دانشکده مهندسی کامپیوتر

استاد درس: دکتر دیانت

بهار ۱۴۰۲

گزارش پروژه
درس امنیت سیستم های کامپیوتری

نیما کمبرانی، فاطمه زهرا بخشنده
شماره دانشجویی: ۹۸۵۲۱۴۲۳ ، ۹۸۵۲۲۱۵۷

مینی پروژه ۱

---

۱ سوال ۱

۱.۱ توضیحات

در نهان سازی داده ها به روش LSB ، ابتدا با گرفتن تصویر و پیام، مخفی سازی شروع می‌شود. پیام مورد نظر در بیت های LSB پیکسل ها ذخیره می شوند. در مرحله بعد، برای تشخیص پنهان شدن پیام در تصویر، در پایان پیام یک رشته مشخص افزوده می‌شود. از وجود این رشته در هنگام بازیابی پیام نیز استفاده می‌شود. بدین صورت که با شروع از ابتدا با رسیدن به این رشته مشخص، پایان پیام و همچنین وجود پیام مشخص می‌شود.

۲.۱ شکل

نمونه اجرای کد برای تصویری که در آن پیامی مخفی نشده است در شکل ۱ آمده است. همانطور که در شکل پیداست، اگر پیامی در تصویر موجود نباشد، پیام No Hidden Message Found_ چاپ می‌شود.

```
D:\uni\security\projects\p1>python -u "d:\uni\security\projects\p1\LSB.py"
Enter 1 for Encoding and 2 for Decoding:
2
Enter source image path to be decoded:
.\not_encoded.jpg
No Hidden Message Found
```

شکل ۱: خروجی کنسول در هنگام اجرای کد برای تشخیص مخفی شدن پیام در تصویر

حال یک پیام را در این شکل نهان سازی می کنیم.

```
D:\uni\security\projects\p1>python -u "d:\uni\security\projects\p1\LSB.py"
Enter 1 for Encoding and 2 for Decoding:
1
Enter Source_path, destination_path in order seprted by space:
.\not_encoded.jpg .\encoded.png
Enter Message:
hello this message is encoded in the picture :))
Message Successfully Encoded
```

شکل ۲: خروجی کنسول در هنگام اجرای کد برای مخفی سازی پیام در تصویر

حال می خواهیم پیام نهان سازی شده در این تصویر را تشخیص دهیم. نمونه اجرای کد برای شناسایی پیام مخفی شده در یک تصویر در شکل ۳ آمده است.

```
D:\uni\security\projects\p1>python -u "d:\uni\security\projects\p1\LSB.py"
Enter 1 for Encoding and 2 for Decoding:
2
Enter source image path to be decoded:
.\encoded.png
Hidden Message: hello this message is encoded in the picture :))
```

شکل ۳: خروجی کنسول در هنگام اجرای کد برای شناسایی پیام مخفی شده در تصویر

۳.۱ کد سوالات

کد این سوال با پایتون نوشته شده است. با اجرای کد، ابتدا انتخاب می کنیم که هدف مخفی سازی پیام است یا تشخیص پیام مخفی سازی شده. اگر هدف، تشخیص پیام باشد، آدرس تصویر را از ما خواسته و سپس اگر پیامی در تصویر نهان شده بود، آن را چاپ می کند، و در صورتی که پیامی نبود گزارش می دهد.

اگر هدف مخفی سازی پیام باشد، آدرس تصویر مبدا و مقصد ذخیره سازی را وارد کرده و سپس پیام مورد نظر را جهت نهان سازی وارد می کنیم.

۲ سوال ۲

۱.۲ توضیحات

نشانه‌گذاری به افزودن متن یا لوگو بصورت قابل شناسایی یا غیرقابل دیدن به یک تصویر گفته می‌شود. اینکار می‌تواند به منظور حفظ مالکیت یک تصویر و جلوگیری از استفاده از این تصویر بدون اجازه یا با تغییر محتوا صورت پذیرد. نشانه گذاری هایی مانند افزودن آرم یک شبکه تلویزیونی به تصویر به منظور جلوگیری از پخش تصویر بدون اجازه یا قرار دادن امضای دیجیتال به همراه فایل تصویر به منظور حفظ صحت عکس از مثال های نشانه گذاری هستند. در این سوال یک نمونه نشانه گذاری از نوع قابل دیدن (visible) و غیرشکننده (non-fragile) پیاده‌سازی شده است. در این کد با گرفتن یک لوگو آن را برروی تصویر قرار می‌دهد.

۲.۲ نمونه

```
D:\uni\security\projects\p1>python -u "d:\uni\security\projects\p1\Watermarking.py"
Enter watermark path: .\watermark.jpg
Enter base image path: .\yusef.jpg
Enter output image path: y.png
Watermark added successfully
```

شکل ۴: خروجی کنسول برای نشان گذاری تصویر

شکل ۵: تصویر به همراه لوگو اضافه شده به آن به عنوان یه نشانه گذاری قابل دیدن و غیر شکننده

۳.۲ کد سوال

کد سوال با استفاده از زبان پایتون پیاده شده است. در این برنامه با گرفتن آدرس یک تصویر به عنوان لوگو و یک تصویر به‌عنوان تصویر پایه می‌گیرد. سپس با تغییر اندازه لوگو به اندازه ۵۰ در ۵۰ آن را در گوشه تصویر قرار می‌دهد. در نهایت تصویر نهایی در یک فایل جدید ذخیره می‌شود.

```python
import numpy as np
from PIL import Image
end_str="$$\0"
def Encode():
    src, dest = input("Enter Source_path, destination_path in order seprted by space:\n").split()
    message=input("Enter Message:\n")
    img = Image.open(src, 'r')
    width, height = img.size
    array = np.array(list(img.getdata()))

    if img.mode == 'RGB':
        n = 3
    elif img.mode == 'RGBA':
        n = 4
    total_pixels = array.size//n

    message += end_str
    b_message=""
    for ch in message:
        b_message+=format(ord(ch), "08b")

    req_pixels = len(b_message)

    if req_pixels > total_pixels:
        print("File size is not enough!")

    else:
        index=0
        for p in range(total_pixels):
            for q in range(0, n):
                if index < req_pixels:
                    array[p][q] = int(bin(array[p][q])[2:9] + b_message[index], 2)
                    index += 1
                else:
                    break

            if index >= req_pixels:
                break

        array=array.reshape(height, width, n)
        enc_img = Image.fromarray(array.astype('uint8'), img.mode)
        enc_img.save(dest)
        print("Message Successfully Encoded")
```

Listing :۱ Least Significant Bit steganography Encoder

```
import numpy as np
from PIL import Image
end_str="$$\0"
def Decode():
    src = input("Enter source image path to be decoded:\n")

    img = Image.open(src, 'r')
    array = np.array(list(img.getdata()))

    if img.mode == 'RGB':
        n = 3
    elif img.mode == 'RGBA':
        n = 4
    total_pixels = array.size//n

    hidden_bits = ""
    for p in range(total_pixels):
        for q in range(0, n):
            hidden_bits += (bin(array[p][q])[2:][-1])

    hidden_bits = [hidden_bits[i:i+8] for i in range(0,len(hidden_bits), 8)]

    message = ""
    for i in range(len(hidden_bits)):
        if message[-1*len(end_str):] == end_str:
            break

        message += chr(int(hidden_bits[i], 2))

    if end_str in message:

        print("Hidden Message:", message[:-1*len(end_str)])
        return
    else:
        print("No Hidden Message Found")
```

Listing ۲: Least Significant Bit steganography Decoder

```python
import numpy as np
from PIL import Image

logo_addr=input("Enter watermark path: ")
base_path=input("Enter base image path: ")
out_path=input("Enter output image path: ")

logo = Image.open(logo_addr, 'r').resize((50,50))
l_width, l_height = logo.size
base_img=Image.open(base_path, 'r')
b_width, b_height = base_img.size
out_img=base_img.copy()
out_img.paste(logo,(5,5))
out_img.save(out_path)
print("Watermark added successfully")
```

Listing :۳ Adding a logo(visible and non-fragile watermark) to the image implementation