doi:10.3969/j.issn.1000-436x.2012.z2.024

基于软件水印的云平台下软件服务保护安全协议

许金超，曾国荪，王伟
（1. 同济大学 计算机科学及技术系，上海 201804；2. 同济大学 嵌入式系统与服务计算教育部重点实验室，上海 201804）

摘 要：为了抵抗云中软件服务来自运行环境内部的侵权问题，提出了一种基于软件水印的云平台下软件服务保护安全协议。协议结合云计算的实际需求，引入了可信的软件水印服务云负责软件水印的嵌入和提取。协议不仅满足软件版权保护的基本要求，而且给出了可操作的追究策略，提高了抗攻击的能力。分析表明提出的协议具有较强的安全性。

关键词：云计算；数字版权；软件水印；安全协议
中图分类号：TP391 **文献标识码**：B **文章编号**：1000-436X(2012)Z2-0176-06

Software services protection security protocol based on software watermarking in cloud environment

XU Jin-chao, ZENG Guo-sun, WANG Wei
(1. Department of Computer Science and Technology, Tongji University, Shanghai 201804, China;
2. The Key Laboratory of Embedded System and Service Computing Ministry of Education, Tongji University, Shanghai 201804, China)

Abstract: Aimed to resist the infringement of software services in cloud environment, a software services protection security protocol based on software watermarking in cloud environment was presented. On the consideration of actual needs in cloud computing, the protocol introduced the software watermarking as a service cloud which was responsible for software watermark embedding and extraction. The protocol not only met the basic requirements of software copyright protection, but also gived the detailed strategy to solve the infringement, and hardened the ability of the software watermarking to resist various attacks. The analysis shows that the proposed protocol has higher security.

Key words: cloud computing; digital copyright; software watermarking; security protocol

1 引言

云计算的时代已经开启，它将大量的计算资源、存储资源和软件资源集中在一起，为用户提供“无限资源、无限计算能力”的云服务。用户能够充分利用云带来的便利，直接利用云平台下的资源进行运算和存储。根据云计算所处的不同层次，云计算可以分为软件即服务 SaaS、平台即服务 PaaS、设施即服务 IaaS 3 种。其中，SaaS 作为最贴近最终用户的云计算，受到了众多用户的青睐。不仅一些

收稿日期：2012-10-23
基金项目：国家高技术研究发展计划（“863 计划”）基金资助项目（2009AA012201）；国家自然科学基金资助项目（61272107，61103068）；NSFC-微软亚洲研究院联合基金资助项目（60970155）；上海市优秀学科带头人计划基金资助项目（10XD1404400）；教育部博士点基金资助项目（20090072110035）；教育部网络时代的科技论文快速共享专项研究课题基金资助项目（20110740001）
Foundation Items: The National High-Tech Research and Development Program of China(863 Program) (2009AA012201); The National Natural Science Foundation of China (61272107, 61103068); The Joint of NSFC and Microsoft Asia Research Program (60970155); The Program of Shanghai Subject Chief Scientist (10XD1404400); The Ph.D. Programs Foundation of Ministry of Education(20090072110035); The Special Fund for Fast Sharing of Science Paper in Net Era by CSTD (20110740001)

个人软件开发者会将自己的软件部署在云端提供共享服务，许多企业也会将自己的软件放在云中作为云服务供客户使用。从客户的角度来看，这样可以省去在服务器硬件和软件授权上的投入；从软件拥有者的角度来看，这样只需在云平台上维护一个应用服务实例，就可以大幅降低运营成本，从而实现客户和软件拥有者的双赢。

然而云计算带来便利的同时伴随着极大的安全风险，其中一个最大的问题是云服务提供商对用户上传的数据和服务有绝对的控制权和优先访问权。尽管不少研究认为云服务商应该是可信的[1]，但是注意到目前对于可信云服务提供商尚没有评判的标准，即使是较大的云服务商也并非绝对安全，如 2009 年 3 月，Google 就曾发生大批用户文件外泄事件。对于云中运行的用户应用和存储的用户程序资源，仍然需要利用一定的技术手段加以保护。

许多研究是从已有的软件保护技术出发进行云计算中的软件保护[2]，软件水印技术也被应用到了云计算中。软件水印是一种传统的软件保护技术。它通过将标志版权的秘密消息嵌入到待保护的软件中，并可以在需要的时候提取出来验证软件的版权归属。由于软件水印不影响软件的正常使用，且可以对盗版进行追踪，因此成为了研究人员关注的热点，目前已有多种软件水印算法提出[3~8]。然而软件水印技术在云平台中的应用与传统应用下存在许多不一致的地方，在传统应用下，版权所有者拥有对自己软件的全部控制，而在云环境下，软件的主动权掌握在云服务提供商手里。在版权所有者和云服务提供商都不可信的情况下，利用软件水印保护软件需要用户和云服务提供商的共同参与。已有研究尝试结合云计算和软件水印技术。黄铠[2]提出了利用云模型对云中的软件数据进行染色。为了满足云平台下海量软件处理的需要，Yu[1]采用了 MapReduce 技术来实现软件水印的嵌入和提取过程。然而这些技术是将软件作为存储在 IaaS 云中的数据来看待的，而不是作为云平台下的对外服务，并且没有考虑来自于云平台内部的威胁，而这是本文试图解决的主要问题。

假设软件水印技术能够嵌入足够多的信息并且足够坚固，攻击者无法去除其中的软件水印或将一个软件水印修改成其他形式。在此前提下考虑如下情景问题。

用户 Alice 上传自己的软件到 PAAS 云 Cloud 中提供共享服务，这些服务可以供 Cloud 中的学术类用户免费使用，但只能供商业用户免费用到 2012 年 5 月 28 日。这种情况下仍然可能存在以下问题。

1) Cloud 中的软件商业用户使用期限已到，但恶意工作人员仍将软件私自供商业用户使用，事后否认。

2) Cloud 中的恶意工作人员将软件加上用户 Bob 的软件水印并代替云中原有的软件或者作为 Bob 的服务，并声称版权属于 Bob，此时如何判定 Cloud 的责任。

目前传统的软件水印技术还缺乏可信的应对手段来解决上述问题。和传统应用不一样，在云计算中涉及到多方之间的交互，因此为了解决上述问题，本文综合使用软件水印技术和密码学手段，建立云平台下软件水印的使用模型，设计了云计算下软件服务保护协议，并说明了文中提出协议的安全性。

2　问题表述

2.1　软件水印

本文的软件水印包括用户的标识信息、程序的描述和使用权限等信息，这些信息变换组合成一个有限长度的字符串，用 W 表示，设 P 表示待嵌入软件水印的程序。

软件水印嵌入过程用 embed 表示。

embed：$PWK \rightarrow P^W$

embed 用来将一个软件水印嵌入软件中，其中，未嵌入软件水印的程序 P 称作原程序，P^W 称作含水印的程序。

软件水印的提取过程用 extract 表示。

extract：$P^WK \rightarrow W$

extract 用来从含水印的程序中提取软件水印，本文假设无论 P^W 被攻击者做了怎样的修改，总是可以成功地从含水印的程序中完整地提取出原始的软件水印。

软件水印的检测过程用 detect 表示。

detect：$P^WKW \rightarrow$ True/False

detect 检测含水印的程序中是否已嵌入给定的软件水印，若是返回 True，否则返回 False。

2.2　系统模型

为了阐述本文提出的软件水印在保护云服务中的应用方式，本文不考虑用户注册、云费用支付等云计算中的常见操作，主要考虑和软件水印在用

于云服务保护时相关的过程。一个简化的系统模型如图 1 所示。

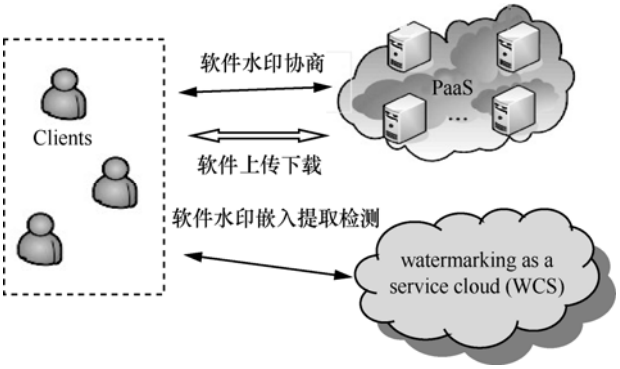


图 1 云计算中软件水印应用模型

图 1 中涉及的实体主要有以下 3 个。

1) 用户（Client）：用户可能是个人或者企业，它们是软件的版权拥有者，依赖云来存储自己的软件或在云中运行自己的软件供客户调用。

2) PaaS 云（CS, PaaS cloud server）：由商业化的云服务商提供并管理，一般有着强大的计算能力，为 Client 上传的软件提供运行平台。它有可信的认证中心颁发的证书，公钥和私钥分别为 PK_S、SK_S。

3) 软件水印服务云（WCS, watermarking as a service cloud）：由完全可信的云服务商管理。软件水印服务云使用各种公开的算法提供软件水印嵌入、检测和提取等服务，提供服务后不记忆任何使用软件水印服务的用户信息。WCS 也有可信的认证中心颁发的证书，设公钥和私钥分别为 PK_W、SK_W。

2.3 威胁分析

图 1 中模型存在的威胁包括来自云内部和外部两方面的威胁。

CS 提供了可供程序运行的应用平台，尽管其主观意愿是善意地为用户提供可靠的服务，然而不能排除内部个别雇员是恶意的，与外部攻击者串通将内部信息泄漏出去。也有可能 CS 自身使用的系统存在漏洞，从而使得攻击者能够成功入侵 CS，从而获得 CS 的操作权限，对 CS 中运行的服务造成威胁。

WCS 中提供的服务总是假设可信的，它有足够的能力保证自己的安全，能够经受任何外部的攻击。而且 WCS 仅提供单调的服务，减少了对外的接口，降低外界攻击的风险，从而提供可信的软件水印服务。

Client 也认为是不可信的，存在恶意的可能。如 Client 可能篡改自己软件中的软件水印，然后以此诬赖 CS 的不可信。

在 Client 与 CS、Client 和 WCS 的通信过程中，由于数据在网络中传输，难免经过不可信的节点，攻击者能够截获双方发送的消息，篡改或伪造传送数据，存在中间人攻击的风险，因此所有的通信过程涉及到敏感数据时都需要加密。

CS 中服务运行时产生的数据如内存中中间运算结果涉及到用户隐私，由于本文仅专注于软件在云平台下的服务版权保护以及服务是否按照软件版权拥有者要求的方式提供而不被滥用，因此本文不涉及软件服务运行中的数据保护。

3 利用软件水印保护云平台下服务版权的安全协议设计

协议设计目的是保证合法的有使用权限的消费者能够按照软件提供者规定的权限正确地使用软件服务，版权拥有者在发现版权受到侵害时能够确定产品版权的归属，规定的服务权限没有被云服务商正确地遵守时能够追究不遵守规则的相关实体责任。

为了行文方便，本节用到下述符号简化描述。

$K_{C,S}$ 表示 Client 和 CS 的会话密钥；$K_{C,W}$ 表示 Client 和 WCS 的会话密钥；$\{a,b,c\}_K$ 表示用密钥 K 加密消息 abc。

云平台下软件服务保护安全协议主要包括如下几个部分。

1) Client 和 CS 的软件水印协商

首先，Client 需要生成一对非对称密钥，设公钥为 PK_C，私钥为 SK_C。生成用户身份信息 ID_C；生成用户软件的标志 SID 和使用限制信息 SL；生成随机数 Ra；然后将 ID_C 和 SL 用 SK_C 签名后，和 PK_C 一起发送给 CS。

Step 1: Client→CS: $\{ID_C, SID, SL\}_{SK_C}$, PK_C, Ra

CS 接收信息后，用 PK_C 解密得到 ID_C、SID、SL，若不认可这些信息或者查找该 Client 的软件信息表发现 SID 曾经使用过，返回 False；否则，将 Client 的签名信息同自己的身份 ID_S 用自己的私钥 SK_S 签名，并生成传输密钥 $K_{C,S}$，和随机数 Ra 一起发送给 Client，并将 SID、ID_C 和 SL 存入该 Client 的软件信息表，作为以后按 Client 的要求运行该软件服务的依据。

Step 2: CS→Client: $\{\{ID_C, SID, SL\}_{SK_C}, ID_S\}_{SK_S}$, $\{\{K_{C,S}\}_{PK_C}, Ra\}_{SK_S}$

Client 获取 CS 公钥来解密判断自己的信息未被改动，并认可 CS 身份信息后将$\{\{ID_C, SID，SL\}_{SK_C}, ID_S\}_{SK_S}$作为软件水印 W。

2) Client 利用 WCS 嵌入软件水印 W

Client 传输给 WCS 的原始程序 P 和密钥ω是协议中最关键的信息，不能被任意 Client、WCS 外的第三者得知，因此 WCS 选择一个传输密钥 $K_{C,W}$。

Step1: Client→WCS: PK_C, Rb

Step2: WCS→Client: $\{\{K_{C,W}\}_{PK_C}, Rb\}_{SK_W}$

Client 验证随机数正确后获得传输密钥 $K_{C,W}$，然后将软件上传到 WCS，WCS 嵌入软件水印 W 和时间戳后送回 Client。

Step3: Client→WCS: $\{P, W, \omega\}_{K_{C,W}}$

Step4: WCS: embed$(P, \{W, t\}, \omega)=P^W$

Step5: WCS→Client: $\{P^W\}_{K_{C,W}}$

3) Client 上传嵌入软件水印后的程序 P^W 到 CS 中，CS 负责 P^W 的正确运行。

Step1: Client→CS：$\{P^W\}_{K_{C,S}}$

4) Client 利用 WCS 提取软件水印

和步骤 2）同样地，Client 和 WCS 重新协商一个传输密钥 $K_{C,W}$，然后 Client 申请 WCS 利用ω从 P^W 中提取软件水印。

Step1: Client→WCS: $\{P^W, \omega\}_{K_{C,W}}$

Step2: WCS: extract $(P^W, \omega)=\{W, t\}$

Step3: WCS→Client: $\{\{W, t\}_{SK_W}\}_{K_{C,W}}$

5) WCS 检测 Client 指定的软件水印 W 是否在软件 P^W 中

同样需要 Client 和 WCS 协商一个传输密钥 $K_{C,W}$，然后 Client 将相关信息传给 WCS，WCS 检测软件水印是否存在，然后返回检测结果。

Step1: Client→WCS：$\{P^W, \omega, W\}_{K_{C,W}}$

Step2: WCS: detect$(P^W, \omega, W)=\{BOOL, t\}$

Step3: WCS→Client: $\{\{BOOL, t\}_{SK_W}\}_{K_{C,W}}$

6) 对多 Client 进行版权验证

当多个 Client 声称对某个软件拥有版权时，则申请仲裁方（Judge）来判断版权归属。涉及版权的 Client 设有 Client A 和 B，分别将自己的程序 P_A^W、P_B^W 和密钥ω_A、ω_B提供给仲裁方，若 Client 双方提供的软件相同则仲裁方分别请求 WCS 获得双方的软件水印$\{W_A, t_A\}$和$\{W_B, t_B\}$，然后仲裁方分别验证返回的软件水印中的用户 ID 是否是版权请求者所有。

Step1: Judge: $\{W\}_{PK_S}=\{\{ID_C, SID, SL\}_{SK_C}, ID_S\}$,

Step2: Judge: $\{\{ID_C, SID, SL\}_{SK_C}\}_{PK_C}\rightarrow ID_C$，判断 ID_C 是否版权请求者标识。

如果两者嵌入的软件水印都是正确的，则仲裁方判断嵌入的软件水印时间先后确定版权所有者。

Step3: Judge: IF t_A 早于 t_B，RETURN A ELSE RETURN B

7) CS 侵权的追究

如果 Client 发现 CS 超出了 Client 规定的权限使用软件 P^W，则 Client 申请 Judge 进行侵权仲裁，Judge 要求 CS 提供 P^W，要求 Client 提供密钥ω，然后申请 WCS 返回私钥签名后的$\{W, t\}_{SK_W}$。

Step1: Judge:$\{\{W\}PK_W\}_{PK_S}=\{\{ID_C, SID, SL\}_{SK_C}, ID_S\}$，判断 ID_S 是否为 CS 标志

Step2: Judge: $\{\{ID_C, SID, SL\}_{SK_C}\}_{PK_C}=\{ID_C, SID, SL\}$，验证 ID_C 是否为 Client 标识，SL 是否包括 CS 当前使用软件的权限。

4　安全性分析

协议的主要目的有 2 个：1) 能够有效地证明 Client 对软件服务的所有权。2) 保证 CS 按照 Client 给定的权限提供软件服务。根据第 3 节的协议可以看出，如果协议的整个过程，所有参与方都是按照协议规定的方式执行，则上述 2 个目标必然能够达到。然而根据第 2.3 节的分析，在开放的云环境下 Client 和 CS 都可能是恶意的，信息在网络传输时可能会遭到攻击者 Attacker 的重放篡改等攻击，这些情况下依然要求协议可以有效地达到设计目标，这需要保证协议能提供足够的安全性，下面分别考虑不同对象的攻击行为并进行分析。

恶意的 CS 可能不按 Client 规定的权限为消费者提供软件服务从而获得额外的利益，也可能试图攻击软件服务修改软件所有权。这 2 种情况下，由于软件的所有权和权限说明同时存在于软件水印和 CS 端的软件信息表中，恶意 CS 要想达到目标，仅修改软件信息表无法保证信息的一致，必然还要能够篡改软件水印的内容。

定理 1　恶意的 CS 不存在有效的攻击行为篡改同 Client 协商通过并嵌入到软件服务中的软件水印。

证明　嵌入的整个过程中最终确定什么形式

的软件水印嵌入软件服务中的权力在于 Client，Client 在嵌入之前所做的审查保证了最终嵌入的软件水印不可能在软件上传到 CS 前被 CS 篡改。而软件水印嵌入和提取中的安全性是由可信的 WCS 保证，本文对软件水印的不可破坏性假设保证了软件水印嵌入后的安全性，因此 Client 提供的软件在上传到 CS 云中后依然不会被篡改。

Client 拥有的嵌入提取软件水印的密钥能够决定软件水印的内容，因此 Client 可以通过修改嵌入的软件水印内容或提供虚假的水印提取密钥，从而构造恶意事实指责 CS 不按协商的权限提供服务来破坏 CS 的声誉。

定理 2 恶意的 Client 不存在有效的攻击行为改变与 CS 协商确定并嵌入到软件服务中的软件水印达到说明 CS 不可信任的目的。

证明 由于软件水印中的相关信息内容需由 CS 签名后才有效力，因此 Client 通过虚假密钥提取的或自己构造的软件水印没有作用。对于 CS 传给 Client 的软件水印信息$\{\{ID_C, SID，SL\}_{SK_C}, ID_S\}_{SK_S}$，由于 Client 没有 CS 的私钥，所以 Client 无法篡改软件水印中的内容。此时 Client 可能重用以前和 CS 协商好的软件水印，若之前的软件水印中的权限和当前软件服务指定的权限并不相同，将此软件水印嵌入当前软件并上传，则可作为指责 CS 没按照指定的权限提供服务的依据。然而在 Judge 进行仲裁时，CS 根据提取的软件标识 *SID* 查找该 Client 的软件信息表，由于 *SID* 在 CS 端的软件信息表中只对应唯一的一个软件，因此必然可以发现已存在该 Client 的另外一个软件使用同样的软件水印，而 CS 无法篡改软件水印，因此责任方必然是该 Client，Client 不能证明 CS 不可信任。

定理 3 原始软件、含水印的软件、软件水印、水印密钥等秘密数据在通信过程中是安全的。

证明 原始软件、含水印的软件、软件水印、水印密钥等秘密数据在网络中传输时，都是通过会话密钥加密保护的，因此秘密数据的通信安全取决于会话密钥分配的正确性，本协议中使用的密钥分配方法是 ISO/IEC 11770-3 密钥分配协议[11]的简化，文献[12]证明了该密钥分配方法的正确性。

定理 1~定理 3 说明协议执行过程中恶意的行为不会威胁协议的安全。因此只要保证密钥的安全性，软件水印在嵌入前、嵌入中和嵌入后都是安全的，从而整个安全协议在一定程度上是安全可靠的。

5 结束语

在一切皆服务的云计算时代，云平台中为外界提供计算能力的软件服务尚缺乏有效的保护。本文考虑云服务特殊需求，结合软件水印技术和密码机制，给出了云平台下软件服务保护的安全协议；文中建立了一个简化的安全模型，分析了模型中的相关实体，给出各个实体之间的交互过程，在此基础上分析给出协议的安全性。利用给出的协议，能够解决云端软件服务的侵权问题。由于本文的协议还比较简单，一些未知的攻击都有待发现。进一步地完善将是下一步的一个研究方向。

参考文献：

[1] YU Z, WANG C, THOMBORSON C. A novel watermarking method for software protection in the cloud[J]. Software Practice and Experience, 2012, 42(4):409-430.

[2] HWANG K, LI D. Trusted cloud computing with secure resources and data coloring[J]. IEEE Internet Computing, 2010, 14(5):14-22.

[3] STERN J, HACHEZ G., KOEUNE F. Robust object watermarking: application to code[A]. Proceedings of the Third International Workshop on Information Hiding[C]. Dresden, Germany, 1999. 368-378.

[4] COLLBERG C, CATER E. Dynamic path-based software watermarking[A]. Proceedings of the ACM Conference on Programming Language Design and Implementation[C]. New York, USA, 2004. 107-118.

[5] COLLBERG C. Software watermarking: models and dynamic embedding[A]. Proceedings of Symposium on Principles of Programming Languages[C]. New York, USA, 1999.311-324.

[6] COLLBERG C, THOMBORSON C. Watermarking, tamper-proofing, and obfuscation-tools for software protection[J]. IEEE Transactions on Software Engineering, 2002, 28(8):735-746.

[7] KAMELA I, ALBLUWIB Q. A robust software watermarking for copyright protection[J]. Computers & Security, 2009, 28(6):395-409.

[8] COLLBERG C, HUNTWORK A, CATER E. Software watermarks: Implementation, analysis, and attacks[J]. Information and Software Technology, 2009, 51(1):56-67.

[9] 陈晓苏,刘立刚,卢正鼎.网络环境下数字图像版权保护安全协议的设计与分析[J].计算机学报,2006, 29(9):1722-1727.
CHEN X S, LIU L G, LU Z D. Design and analysis of digital image copyright protection security protocol in internet environment[J]. Chinese Journal of Computers, 2006, 29(9):1722-1727.

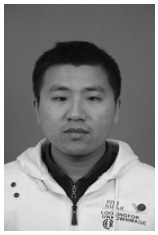
[10] 胡军全, 黄继武, 张龙军. 结合数字签名与数字水印的多媒体认证系统[J].软件学报, 2003, 14(6):1157-1162.
HU J Q, HUANG J W, ZHANG L J. A multimedia authentication

system combining digital signature and digital watermarking[J]. Journal of Software, 2003, 14(6):1157-1162.

[11] ISO. Information Technoly-Security Techniques-Key Management-part 3: Mechanisms Using Asymmetric Techniques ISO/IEC 11770-3[S]. International Standard, 1999.

[12] SHOUP V. On formal models for secure key exchange[EB/OL]. http://www.shoup.net/papers/skey.pdf, 2012.

作者简介：



许金超（1982-），男，山东临沂人，同济大学博士生，主要研究方向为软件保护、信息安全。

曾国荪（1964-），男，江西吉安人，同济大学教授，主要研究方向为可信软件、信息安全。

王伟（1979-），男，湖北武汉人，同济大学讲师，主要研究方向为内容安全、并行计算。

（上接第 175 页）

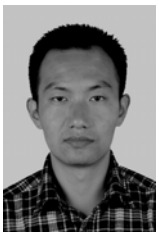
[7] FONTAN F P. Channel modeling for land mobile satellite services[A]. Proceedings of the Fourth European Conference on Antennas and Propagation (EuCAP)[C]. Spain, 2010.1-5.

[8] LOO C. A statistical model for a land mobile satellite link[J]. IEEE Transactions on Vehicular Technology, 1985, 34(3):122-127.

作者简介：

韦娟（1973-），女，陕西渭南人，博士，西安电子科技大学副教授，主要研究方向为卫星通信与移动通信。

刘达（1984-），男，辽宁锦州人，西安电子科技大学硕士生，主要研究方向为卫星通信。



田刚旗（1987-），男，陕西武功人，西安电子科技大学硕士生，主要研究方向为卫星通信。