重要课题.

致谢 本文作者对那些匿名审稿老师严谨、审慎的评审表示诚挚的感谢.

References:

[1] Bordoloi B, Ilami P, Mykytyn PP, Mykytyn K. Copyrighting computer software: The “look and feel” controversy and beyond. Information & Management, 1996,30(5):211~221.

[2] Milagros DC. Copyright in the digital environment. International Information & Library Review, 1997,29(2):201~204.

[3] Su JK, Hartung F, Girod B. Digital watermarking of text, image, and video documents. Computers & Graphics, 1998,22(6): 687~695.

[4] Ruanaidh JO, Petersen H, Herrigel A, Pereira S, Pun T. Cryptographic copyright protection for digital images based on watermarking techniques. Theoretical Computer Science, 1999,226(1-2):117~142.

[5] Chi CH, Lin Y, Deng J, Li X, Chua TS. Automatic proxy-based watermarking for WWW. Computer Communications, 2000,24(2): 144~154.

[6] Lin PL. Digital watermarking models for resolving rightful ownership and authenticating legitimate customer. Journal of Systems and Software, 2001,55(3):261~271.

[7] Lee WB, Chen TH. A public verifiable copy protection technique for still images. Journal of Systems and Software, 2002,62(3): 195~204.

[8] Eskicioglu AM, Delp EJ. An overview of multimedia content protection in consumer electronics devices. Signal Processing: Image Communication, 2001,16(7):681~699.

[9] Waller AO, Jones G, Whitley T, Edwards J, Kaleshi D, Munro A, MacFarlane B, Wood A. Securing the delivery of digital content over the Internet. Electronics & Communication Engineering Journal, 2002,14(5):239~248.

[10] Biehl I, Meyer B. Cryptographic methods for collusion-secure fingerprinting of digital data. Computers and Electrical Engineering, 2002,28(1):59~75.

[11] Oz E. Acceptable protection of software intellectual property: A survey of software developers and lawyers. Information & Management, 1998,34(3):161~173.

[12] Yoon K. The optimal level of copyright protection. Information Economics and Policy, 2002,14(3):327~348.

[13] SafeDisk: 128bit on-the-fly disk encryption. http://www.guardcomplete.com

[14] Rainbow. Internet security software, security software, software encryption protection, password protection. http://www.rainbow.com

[15] Crypkey. Copy protection and license control. http://www.crypkey.com

[16] Softkey. The key to software development success. http://www.rockey.com.my/index.htm

[17] Wang YM, Liu JW. Security of Communication Net: Theory and Technology. Xi'an: Xidian University Press, 1999 (in Chinese).

[18] Schneier B. Applied Cryptography-Protocols, Algorithm and Source Code in C. 2nd edition, New York: John Wiley & Sons Inc., 1996.

附中文参考文献:

[17] 王育民,刘建伟.通信网的安全——理论与技术.西安:西安电子科技大学出版社,1999.