ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO BÀI TẬP

AN TOÀN VÀ PHỤC HỒI DỮ LIỆU

Đề tài:

PDF STEGANOGRAPHY TOOL

(Công cụ ẩn và trích xuất dữ liệu trong PDF)

GVHD: Đặng Trần Minh Hậu

SVTH: Hoàng Quốc Việt
MSSV: 22120429
Lớp: An toàn và phục hồi dữ liệu - CQ2022/22

TP. Hồ Chí Minh, tháng 12 năm 2025

Mục lục

1 Giới thiệu

1.1 Tổng quan về Steganography

Steganography (ẩn giấu thông tin) là nghệ thuật và khoa học về việc ẩn thông tin trong một phương tiện khác sao cho sự tồn tại của thông tin ẩn không bị phát hiện. Khác với mã hóa (cryptography) - làm cho thông tin không thể đọc được, steganography tập trung vào việc ẩn sự tồn tại của thông tin.

Steganography có lịch sử lâu đời, từ thời cổ đại với việc sử dụng mực vô hình, đến thời hiện đại với các kỹ thuật kỹ thuật số phức tạp. Trong thời đại số, steganography được ứng dụng rộng rãi trong:

- Bảo mật thông tin nhạy cảm
- Watermarking kỹ thuật số
- Chống vi phạm bản quyền
- Truyền thông bí mật
- Xác thực và toàn vẹn dữ liệu

1.2 Động lực và mục tiêu

Trong bối cảnh an toàn thông tin ngày càng được chú trọng, việc nghiên cứu và phát triển các công cụ steganography trở nên cấp thiết. PDF (Portable Document Format) là một trong những định dạng tài liệu phổ biến nhất, được sử dụng rộng rãi trong trao đổi thông tin.

Mục tiêu của bài tập này là xây dựng một công cụ hoàn chỉnh cho phép:

1. Ẩn dữ liệu file vào trong file PDF mà không làm thay đổi nội dung hiển thị
2. Trích xuất dữ liệu file đã ẩn từ file PDF
3. Hỗ trợ nhiều định dạng file phổ biến
4. Cung cấp giao diện dễ sử dụng (CLI và GUI)
5. Đảm bảo tính toàn vẹn của dữ liệu

1.3 Phạm vi bài tập

Bài tập tập trung vào:

- Nghiên cứu cấu trúc file PDF
- Phát triển thuật toán ẩn/trích xuất dữ liệu
- Triển khai công cụ bằng Python
- Hỗ trợ các định dạng: .txt, .jpg, .png, .pdf, .docx, .exe
- Xây dựng giao diện người dùng thân thiện

2 Cấu trúc file PDF

2.1 Tổng quan về PDF

PDF (Portable Document Format) là định dạng file được phát triển bởi Adobe Systems vào năm 1993. PDF được thiết kế để trình bày tài liệu một cách nhất quán trên các nền tảng khác nhau, bao gồm phần mềm, phần cứng và hệ điều hành.

2.2 Cấu trúc cơ bản

Một file PDF bao gồm 4 phần chính:

1. **Header (Phần đầu)**
   - Chứa thông tin về phiên bản PDF
   - Định dạng: `%PDF-x.x` (ví dụ: %PDF-1.7)
   - Luôn nằm ở dòng đầu tiên của file
2. **Body (Thân)**
   - Chứa các object (đối tượng) của PDF
   - Bao gồm: text, images, fonts, pages, metadata
   - Mỗi object có định danh duy nhất (object number)
3. **Cross-Reference Table (Bảng tham chiếu chéo)**
   - Bắt đầu bằng từ khóa `xref`
   - Chứa vị trí của mỗi object trong file
   - Cho phép truy cập nhanh các object mà không cần đọc toàn bộ file
4. **Trailer (Phần cuối)**
   - Bắt đầu bằng từ khóa `trailer`
   - Chứa thông tin về root object và số lượng object
   - Kết thúc bằng marker `%%EOF`

2.3 Ví dụ cấu trúc PDF

```
%PDF-1.7                                <- Header

1 0 obj                                 <- Body
<<
  /Type /Catalog
  /Pages 2 0 R
>>
endobj

2 0 obj
<<
  /Type /Pages
```

```
  /Kids [3 0 R]
  /Count 1
>>
endobj

...

xref                                    <- Cross-reference table
0 5
0000000000 65535 f
0000000009 00000 n
0000000058 00000 n
...

trailer                                 <- Trailer
<<
  /Size 5
  /Root 1 0 R
>>
startxref
500
%%EOF                                   <- End marker
```

2.4 Marker %%EOF

Marker %%EOF (End Of File) đóng vai trò quan trọng:

- Đánh dấu kết thúc của PDF document
- PDF readers dừng đọc khi gặp marker này
- Mọi dữ liệu sau %%EOF bị bỏ qua bởi PDF viewers
- Đây là điểm then chốt cho kỹ thuật steganography của chúng ta

3 Kỹ thuật ẩn dữ liệu trong PDF

3.1 Phân tích các kỹ thuật steganography trong PDF

Có nhiều kỹ thuật để ẩn dữ liệu trong PDF:

1. **Metadata Embedding:** Ẩn trong metadata của PDF
2. **Text Steganography:** Ẩn trong whitespace, font properties
3. **Image Steganography:** Ẩn trong các hình ảnh được nhúng
4. **Object Modification:** Thay đổi các PDF objects
5. **Appending After EOF:** Thêm dữ liệu sau marker %%EOF

3.2 Phương pháp được chọn: Appending After EOF

Bài tập này sử dụng phương pháp **Appending After EOF** với các lý do:

Ưu điểm:

- Đơn giản và hiệu quả
- Không thay đổi cấu trúc PDF gốc
- Nội dung hiển thị không bị ảnh hưởng
- PDF vẫn mở được bình thường
- Dễ triển khai và bảo trì

Nhược điểm:

- Tăng kích thước file
- Có thể phát hiện qua phân tích kích thước
- Dữ liệu có thể bị xóa bởi công cụ tối ưu PDF

3.3 Cấu trúc dữ liệu ẩn

Dữ liệu được tổ chức theo định dạng:

```
%%EOF
<<HIDDEN_DATA_START>>
[4 bytes]  - Length of the file name
[n bytes]  - Original file name (UTF-8)
[4 bytes]  - Length of the file data
[m bytes]  - Hidden file content
<<HIDDEN_DATA_END>>
```

Giải thích:

- «HIDDEN_DATA_START»: Marker bắt đầu (22 bytes)
- Độ dài tên file: 4 bytes (unsigned integer, little-endian)
- Tên file: Chuỗi UTF-8 chứa tên file gốc
- Độ dài dữ liệu: 4 bytes (unsigned integer, little-endian)
- Nội dung file: Binary data của file ẩn
- «HIDDEN_DATA_END»: Marker kết thúc (20 bytes)

3.4 Thuật toán ẩn dữ liệu

```python
def hide_file(pdf_path, file_to_hide, output_path):
    # Step 1: Read original PDF content
    pdf_content = read_file(pdf_path)

    # Step 2: Read file to hide
    hidden_data = read_file(file_to_hide)
    filename = get_filename(file_to_hide)

    # Step 3: Prepare metadata
    filename_bytes = filename.encode('utf-8')
    filename_length = len(filename_bytes)
    data_length = len(hidden_data)

    # Step 4: Create hidden data package
    hidden_package = (
        pack('<I', filename_length) +
        filename_bytes +
        pack('<I', data_length) +
        hidden_data
    )

    # Step 5: Find %%EOF position
    eof_position = pdf_content.rfind(b'%%EOF')

    # Step 6: Insert data after %%EOF
    modified_pdf = (
        pdf_content[:eof_position + len(b'%%EOF')] +
        b'\n' +
        MARKER_START +
        hidden_package +
        MARKER_END +
        b'\n'
    )

    # Step 7: Write output file
    write_file(output_path, modified_pdf)

    return True
```

Listing 1: Thuật toán ẩn file

3.5 Thuật toán trích xuất dữ liệu

```python
def extract_file(pdf_path, output_dir):
    # Step 1: Read PDF content
    pdf_content = read_file(pdf_path)

    # Step 2: Find markers
    start_pos = pdf_content.find(MARKER_START)
    end_pos = pdf_content.find(MARKER_END)

```

```
    if start_pos == -1 or end_pos == -1:
        raise Error("No hidden data found")

    # Step 3: Extract package
    start_pos += len(MARKER_START)
    hidden_package = pdf_content[start_pos:end_pos]

    # Step 4: Parse metadata
    filename_length = unpack('<I', hidden_package[0:4])[0]
    filename = hidden_package[4:4+filename_length].decode('utf-8')

    data_length_pos = 4 + filename_length
    data_length = unpack('<I',
        hidden_package[data_length_pos:data_length_pos+4])[0]

    # Step 5: Extract data
    data_start = data_length_pos + 4
    hidden_data = hidden_package[data_start:data_start+
        data_length]

    # Step 6: Check integrity
    if len(hidden_data) != data_length:
        raise Error("Data corruption detected")

    # Step 7: Write file
    output_path = os.path.join(output_dir, filename)
    write_file(output_path, hidden_data)

    return output_path
```

Listing 2: Thuật toán trích xuất file

4 Tính năng và chức năng

4.1 Danh sách tính năng

Công cụ PDF Steganography cung cấp các tính năng sau:

1. **Ẩn file vào PDF**
   - Nhúng file bất kỳ vào trong PDF
   - Hỗ trợ các định dạng: .txt, .jpg, .png, .pdf, .docx, .exe
   - Giữ nguyên tên file và metadata
   - Không làm thay đổi nội dung hiển thị
2. **Trích xuất file từ PDF**
   - Tự động phát hiện và trích xuất file ẩn
   - Khôi phục chính xác tên file gốc
   - Kiểm tra tính toàn vẹn dữ liệu

- Xử lý trùng tên file tự động

3. **Kiểm tra PDF**
   - Phát hiện sự tồn tại của dữ liệu ẩn
   - Hiển thị thông tin về file ẩn
   - Không cần trích xuất để xem thông tin
4. **Giao diện người dùng**
   - Command Line Interface (CLI)
   - Graphical User Interface (GUI)
   - Log output chi tiết
   - Xử lý lỗi thân thiện

4.2 Các file hỗ trợ

Công cụ hỗ trợ ẩn các định dạng file sau:

Định dạng	**Mô tả**	**Use case**
.txt	Text files	Văn bản, note, code
.jpg	JPEG images	Hình ảnh, photos
.png	PNG images	Logo, screenshots
.pdf	PDF documents	Tài liệu, reports
.docx	Word documents	Văn bản định dạng
.exe	Executables	Programs, tools

Bảng 1: Các định dạng file được hỗ trợ

4.3 Kiến trúc hệ thống

Hệ thống được tổ chức theo kiến trúc 3 lớp:

1. **Core Layer (pdf_stego.py)**
   - Class `PDFSteganography`
   - Xử lý logic ẩn/trích xuất
   - Validate input/output
   - Error handling
2. **Interface Layer**
   - CLI (cli.py): Command-line interface
   - GUI (gui.py): Tkinter graphical interface
3. **Support Layer**
   - Requirements management
   - Documentation
   - Sample files

5 Chi tiết triển khai

5.1 Công nghệ sử dụng

Component	Technology	Version
Programming Language	Python	3.7+
PDF Processing	PyPDF2	3.0.1
Image Processing	Pillow	10.1.0
Document Processing	python-docx	1.1.0
GUI Framework	Tkinter	Built-in

Bảng 2: Stack công nghệ

5.2 Module Core: pdf_stego.py

Class PDFSteganography:

```
class PDFSteganography:
    # Constants
    MARKER = b"<<HIDDEN_DATA_START>>"
    MARKER_END = b"<<HIDDEN_DATA_END>>"
    SUPPORTED_FORMATS = ['.txt', '.jpg', '.png',
                         '.pdf', '.docx', '.exe']

    # Methods
    def validate_pdf(pdf_path) -> bool
    def validate_file_format(file_path) -> bool
    def hide_file(pdf_path, file_to_hide, output_path) -> bool
    def extract_file(pdf_path, output_dir) -> Optional[str]
    def check_hidden_data(pdf_path) -> bool
    def get_hidden_file_info(pdf_path) -> Optional[Tuple[str, int
       ]]
```

Listing 3: Cấu trúc class chính

Các phương thức chính:

- `validate_pdf()`: Kiểm tra tính hợp lệ của PDF
- `validate_file_format()`: Kiểm tra định dạng file hỗ trợ
- `hide_file()`: Ẩn file vào PDF
- `extract_file()`: Trích xuất file từ PDF
- `check_hidden_data()`: Kiểm tra sự tồn tại dữ liệu ẩn
- `get_hidden_file_info()`: Lấy thông tin file ẩn

5.3 Command Line Interface: cli.py

CLI cung cấp 3 sub-commands:

```
# Hide command
python cli.py hide <cover_pdf> <file_to_hide> <output_pdf>

# Extract command
python cli.py extract <stego_pdf> <output_dir>

# Check command
python cli.py check <pdf_file>
```

Listing 4: Cấu trúc CLI

Sử dụng `argparse` để xử lý arguments và provides detailed help messages.

5.4 Graphical User Interface: gui.py

GUI được xây dựng bằng Tkinter với 3 tabs:

1. **Tab "Ẩn File"**
   - File browser cho PDF gốc
   - File browser cho file cần ẩn
   - File browser cho output
   - Button thực hiện ẩn
2. **Tab "Trích xuất File"**
   - File browser cho PDF chứa dữ liệu ẩn
   - Directory browser cho output
   - Info panel hiển thị thông tin file ẩn
   - Button thực hiện trích xuất
3. **Tab "Kiểm tra File"**
   - File browser cho PDF cần kiểm tra
   - Button thực hiện kiểm tra
   - Text area hiển thị kết quả chi tiết

Tất cả các tab đều có:

- Log output realtime ở phía dưới
- Status bar hiển thị trạng thái
- Error handling với message boxes

5.5 Xử lý lỗi

Hệ thống xử lý các lỗi sau:

- File not found
- Invalid PDF format
- Unsupported file format
- No hidden data found
- Data corruption
- Permission errors
- I/O errors

Mỗi lỗi được xử lý với:

- Exception handling cụ thể
- Error message rõ ràng
- Logging chi tiết
- User-friendly notifications

6 Hướng dẫn sử dụng

6.1 Cài đặt

Bước 1: Cài đặt Python

Tải và cài đặt Python 3.7 trở lên từ https://www.python.org/downloads/

Bước 2: Cài đặt dependencies

```
pip install -r requirements.txt
```

6.2 Sử dụng GUI

Khởi động:

```
python gui.py
```

Ẩn file:

1. Chọn tab "Ẩn File"
2. Chọn PDF gốc
3. Chọn file cần ẩn
4. Chọn vị trí lưu output
5. Click "Ẩn File vào PDF"

Trích xuất file:

1. Chọn tab "Trích xuất File"
2. Chọn PDF chứa dữ liệu ẩn
3. Chọn thư mục output
4. Click "Trích xuất File từ PDF"

6.3 Sử dụng CLI

Ẩn file:

```
python cli.py hide cover.pdf secret.txt output.pdf
```

Trích xuất file:

```
python cli.py extract output.pdf ./extracted/
```

Kiểm tra PDF:

```
python cli.py check output.pdf
```

6.4 Ví dụ thực tế

Scenario 1: Ẩn tài liệu nhạy cảm

```
# Hide file confidential.docx into report.pdf
python cli.py hide report.pdf confidential.docx report_stego.pdf

# Send report_stego.pdf to email
# Receiver extract hidden file:
python cli.py extract report_stego.pdf ./output/
```

Scenario 2: Ẩn hình ảnh

```
# Hide image into PDF
python cli.py hide document.pdf photo.jpg document_with_photo.pdf

# Check
python cli.py check document_with_photo.pdf
```

7 Testing và đánh giá

7.1 Test cases

7.2 Performance

Đánh giá hiệu năng với các file sizes khác nhau:

ID	Test case	Expected result	Status
TC01	Ẩn file .txt vào PDF	File ẩn thành công	PASS
TC02	Ẩn file .jpg vào PDF	File ẩn thành công	PASS
TC03	Ẩn file .pdf vào PDF	File ẩn thành công	PASS
TC04	Trích xuất file từ PDF	File giống gốc	PASS
TC05	Kiểm tra PDF không có dữ liệu ẩn	Báo không tìm thấy	PASS
TC06	Kiểm tra PDF có dữ liệu ẩn	Hiển thị thông tin	PASS
TC07	Ẩn file không hỗ trợ	Báo lỗi định dạng	PASS
TC08	Mở PDF bằng viewer	PDF hiển thị bình thường	PASS
TC09	Ẩn file lớn (>10MB)	Xử lý thành công	PASS
TC10	Trích xuất với tên trùng	Tự động đổi tên	PASS

Bảng 3: Kết quả test cases

File size	Hide time	Extract time	Memory
< 1 MB	< 0.1s	< 0.1s	< 10 MB
1-10 MB	< 0.5s	< 0.3s	< 50 MB
10-50 MB	< 2s	< 1s	< 200 MB
> 50 MB	< 5s	< 3s	< 500 MB

Bảng 4: Performance metrics

7.3 Limitations

Giới hạn kỹ thuật:

- Tăng kích thước file PDF
- Có thể phát hiện bằng phân tích file size
- Không hoạt động với PDF encrypted
- Dữ liệu có thể bị xóa bởi PDF optimizer

Giới hạn bảo mật:

- Dữ liệu không được mã hóa
- Dễ bị phát hiện bởi công cụ chuyên dụng
- Không chống được forensic analysis

8 Kết luận

8.1 Tổng kết

Bài tập đã hoàn thành mục tiêu xây dựng công cụ PDF Steganography với đầy đủ các tính năng:

- Ẩn và trích xuất file từ PDF thành công
- Hỗ trợ đa dạng định dạng file
- Giao diện người dùng thân thiện (CLI và GUI)
- Không làm thay đổi nội dung hiển thị PDF
- Xử lý lỗi tốt và logging chi tiết

Kỹ thuật "Appending After EOF"được chứng minh là hiệu quả cho mục đích steganography cơ bản, phù hợp cho các ứng dụng cần độ phức tạp thấp và triển khai nhanh.

8.2 Đóng góp

Bài tập đóng góp:

1. Nghiên cứu và phân tích cấu trúc PDF
2. Triển khai thuật toán steganography đơn giản nhưng hiệu quả
3. Xây dựng công cụ hoàn chỉnh với cả CLI và GUI
4. Tài liệu hóa đầy đủ code và hướng dẫn sử dụng
5. Cung cấp ví dụ và test cases cụ thể

8.3 Hướng phát triển

Các cải tiến có thể thực hiện:

1. **Bảo mật:**
   - Thêm mã hóa AES cho dữ liệu ẩn
   - Password protection
   - Digital signature
2. **Tính năng:**
   - Nén dữ liệu trước khi ẩn
   - Ẩn nhiều file cùng lúc
   - Steganography vào PDF objects
   - Checksum validation
3. **Interface:**
   - Web interface
   - Mobile app
   - Batch processing
   - Drag and drop support
4. **Advanced techniques:**
   - LSB steganography in images
   - Metadata steganography
   - Hybrid approaches

8.4 Bài học kinh nghiệm

Qua quá trình thực hiện bài tập, tôi đã học được:

- Hiểu sâu về cấu trúc file PDF
- Kỹ thuật steganography và ứng dụng thực tế
- Phát triển ứng dụng Python hoàn chỉnh
- Thiết kế giao diện người dùng thân thiện
- Testing và debugging
- Documentation và technical writing

Tài liệu tham khảo

1. Adobe Systems (2020). PDF Reference, sixth edition, version 1.7. Adobe Systems Incorporated.
2. PyPDF2 Documentation. https://pypdf2.readthedocs.io/
3. Python Tkinter Documentation. https://docs.python.org/3/library/tkinter.html
4. Katzenbeisser, S., & Petitcolas, F. A. (2000). Information hiding techniques for steganography and digital watermarking. Artech house.
5. Cheddad, A., Condell, J., Curran, K., & Mc Kevitt, P. (2010). Digital image steganography: Survey and analysis of current methods. Signal processing, 90(3), 727-752.
6. Warkentin, M., Bekkering, E., & Schmidt, M. B. (2008). Steganography: Forensic, security, and legal issues. Journal of Digital Forensics, Security and Law, 3(2), 17-34.

A Source Code

Full source code: https://github.com/hVie1314/PDF-steganography-tool

A.1 Cấu trúc project

```
source/
    |-- pdf_stego.py          # Core module
    |-- cli.py                # CLI interface
    |-- gui.py                # GUI interface
    |-- requirements.txt      # Dependencies
    |-- README.txt            # User guide
    |-- report.tex            # This report
    |-- samples/              # Sample files
        |-- sample.pdf
        |-- secret.txt
```

A.2 Dependencies

```
PyPDF2==3.0.1
Pillow==10.1.0
python-docx==1.1.0
```