

به نام خدا



درس امنیت سیستم های کامپیوتری

تمرین سری دوم فصل دو (پرومتم)

مدرس درس:

جناب آقای دکتر دیانت

تهیه شده توسط:

فرزان رحمانی،

محمدحسین عباسپور

تاریخ ارسال: ۱۴۰۳/۰۱/۳۰

سوال ۱:

پیاده‌سازی یک روش نهان‌کاوی به عنوان آشکارسازی بر نهان‌نگاری به روش LSB. یعنی فرض کنید که ما با روش LSB عملیات نهان‌نگاری را انجام دادیم، شما باید یک روش نهان‌کاوی به منظور تشخیص آن پیاده‌سازی کنید.

پاسخ ۱:

کد موردنظر برای این بخش را در فایل Q1.ipynb پیاده‌سازی کردیم. برای اینکه یک تصویر داشته باشیم تا بتوانیم عملیات نهان‌کاوی را روی آن انجام دهیم، در کدمان یک بخش برای encode کردن تصویر، و یک بخش برای decode آن قرار دادیم. سپس تصویر موردنظر را با پیام دلخواه encode کردیم. همانطور که می‌دانیم در این روش برای encode کردن تصویر، پیام در بیت‌های LSB پیکسل‌ها قرار می‌گیرند؛ چرا که این بیت‌ها کم‌ارزش‌ترین بیت‌ها هستند و تغییر در آن‌ها، تفاوت چندانی در تصویر ایجاد نمی‌کند. در تابع Encode، این اتفاق می‌افتد.



شکل ۱: تصویر اصلی قبل از عملیات encoding

حال پس از انتخاب تصویر موردنظر برای عملیات نهان‌کاوی، برنامه را run کرده و گزینه encode را انتخاب می‌کنیم تا تصویر رمزگذاری شده برای مرحله بعد را آماده نماییم. همچنین کلمه Security را به عنوان پیام پنهان به برنامه می‌دهیم. در انتها نیز آدرس مقصد برای ذخیره عکس encode شده را می‌دهیم.

```
Encode the input image

[30] if __name__ == "__main__":
    Stego()

--Welcome to steganography program--
1: Encode message into image.
2: Decode message from image.
1
Enter Source Image Path.
farzan.png
Enter Message to Hide.
security of computer systems
Enter Destination Image Path.
farzan_encoded.png
Encoding...
RGBA
Image Encoded Successfully
```

شکل ۲: نحوه encode کردن تصویر

حال تصویر رمزنگاری شده را بررسی می‌کنیم. همانطور که در شکل زیر نیز می‌بینید، تصویر رمزنگاری شده تفاوت چندانی با تصویر اصلی ندارد؛ به گونه‌ای که تغییرات با چشم غیرمسلح قابل مشاهده نیست. یکی از دلایل این پدیده می‌تواند کوتاه بودن متن پیام باشد. عبارت \$t3g0 به کار رفته هنگام ذخیره‌سازی تصویر نیز برای تشخیص انتهای پیام می‌باشد.



شکل ۳: تصویر encode شده

حال از تابع Decode برای نهان‌کاوی تصویر encode شده استفاده می‌نماییم. همانطور که در شکل زیر مشاهده می‌نمایید، برنامه به درستی کلمه Security که به عنوان پیام پنهان به تابع Encode داده بودیم را برای ما چاپ می‌کند.

```
Decode the encoded image

[33] if __name__ == "__main__":
      Stego()

--Welcome to steganography program--
1: Encode message into image.
2: Decode message from image.
2
Enter Source Image Path.
farzan_encoded.png
Decoding...
Hidden Message: security of computer systems
```

شکل ۴: نحوه decode کردن تصویر

Reference

سوال ۲:

پیاده سازی یک روش نشان گذاری. پیاده سازی باید به زبان های ++C یا Python باشد.

پاسخ ۲:

نشان گذاری (watermarking) فرآیند جاسازی یک سیگنال دیجیتال منحصر به فرد و قابل شناسایی است که به عنوان واترمارک شناخته می شود. واترمارکینگ در یک شی چندرسانه ای مانند تصویر، ویدئو یا فایل صوتی واترمارک می تواند برای تأیید صحت یا مالکیت شی یا برای ردیابی توزیع و استفاده از آن استفاده شود. واترمارک ها اغلب به گونه ای طراحی می شوند که برای چشم یا گوش انسان نامحسوس باشند تا در کیفیت یا محتوای رسانه تداخلی ایجاد نکنند. تکنیک های مختلفی برای واترمارک وجود دارد، از جمله واترمارک های قابل مشاهده و غیر قابل مشاهده و واترمارک های شکننده و غیرشکننده.

ما برای این پیاده سازی این تمرین از روش نشان گذاری (واترمارکینگ) از نوع قابل مشاهده و غیرشکننده استفاده کردیم.

واترمارک قابل مشاهده: نمونه ای از واترمارک قابل مشاهده زمانی است که یک عکاس آرم یا اطلاعات حق چاپ خود را به یک تصویر اضافه می کند. این باعث می شود که تصویر متعلق به آنها باشد و بدون اجازه نمی توان از آن استفاده کرد.

واترمارک غیرشکننده: واترمارک غیرشکننده نوعی از واترمارک است که در آن واترمارک همچنان قابل تشخیص است حتی اگر رسانه به نحوی تغییر کرده باشد. به عنوان مثال، اگر هنرمندی بخواهد از موسیقی خود در برابر دزدی محافظت کند، می تواند یک واترمارک قوی در فایل صوتی جاسازی کند. حتی اگر کسی سعی کند فایل را با تغییر میزان بیت، فرمت یا طول تغییر دهد، واترمارک همچنان قابل تشخیص است. این باعث می شود که برای ردیابی توزیع رسانه های دارای حق چاپ مفید باشد. هدف از واترمارک غیرشکننده این است که اطمینان حاصل شود که حتی اگر رسانه به نحوی تغییر کرده باشد، مانند فشردن سازی، برش، تغییر اندازه، چرخش، اضافه کردن نویز هنوز واترمارک قابل شناسایی است. در ادامه تصویر اصلی را مشاهده می کنید که برای نشان گذاری انتخاب کرده ایم.



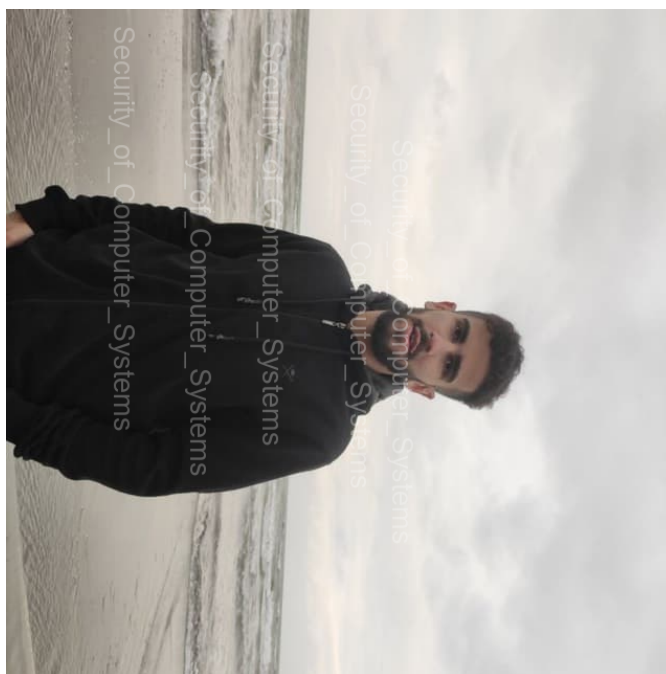
شکل ۵: تصویر اصلی

در برخی موارد، واترمارک های غیرشکننده نیز قابل مشاهده هستند. به عنوان مثال، یک شبکه تلویزیونی ممکن است یک لوگوی قابل مشاهده به پخش خود اضافه کند و در عین حال یک واترمارک غیرشکننده را نیز تعبیه کند که برای بینندگان قابل مشاهده نیست. در این سوال ما با استفاده از کتابخانه های PIL و OpenCV یک جمله را به عنوان واترمارک به صورت تصادفی در ۵ نقطه ی تصویر اصلی با شفافیت کم قرار دادیم.



شکل ۶: تصویر watermark شده

سپس تصویر واترمارک شده را تحت چهار عملیات چرخش، برش، تغییر اندازه و افزودن نویز بررسی کردیم. همانطور که مشاهده می‌کنید، واترمارک‌ها از بین نرفته‌اند و نشان‌دهنده‌ی غیرشکننده و قابل مشاهده بودن آن است.



شکل ۷: اعمال چرخش بر روی تصویر watermark شده



شکل ۸: اعمال تغییر اندازه بر روی تصویر watermark شده



شکل ۹: اعمال نویز بر روی تصویر watermark شده



شکل ۱۰: اعمال برش بر روی تصویر watermark شده

[Reference](#)