版权护卫

伴随着互联网和图像技术的快速发展，人们的生活与工作变得越来越方便，交流和娱乐也越来越多样化，但这其中同时也带来了不少安全隐患。回想过去十年的中国互联网暴力野蛮的发展，数字图像的易拷贝和易传播特性，带来了大量的侵权和盗版问题，给著作人的利益带来了严重的损害，对大量的数字行业带来的深刻并且严重的破坏。幸运的是，现在版权问题已经获得越来越多的人重视，社会和国家也在促进版权保护的发展，各个公司更是视版权为财富。即便如此，由于传统的可见水印嵌入方案的版权保护能力已显乏力，不能跟上时代对版权的号召，存在着大量的安全漏洞，可见水印对版权的保护形同虚设。本平台将鲁棒的不可见水印技术和互联网技术进行结合，为此类侵权盗版问题的解决提供了新的视角。虽然不可见水印技术目前仍然存在不足，没有大量的商业化，也不能彻底解决存在的安全问题，但是其提供了一个较为新颖的解决方案，并可以提供比可见水印更强的版权保护。

一、简介

本文产品的名称叫做“版权护卫”，以鲁棒的不可见水印技术为核心，构建了一套精简的版权保护平台，为用户提供最基本的版权保护能力的同时，也能保障图像的观赏性，甚至在图片在一定程度破坏的情况下，同样能够提取出水印。图 1 是本产品的首页。

【图 1】

产品主要提供了“可见水印”、“不可见水印”和“图片注册”的三大功能用以进行版权保护，这三种功能相互独立，每种功能都有其各自的应用场景。1).可见水印。该功能就是现在最场景的水印嵌入方案，嵌入一个肉眼可见的水印，水印内容是一串文本。2).不可见水印。该功能的水印内容同意是一串文本，但是嵌入后不可见，可以通过密钥将嵌入的水印文本提取出来。3).图片注册。该功能将原始图片进行注册，并返回一个注册图片，注册图片和原始图片之间的区别肉眼不可见。用户通过注册图片，可以追溯到该注册图片的注册人，并和注册人进行联系。

二、需求分析

2.1 现状分析

随着数字技术和通信技术的发展，图像成为目前互联网中最受欢迎的多媒体形式之一。2010 年以后，智能手机的大面积普及，也让越来越多的人通过手机拍照来记录生活点滴，再加上手机中各类滤镜的使用与后期手段，手机已经可以拍出不逊色于专业相机的效果，因此人们热于通过移动端的操作来进行图片的制作和分享。

发达的数字图像技术和便捷的分享途径，在给人们带来丰富乐趣和充实的图像内容的同时，也隐藏着多媒体安全问题。在互联网平台上发布自己创作的图片时，容易遭它人盗用，这将会极大侵害图片创作者的个人权益。现阶段，作者在发布图像作品时，为了达到最基本的版权保护功能，通常会往图片中加入可见的水印，以起到保护的作用，水印中通常包含作者或图像本身的相关信息，如图 2 中的照片加入了可见的水印信息“大阪”。

图 2 普通水印

可见水印的引入，改善了图片侵权的现状，也在一定程度上促进了大众的版权保护意识，但是可见水印也存在诸多的缺点亟待解决：

1).安全性过低。数字图像修复技术可以对破损图片进行一定程度的修复，水印本身就是一种可见的图片破损，因此水印极容易被包括数字图像修复技术在内的相关技术进行去除，使得不法分子还原出高质量的原图，进而满足自己的利益。对于一些放置于边边角角的水印而言，直接可以通过截图技术就能去掉水印。

2).对视觉的影响。为了保证安全性，会采用将图像放置图片中央，甚至布满图片的形式，如图 3 所示。这样子的可见水印虽然提高了安全性，不易被截去，但是也对视觉效果带来了较大的破坏，影响了图片的观赏性，不利于图像放在互联网平台上的初衷“传播”。

3).真实性问题。为了提升安全性并且降低对视觉的影响，现在越来越多图像创作者喜欢在图像中引入花式的水印，这些水印看起来较为真实，不容易被攻击者理解为水印，进而被去除，如图 4 所示，碗上的“大阪”标识实际上为水印。虽然这样可以确实能起到效果，但是若被理解为水印，同意也容易被去除，并且容易让观赏者误会。

图 3 强可见水印

图 4 花式可见水印

为了解决可见水印存在的诸多问题，学术界在早期就已经提出了包含 LSB 在内的不可见水印技术，然而目前在不可见水印方面的小程序极少，在小程序搜索栏中搜索“水印”关键词，只能查询出寥寥几款可见水印小程序。不仅仅是小程序，在 App、Web 和桌面应用中，都极少见到不可见水印平台的应用程序。虽然由于技术不够成熟，此类技术持续沉浸于实验室多年，但随着近几年鲁棒不可见水印技术突飞猛进的发展，不可见水印越来越丰满和成熟，已经具备一定的商业使用价值。图 5 中分别显示了原始图像、水印图像和含水印的图像，很明显，从含水印图像中完全无法看出关于水印的信息。

(a) 原始图像　　(b) 水印图像　　(c) 含水印图像

图 5 原始图像、水印图像与含水印图像

2.2 应用场景

2.2.1 版权保护

图片创作者发布不含水印的图片，虽然可以促进图片在互联网中的传播，但是却极易被盗用和篡改，因此如今很多创作者都嵌入可见水印并发布图片，水印中包含了创作者的相关信息，以便意图使用图片的人可以和图片创作者进行联系，但可见水印对图片带来的视觉影响，以及其较差的安全性，都使得其保护形同虚设。不可见水印技术可以最大程度降低对视觉的影响，其具有的鲁棒性也可以使得图片在一定程度破坏的情况下仍然能够提取水印，并且不可见水印本身也隐藏了图片“含水印”这一事实，避免引起水印攻击者的注意，提升安全性。本产品提供了“不可见水印”功能，提升图片安全性，并且促进图片在互联网中的传播。产品的“不可见水印”支持用户上传原始图片，并输入不可见水印的文本内容，并选填密钥，小程序将会把必要数据上传到服务器，并由服务器嵌入不可见水印，并将含水印图像返回给小程序。意图使用图片的他人，可以在本产品的中输入相关密钥提取出不可见水印，。需要注意的是，在嵌入不可见水印时，图片创作人若是输入了密钥，这一行为意味着图片制作人禁止网络上的他人使用自己的图片，因为除了图片创作者，没有人知道密钥，也就无法提取作者的相关信息并和作者取得联系。图片创作人若是没有输入密钥，意味着每个人都可以提取出不可见水印的信息，水印中往往会有创作人留下的相关信息，意图使用图片的他人

便可以和图片制作人取得联系。

我们在进行自己的创意设计时，通常会在网上查阅相关的图片，并引用网上查找到的图片，这些图片往往没有图片创作者的信息，若是在引用时稍不注意，极有可能导致侵权行为。另外一个方面，网络上存在着大量的虚假信息，有人为了谋取一己私利谎称是图片的作者，并且直接采用可见水印或是不可见水印的功能，将会暴露作者的信息，而很多作者为了自己的隐私安全，不愿意对外公布这些信息。通过本产品提供的“图片注册”功能，可以在一定程度上抑制上述问题。图片创作者需要在平台上进行图片的注册，平台将会返回一个注册图片，该图片用于创作者进行发布，普通用户在网上查找到图片时，若该图片是平台的注册图片，则用户可以通过平台查询到图片注册人。很明显，通过这样的方式也可以辨识某人是否为图片作者，也可以避免作者隐私信息遭到泄露。

2.2.2 盗版追踪

数字图像的制作者除了可以嵌入鲁棒的版权水印以保护自身的版权外，还可以嵌入数字指纹以实现盗版追踪。当用户获得作者的认可时，作者将会拷贝图像给该用户，并且该数字图像中包含了唯一标识的水印数据，这样的水印就被称为数字水印。当作者在网上发现自己的数字图像作品被盗版传播时，即可获取作品中的数字水印，以获知是哪位用户进行盗版传播，并绳之以法。产品可以通过“不可见水印”以达到盗版追踪的目的。将数字指纹通过本产品进行不可见水印嵌入，发现盗版后，在本产品中提取数字指纹便可获悉盗版源头。

2.3 目标用户

四、功能详解

4.1 不可见水印

该功能可以往图像中嵌入不可肉眼察觉的水印，需要在选择了原始图片后，输入图片的标题(选填)，不可见水印文字内容(必填)，提取水印的密钥(必填)。由于不可见水印涉及到较为复杂的计算，通过手机来进行运算速度太慢因此当前版本会将图像发送到服务器进行不可见水印内容的嵌入，嵌入完成后会将含密水印返回。图 6 描述了不可见水印嵌入的相关页面。

(a).输入必要信息 (b).等待服务器嵌入完成 (c).服务器返回的图像

【图 6 嵌入不可见水印】

存在水印的嵌入操作，就会有水印的提取操作，需要在选择了图像后，输入水印密钥。图 7 描述了不可见水印提取的相关页面。

【图 7 提取不可见水印】

4.2 图片注册和追溯

图片注册和追溯，是一个闭环的版权保护功能，当图片创作者在本平台上进行了照片的注册后，平台将会绑定图片和作者，并返回一个注册图片(该图片含水印)，作者可以将服务器返回的注册图片用于任何活动的使用。在图 8 给出了图片注册的相关页面。

【图 8 图片注册】

在平台上，用户对注册图片进行溯源，能够找到该图片所对应的注册者，并和该注册者取得联系。在图 9 中给出了图片追溯的相关页面。由于微信是一个较为私人敏感的平台，因此为了避免注册者的隐私遭到泄露，因此追溯成功是不会显示关于注册者任何相关信息的。

【图 9 图片追溯】

4.3 留言收发

对于在平台上注册的图片，用户能够在平台上进行图片追溯，以找到图片的注册者，当找到注册者以后，用户可以向注册者发送消息。为了避免消息发送者的隐私泄露，本平台不会发送关于发送者的任何隐私，因此需要用户在消息中写入自己的愿意透露的别名和联系方

式等信息。图 10 中给出了发送留言的相关页面。

【图 10 发送留言】

在留言发送成功后，注册者会接收到未读的留言，注册根据留言发送者中透露的信息，自行选择是否和该用户进行联系。在图 11 中显示了留言查阅的相关页面。当前版本还未引入黑名单功能，在后续会加入。

【图 11 留言查阅】

4.4 可见水印

本产品虽然是以不可见水印技术核心的，但是为了更为平稳的往不可见水印技术进行过度，并且提供更广全方面的版权保护功能，因此同样也提供了可见水印的功能。可见水印的功能较为简单，并不会涉及到复杂的数学计算，因此在小程序端进行完成，并支持离线功能。在图 12 中给出了可见水印的相关页面。

【图 12 可见水印】

4.5 历史记录

考虑到用户进行了水印嵌入操作以后，由于某些原因导致了含水印图像的丢失，因此在服务器上保存了含水印的图像，用户可以通过历史记录查看所有操作的含水印图像，需要注意的是服务器中没有保存用户提供的原图，历史记录中也不会提供原图的记录，并且也不会包含可见水印图像的记录。在图 13 中给出了历史记录的相关页面。

【图 13 历史记录】

五、技术开发方案

5.1 整体架构

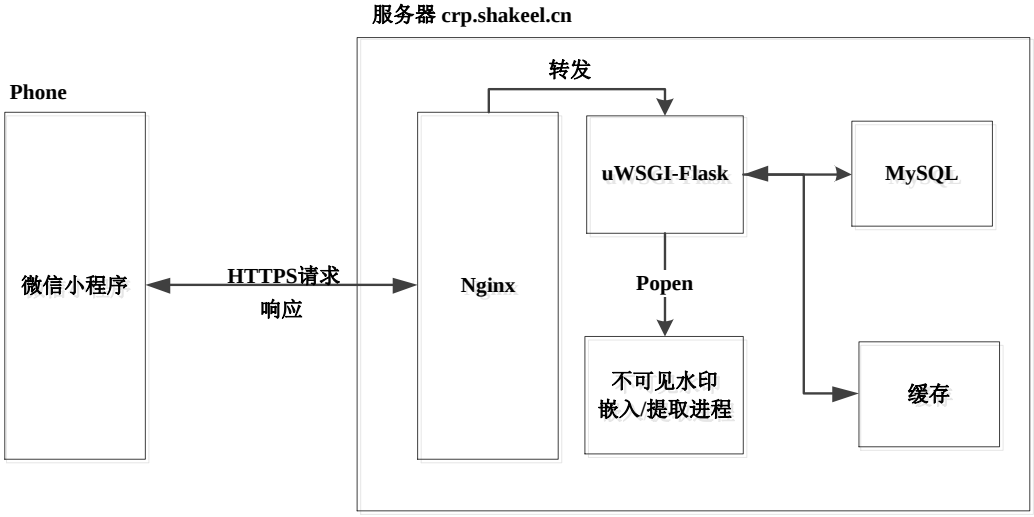

图 1 系统整体框图

5.1.1 服务器端

服务器后台将会提供 HTTPS 接口，以便小程序获得“不可见水印”和“图片注册”的相关服务。后台采用 Python-Flask 进行开发，通过 uWSGI 运行，为了更好的支持 SSL 功能，搭建了 Nginx 服务器接收 HTTPS 请求，并将请求转发给 uWSGI 以处理。

Python 后台在产品中的主要作用是接收和返回网络 IO 任务以及相关数据的出库入库，而对于不可见水印的嵌入和提取过程，这是一个 CPU 密集的任务，考虑到 C++在 CPU 密集型任务上处理的优势，因此对于不可见水印的嵌入和提取请求，Python 将会转交给 C++进程进行处理。目前转交给 C++处理的形式较为简单和粗暴，是通过 Python 调用 C++进程的

形式进行的，这样的形式缺点主要在于对于高并发情况下，大量的进程创建和销毁，这将会成为性能瓶颈，因此后期考虑通过 C++进程进行端口监听，Python 通过 Protobuf 将任务序列化，并交给 C++进程，C++进程接收到 Python 的任务后，提交任务给线程池来处理不可见水印提取和嵌入的任务，如图所示。改进的机制不但避免了大量的进程创建销毁所需要花费的代价，并且限制了进程的无限制创建，通过线程池机制可以在高并发环境下削峰填谷。

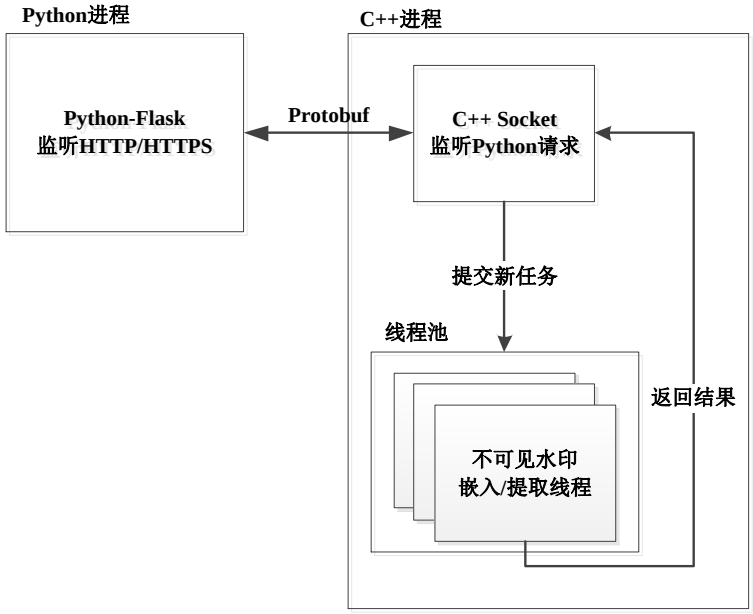

图 2 改进的 Python 转发 C++机制

5.1.2 小程序端

5.2 解决方案

5.2.1 小程序端

5.2.2 服务器端

1). 会话机制

Session(会话)是 Web 架构中常见的机制，通过在 cookie 中保存 sessionId 的形式来实现，用来在服务器上保存当前会话的重要数据，例如登录信息和配置信息等。在微信小程序中，并没有实现 Cookie 机制，进而无法支持 Web 的 Session 机制，因此需要我们自己设计 Session 机制。当小程序每次登录的时候，都会发送一个会话建立请求到服务器，该请求中包含了微信登录凭证 code 和设备唯一码 did，便于服务器获取建立会话并获取该会话所对应的微信 ID。服务器在会话建立的时候，会立即将微信 ID 放到会话的缓存中，以便获取后续每个请求所对应微信 ID。

同一个设备上的同一个微信用户，若已经和服务器建立了会话后，由于某些原因(例如 sessionId 丢失)将会再次尝试和服务器建立会话，服务器将会进行会话合并，即返回该服务器和该微信小程序已有会话的 sessionId。微信自身虽然可以避免同一个微信用户在不同的设备上进行登录，但由于本产品是基于 HTTPS 的，因此极易被模拟登入，会有同一个微信用户在不同的设备上登录本平台的风险，进而导致同一个会话服务于多个设备(会话合并造成的)，因此需要设备唯一码 did 进行标识。微信小程序本身无法获取 did，因此需要到服务器上获取 did。服务器的 did 接口每次调用都会返回一个不一样的 ID 号，小程序在第一次登入的时候将会缓存该 ID 号，作为自己的设备唯一码，以后每次会话建立请求都应该带上该设备唯一码。图 x 是服务器接收到会话建立请求后的处理流程图。

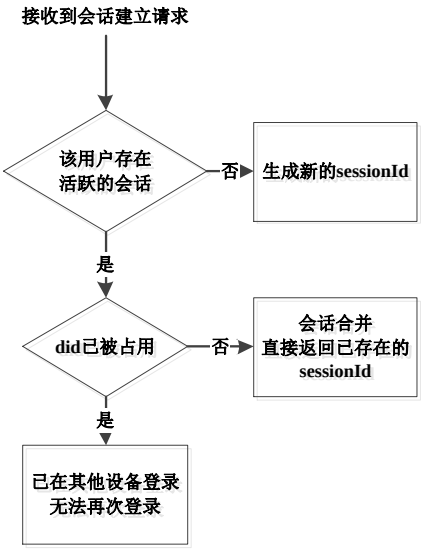

图 3 会话建立流程图

服务器有两种方式销毁接口，一种是服务器提供了会话销毁接口，在小程序退出的时候应该调用会话销毁接口。另一种时超时自动销毁，会话每次建立 1 小时候，服务器会自动销毁掉会话，会话合并将会重新刷新持续时间为 1 小时。持续时间是为了避免小程序端由于某些原因，导致会话销毁请求无法发送出去，进而占用服务器资源。小程序端应该持续检测用户的活动情况，当用户存在活跃点击的时候，小程序将会周期性发送会话建立请求，触发会话合并，进而方便刷新会话超时时间。图 x 是整个会话的生命周期时序图。

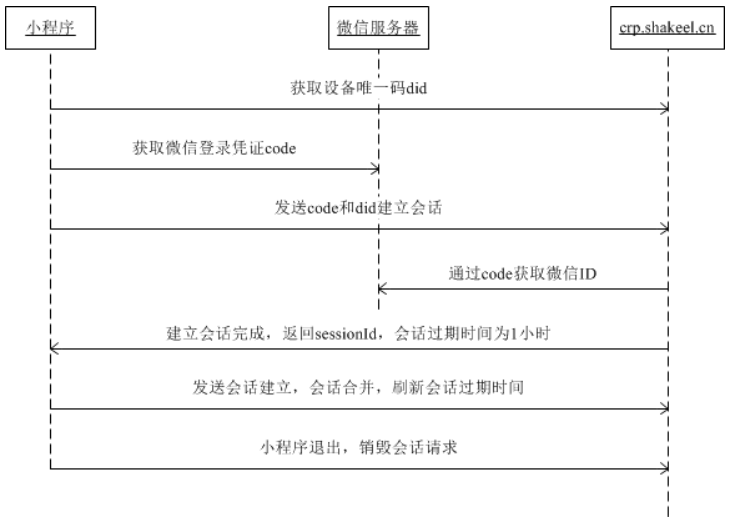

图 4 会话生命周期

2). 异常机制

3). 限流机制

为了避免 HTTPS 接口被机器模拟登入，造成服务器资源被滥用，导致服务器崩溃，因此需要对请求频次进行限制。在本平台中，限制每个访问 IP 每分钟最多访问 20 次，超过 20 次将会进制该 IP 的任何请求。

4). 图片追溯的隐私保护机制

图片注册相比于不可见水印的最大的优点是在保护版权的同时，避免图片创作者在图片中加入自己的信息，导致的隐私泄露。图片追溯可以让意图使用图片的人在平台上和作者取得联系，作者视情况自行回复，避免了隐私的泄露。因此在整个图片注册，图片追溯和消息

发送的流程中，都不应该泄露除了图片本身以外的任何有关作者的信息。

【图 x 图片追溯时序图】

在图 x 中给出整个流程时序图，首先图片注册就会先生成一个 imgid，将其作为水印信息嵌入到图片中返回给小程序，并将 imgid 和微信 ID 在数据库中绑定在一起。图片追溯时，后台将会提取出水印作为 imgid，并且当 imgid 在数据库中存在时，则认为图片经过了注册，能够追溯到原作者，这时将会返回小程序该图片的 imgid。用户在追溯成功后，可以发送消息，消息发送会带上追溯成功时返回的 imgid，服务器接收到了消息发送的请求，会将 imgid 所对应的微信用户查询出来，再将消息发送给该微信用户。很明显，整个追溯过程都不会泄漏关于图片注册的任何信息，都是由希望使用图片的用户发起的所有操作，并等待图片作者的回复，这样确保了图片作者的隐私不被泄露。

5). 变分辨率问题

5.2.3 核心算法

六、团队的组成与分工

1).陈智隆(队长)

负责鲁棒水印核心算法编写，包括了：1).测试微信发送图片对图片本身的 JPEG 压缩和变分辨率情况。2).查阅相关文献，进行鲁棒水印的 Matlab 仿真，并测试在微信发送后水印的提取情况。3).采用 OpenCV 库，将鲁棒水印算法从 Matlab 移植到 C++平台，并分别编译了 Windows 和 Linux 版本的动态链接库，以便 Python 调用。

2).李素静

负责小程序端的界面显示以及所有交互效果，包括了：1).了解并熟悉小程序端开发组件和工具。2).设计并开发了小程序的显示界面。3).完成人机交互效果，优化用户体验。

3).卢帅吉

负责系统架构的设计以及系统的开发与部署，包括了：1).通过腾讯云架设服务器运行环境，如域名备案，Nginx 搭建，Python-Flask 环境搭建，SSL 注册等。2).设计服务器的 HTTPS 接口。3).开发服务器 CRUD 业务代码。