



作者：吴圣健 | 新加坡法律出版社特邀技术顾问

“

摘要

游戏换皮案件中的技术取证，不能只停留在截图、公证和人工比对层面。

对于网络游戏、App、小程序和网站等复杂互联网产品而言，用户最终看到的画面，往往由客户端、服务器、数据库、配置文件、资源包、热更新内容、账号状态和时间窗口共同形成。

本文以《太极熊猫》诉《花干骨》案为观察样本【案号：（2018）苏民终1054号】，讨论游戏侵权取证中的两个基础问题：

1. 历史版本如何证明

2. 服务端内容如何归因

只有先解决“被比对的版本是否真实”“争议画面到底从哪里来”这两个问题，后续界面、玩法、数值和开发文档的比对才具有稳定基础。

目录

一、案件事实与技术争议焦点

1. 案件基本情况
2. 本案中的核心技术问题
3. 上篇讨论范围

二、游戏换皮案件中的技术取证框架

1. 固定版本：先确定取证对象是什么
2. 还原运行：证明版本能够可信运行
3. 结构化比对：把相似感变成可复核材料
4. 解释证据：区分通用设计与复制痕迹

三、技术动作一：历史版本如何证明

1. 本案中的技术动作
2. 双方争议
3. 技术升级路径：版本证据保险箱
4. 哈希、时间戳与容器镜像的作用

四、技术动作二：服务端内容如何归因

1. 本案中的技术动作
2. 争议实质
3. 服务端如何影响画面内容
4. 技术升级路径：建立证据归因闭环

五、上篇小结

一、案件事实与技术争议焦点

1. 案件基本情况

《太极熊猫》诉《花千骨》案，是两款手机游戏之间的著作权侵权纠纷。

原告主张，被控游戏在以下方面与权利游戏构成实质性相似，并请求停止侵权、下架游戏、赔偿经济损失：

- 游戏结构
- 玩法规则
- 数值内容
- 投放节奏
- 软件文档

被告则主要从以下角度进行抗辩：

- 权利版本证据的客观性
- 游戏玩法的通用性
- 两款游戏在美术和题材上的差异
- 赔偿数额与侵权内容之间的因果关系

因此，本案并不是单纯比较两张游戏截图是否相似，而是围绕复杂网络游戏产品的**版本、服务器、界面、玩法、数值和文档**展开争议。

2. 本案中的核心技术问题

从技术视角看，本案集中呈现了游戏侵权取证中的五个问题：

技术问题	对应证明目的
历史版本如何证明	被比对的版本是否真实、可靠
服务端内容如何归因	争议画面到底从哪里来
界面相似如何结构化比对	相似点是否可复核、可拆解
动态玩法如何转化为流程证据	玩法流程是否存在对应关系
底层数据和开发文档如何寻找复制指纹	是否存在更强指向性的复制痕迹

这五个问题构成一条逐步推进的证据链：

“

先证明被比对的版本可靠，再说明画面内容从哪里来；
在此基础上，才进一步分析界面哪里相似、玩法流程是否对应，以及底层数据和开发文档中是否存在更有指向性的复制痕迹。

游戏换皮案件中的技术证据链

从技术视角看，游戏侵权取证不是单张截图，而是多层证据对象的组合。



3. 上篇讨论范围

本文分为上下两篇。

上篇重点讨论前两个技术动作：

1. **历史版本证明**
2. **服务端内容归因**

这两个问题解决的是证据链的前端基础，即：

- 被比对对象是否可靠
- 争议内容来源是否清楚

下篇将继续讨论后三个技术动作：

1. 界面结构化比对
2. 动态玩法流程证据
3. 底层数据和开发文档中的复制指纹

二、游戏换皮案件中的技术取证框架

1. 固定版本：先确定取证对象是什么

游戏侵权取证的第一步，是固定版本。

这里的“版本”不只是应用商店页面上的版本号，也包括：

- 客户端安装包
- 服务端部署文件
- 数据库结构
- 配置表
- 资源包
- 热更新包
- 脚本
- 账号状态
- 上线时间

如果版本对象无法明确，后续所有相似性比对都会失去稳定基础。

因为被告可以质疑：

“

原告拿来比对的不是当年真实存在的版本，而是后来重新恢复、拼接或修改后的运行状态。

2. 还原运行：证明版本能够可信运行

网络游戏往往不能只靠客户端独立运行。

很多内容都可能依赖服务端配置、数据库记录或账号状态，例如：

- 界面内容
- 商城信息
- 活动规则
- 道具数值
- 功能开启状态

因此，还原运行的目标，不只是证明游戏“能跑起来”，而是证明：

- 它在什么环境下跑起来
- 使用了哪些服务端文件
- 数据库如何导入
- 配置是否被修改
- 日志和数据库是否能够相互印证

只有运行过程**可审计、可重复、可解释**，历史版本证据才更有说服力。

3. 结构化比对：把相似感变成可复核材料

在版本和运行环境被初步固定之后，才进入结构化比对阶段。

游戏换皮案件中的相似，可能存在于多个层面：

- 界面布局
- 按钮位置
- 玩法流程
- 数值体系
- 资源文件
- 文案内容
- 开发文档

单纯说“看起来像”并不足够。

更稳妥的做法，是将相似内容拆解成可以复核的材料：

比对象	结构化方式
界面	区域、按钮、文字、坐标
玩法	入口、条件、消耗、奖励
数值	表格、曲线、阈值
文档	图片、元数据、文件名

4. 解释证据：区分通用设计与复制痕迹

并非所有相似都能指向侵权。

游戏行业中存在大量通用玩法、功能性设计和有限表达，例如：

- 商城
- 排行榜
- 副本
- 签到
- 装备强化

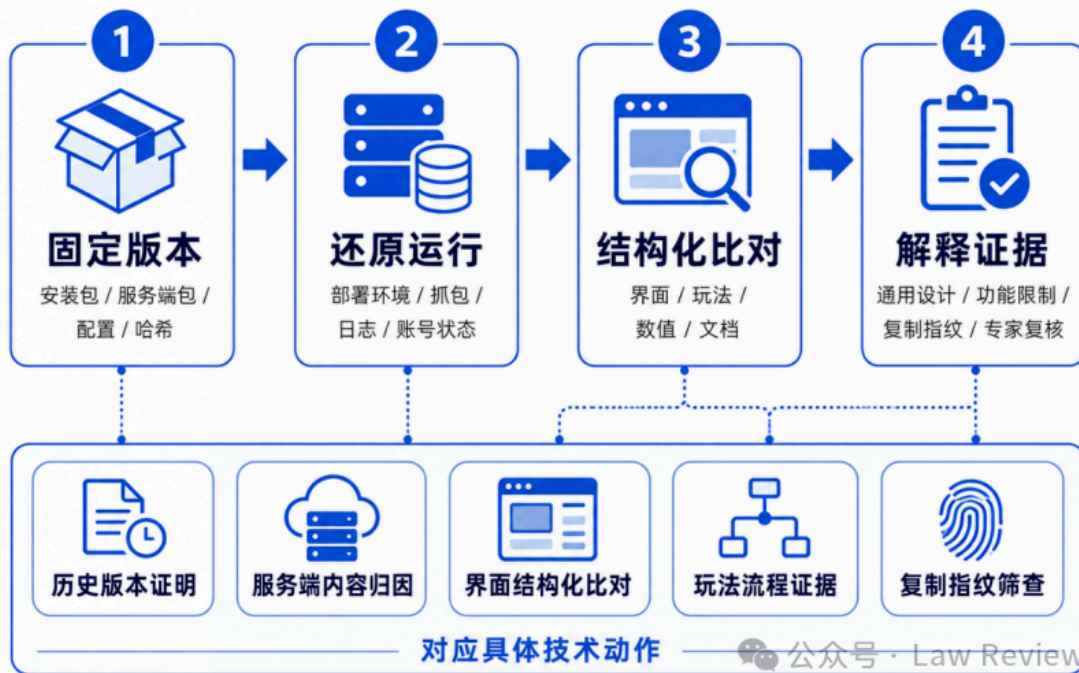
真正需要分析的是：

- 相似是否具有非必要性
- 相似是否跨多个系统重复出现
- 相似是否能与数值、资源和开发文档中的复制痕迹相互印证

技术取证的价值，不是替代法律判断，而是让法律判断建立在更清楚、更完整的技术事实基础上。

游戏侵权取证的四个阶段

从版本固定到证据解释，形成完整的技术取证路径。



三、技术动作一：历史版本如何证明

1. 本案中的技术动作

本案中，原告为了证明权利游戏的历史版本，首先通过公证方式固定了历史客户端的下载、安装和试玩过程。

由于该游戏需要客户端和服务端共同运行，原告又通过临时搭建服务器的方式恢复历史运行环境。

为了说明服务端部署文件的来源，原告提交了经过公证固定的版本管理系统记录，并从历史版本记录中提取用于部署服务器的压缩资源包。

之后，在法院监督下，原告工作人员演示了服务器部署过程，包括：

1. 创建云服务器
2. 上传历史服务端资源
3. 导入数据库
4. 修改配置
5. 启动服务端
6. 使用历史客户端连接服务器进行测试

2. 双方争议

原告证明逻辑是：

“

历史客户端来自公开下载平台，服务端资源来自历史版本管理记录，二者能够匹配运行，并且运行后能够生成数据库记录和服务端日志，因此可以证明权利游戏历史版本的客观状态。

被告的攻击点则集中在“可修改性”上。

被告认为：

- 版本管理系统记录可以被修改
- 服务器端数据可以被修改
- 现场部署过程中还导入了配置文件、数据库脚本和第三方工具
- 每一个环节都可能影响最终运行画面

因此，仅凭临时恢复运行，不能当然证明当年真实线上版本。

3. 技术升级路径：版本证据保险箱

如果今天为游戏公司设计日常证据体系，不应等到诉讼发生后再临时恢复历史版本，而应在每次正式发布时建立：

“

版本证据保险箱

每个正式发布版本，都可以形成一份发布清单，记录以下内容：

证据类型	具体内容
客户端文件	安装包、版本号、签名证书、哈希值
服务端文件	服务端镜像、部署脚本、配置文件
数据库材料	数据库结构、初始化数据、配置表
资源文件	资源包、热更新包、素材文件
构建记录	构建日志、代码提交编号、发布审批信息
运行材料	关键玩法录屏、测试账号状态、运行日志

这份发布清单的作用，是证明某个版本在某个时间点包含哪些文件、配置和运行环境。

4. 哈希、时间戳与容器镜像的作用

对关键文件，可以分别计算哈希值，例如 [SHA-256](#)。

这些关键文件包括：

- 客户端安装包
- 服务端镜像
- 配置文件
- 资源包
- 数据库结构
- 热更新包

哈希可以理解为文件的“数字指纹”。只要文件内容发生变化，哈希值通常也会变化。

企业还可以将关键哈希值通过以下方式进行锚定：

- 可信时间戳
- 公证存证
- 电子数据存证平台
- 区块链存证

但需要注意的是：

“

区块链或时间戳并不能证明文件内容本身一定真实。
它证明的是某个哈希值在某个时间点已经存在。

真正的证明力，来自以下要素共同形成的闭环：

1. 发布流程
 2. 文件哈希
 3. 时间固定
 4. 运行日志
 5. 可复验部署
-

发布版本的完整性固定与未来复验路径

与其在诉讼中临时恢复，不如在发布时就形成版本证据保险箱。



四、技术动作二：服务端内容如何归因

1. 本案中的技术动作

本案中，被告提交了技术鉴定意见，证明同一客户端安装运行后，界面内容可以通过更改服务器端相关内容发生变化。

这个结论在游戏案件中非常重要。

它说明，客户端画面可能不是单一文件的结果，而是多个因素共同作用的结果：

- 客户端
- 服务器
- 数据库
- 账号状态
- 运营配置

2. 争议实质

原告需要证明：

“

其临时恢复的服务端资源来自历史版本记录，客户端与服务端能够匹配运行，因此运行画面可以作为权利版本证据。

被告则强调：

“

既然服务端可以改变客户端显示内容，就不能排除原告通过修改服务端配置、数据库数据或脚本，使历史客户端显示出对原告有利的内容。

也就是说，争议的核心不是抽象讨论“服务器能不能改”，而是要具体判断：

“

争议内容到底来自哪里？

3. 服务端如何影响画面内容

在网络游戏中，客户端截图往往只是最终呈现结果。

画面背后，可能存在多种技术来源：

画面内容	可能来源
商城价格	数据库配置
活动规则	服务器接口
道具名称	资源包或服务端返回
功能入口是否显示	账号等级、权限或时间窗口
奖励结算	服务器逻辑
部分资源	热更新下发

因此，技术分析不能只停留在截图层面，而要进一步说明这个结果是如何生成的：

- 是客户端本地写死的？
- 还是服务端下发的？
- 是固定版本内容？
- 还是运营活动配置？
- 是所有用户可见？
- 还是特定账号、等级或时间窗口下才出现？

4. 技术升级路径：建立证据归因闭环

在合法取证和授权分析范围内，抓包、日志、数据库快照等技术手段的价值，不在于“攻击服务器”，而在于建立证据归因。

更完整的取证流程，应当尽量固定以下内容：

(1) 客户端侧证据

- 客户端安装包
- 版本号
- 签名证书
- 哈希值
- 应用商店页面
- 下载来源
- 更新时间
- 渠道信息

(2) 运行环境证据

- 干净设备或模拟器环境
- 系统版本
- 网络环境
- 测试账号状态
- 操作时间线
- 客户端录屏

(3) 网络与接口证据

- DNS 记录
- TLS 证书
- 资源更新地址
- 关键接口请求
- 关键接口响应内容

(4) 服务端与数据库证据

- 服务端日志
- 数据库快照
- 配置文件
- 资源更新记录
- 后台操作记录

最终目标，是建立以下对应关系：

“

操作 → 请求 → 响应 → 画面 → 日志 → 数据库

只有形成这样的归因闭环，才能更清楚地说明：争议画面究竟是如何生成的、由什么文件或数据控制、是否属于当时真实存在的版本状态。

手游运行内容的多源归因示意

同一游戏画面，可能由客户端、服务端和配置共同生成。



核心问题：争议内容到底来自哪里？

公众号 · Law Review

五、上篇小结

游戏换皮案件不能从“像不像”直接开始。

在比较界面、玩法、数值和文档之前，首先要解决的是：

1. 被比对的历史版本是否真实
2. 运行环境是否可信
3. 争议画面到底来自客户端、服务器、数据库、配置文件还是热更新资源

其中：

“

历史版本证明解决的是：

当年的游戏到底是什么样。

“

服务端内容归因解决的是：
画面内容到底从哪里来。

只有这些问题被拆清楚，后续的界面结构化比对、玩法流程比对、数值分析和开发文档比对，才有稳定基础。



「

作者简介



吴圣健

新加坡法律出版社研究院

特邀技术顾问

吴圣健先生毕业于华东师范大学，先后取得软件工程本科学位及电子信息硕士学位。其长期从事人工智能、计算机视觉、图像处理、网络数据分析及软件系统研发工作，具有十余年技术研发与项目落地经验。技术方向涵盖目标检测、多目标跟踪、人脸识别、ReID、图像处理、模型部署推理、网络数据分析及后端系统开发等领域。曾以第一作者身份在AI领域的国际顶级会议（CCF-A类）上发表多篇论文，并作为第一完成人获得多项国家发明专利授权。其产业实践涉及无人智能零售图像识别、金融反欺诈、伪造检测、网络流量预警及后端系统建设等方向。



关于新加坡法律出版社研究院

Singapore-Law-Press Research Institute

新加坡法律出版社研究院系依托新加坡法律出版社设立的研究与内容平台，聚焦知识产权、互联网争议、电子证据与法律技术等方向，持续开展研究观察、专题交流与内容出版，致力于输出兼具专业性与实践价值的法律内容。



官网：www.chinalegal.com.sg/slp



邮箱：sglawpress@chinalegal.com.sg



地址：7500A Beach Road, #08-305,
The Plaza, Singapore 199591



知识产权

| 互联网争议

| 电子证据

| 法律技术



公众号 · Law Review

