Maskbook 图片隐写功能优化

开源软件供应链点亮计划——暑假 2020

学生：全丽瑾 atlaschuen@uchicago.edu
导师：刘怿斯 yisiliu@gmail.com

项目简介

Maskbook 是一个开源的浏览器隐私插件，帮助用户在 Facebook，Twitter 等社交平台自动加密帖子和聊天记录，防止被社交平台审查和滥用用户数据，仅用户本人和指定好友能够解密。

Maskbook 现在有两种 payload 的模式，第一种是纯文字的 payload，格式是 base64 encoded 的加密文本，例如 `https://www.google.co.in/%204/4.UN3m8PThWagRN/YaV4HBFVZhmzU-wuIzD1Yugjm7JAPg1BgX9UilZl5jlHp0YsRs2cRu+lNTisyp1J45zvXoSD1fzbsK8+2Do5itncakns29P3BzH8YJACmG64G6tNikZP5oEme+W/to9wv1C43aSTs9rY1F8mFCwimOsZvki1+FN6LuItO3cX/1.NdahzvHRaBdRMI5gQvT5tg_=.27WQHRwyEwW8QwhIR0oOcPU9fbR35j8vKumBMyupnQ==.y5402vqUQoQub+RKHofpxgmQtetSiRwumF4KXaL/l5rmOHHRJBTL9YcZOjz39WtJAgbdg+iYsJdq5gm3MwWAXQ==.Ao+TblKN40/JUXl4B1HsISVoJc5gSVBYCXiolhxa8D5Q.0%40`，由于各个平台字数限制，所以不得不用一些特殊的方法进行 Hack，例如在 Twitter 上将 payload 编码在了链接中，以突破 Twitter 的 140 字限制，如上例所示。

另一种是图片模式：对图片进行时域到频域的变换后，将密文嵌入到图片中。这种模式更具有隐蔽性，也有更加美观的效果。目前的实现是基于一维快速傅立叶变换（$1dfft$）的一种方法，但是由于算法的限制，效果不甚理想，一方面图片中会有一些明显带有信息的像素点，没有完全实现人类感官的不可察觉性；另一方面抗平台压缩能力不够稳定，没有办法在 Facebook 和 Twitter 中都可以抗平台压缩，导致信息缺失。

本项目的短期任务是开发一套鲁棒性更强，抗平台压缩能力更强，并且人体感官不可察觉性更高的一套图片隐写算法。长期任务是使这套算法兼容性更强，可以接受用户指定的所有格式的任何图片，拓展任务是进一步设计并实现一套音频或视频隐写算法；另一个目标是使隐写后的图片中的模式变得更加难以被算法识别 (algorithmically hard to detect)，更难以被平台所检测，更好地保护用户的隐私。

1 项目详细方案

针对图片隐写部分，目前的实现是首先将图片分隔为的 $N \times N$ 的方块，每个方块存储 1 比特的信息。对每个方块来讲，首先进行一维傅里叶变换，然后选取方块中的一个特定位置，通过修改它的奇偶性来分别表示 0 或 1。此外，为了抗平台压缩，还需要将此位置的值扩大一定的倍数，提高算法的鲁棒性。最后再对图片进行逆傅里叶变换，就完成了对信息的隐写。然而目前的方案在人体感官不可察觉性方面表现不理想，为了改进当前方案，需要浏览相关论文，学习前沿图片隐写算法，并进行测试与评估。

为了使 Maskbook 的用户体验更加良好，还需要扩大增加的图片格式。目前该项目支持的图片格式有 JPEG 和 PNG。而 Twitter 支持的上传图片格式有 GIF, JPEG, 和 PNG[1]，因此后期需要还根据 GIF 的图片格式，进行相应的算法迁移。此外，Maskbook 还应该根据用户上传的图片大小与想要隐写的数据，并结合用户想要上传的社交媒体网络的压缩特性，评估此次隐写是否可以成功。

为了进一步实现音频和视频的隐写，也需要查阅相关资料，浏览相关论文，进行评估。

2 项目开发时间计划

本次项目时间计划大体基于实际的工作日数分配工作量，并且特别根据中期报告的时间节点，将短期任务分配在中期报告前完成，长期任务放在中期报告之后。具体的任务安排见表-1。

Table 1: 项目开发时间计划

时间	任务描述
7 月 1 日 - 7 月 11 日（9 个工作日）	熟悉现有图片隐写代码的实现算法，并深入学习当前代码使用的语言 TypeScript 和使用的框架 React
7 月 12 日 - 7 月 26 日（10 个工作日）	阅读文献，查找资料，从算法层面设计出解决方案的草稿
7 月 27 日 - 8 月 8 日（8 个工作日）	代码实现基本的解决方案
8 月 9 日 - 8 月 14 日（4 个工作日）	在 Facebook 和 Twitter 上分别测试现有解决方案，评估是否较之前的方案在稳定性和人类感官不可察觉性上是否有所提高，并总结出需要改进的方向。
8 月 15 日（中期报告）- 8 月 20 日（5 个工作日）	根据导师的建议继续完善算法，修改
8 月 21 日 - 8 月 28 日（5 个工作日）	拓展当前代码，使当前代码可以接收所有格式的图片
8 月 29 日 - 9 月 5 日（5 个工作日）	整合当前所有实现的方案，并再次进行完整的测试
9 月 6 日 - 9 月 20 日（10 个工作日）	根据导师的反馈和意见，如果图片隐写算法实现顺利的话，可以进一步调研相关文献，考察前沿的视频和音频隐写解决方案
9 月 21 日 - 9 月 30 日（6 个工作日）	总结调研结果，设计一套基于 Maskbook 的实际需求的算法，并撰写结项报告

[1]https://help.twitter.com/en/using-twitter/tweeting-gifs-and-pictures