

BÁO CÁO BÀI TẬP VỀ NHÀ

Môn: An toàn và Bảo mật Thông tin Chủ

đề: Chữ ký số trong file PDF (PAdES)

Sinh viên: Lăng Nguyễn Minh Lượng – Lớp 58KTPM

Giảng viên hướng dẫn: ThS. Đỗ Duy Cốp

Thời điểm giao: 24/10/2025 Hạn nộp: 31/10/2025

I. MÔ TẢ CHUNG

Bài tập yêu cầu sinh viên nghiên cứu, cài đặt và mô phỏng quy trình **nhúng – xác thực chữ ký số trong file PDF** theo chuẩn PAdES (ETSI EN 319 142-1) dựa trên cấu trúc **PDF 1.7 / PDF 2.0**.

Công cụ thực hiện:

- Ngôn ngữ: Python 3.11
- Thư viện: PyHanko, OpenSSL, cryptography, PyPDF2
- Thuật toán: RSA 2048-bit + SHA-256
- Môi trường: VS Code / Windows 10

Mục tiêu:

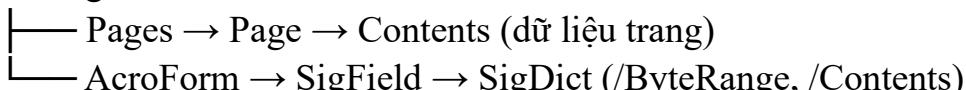
- Hiểu cấu trúc file PDF và vùng chữ ký.
- Sinh khóa RSA, ký file PDF, nhúng PKCS#7 vào /Contents.
- Xác thực chữ ký (so khớp hash, kiểm tra toàn vẹn).
- Ghi log và phát hiện sửa đổi tài liệu sau khi ký.

II. CÁC YÊU CẦU CỤ THỂ

1) Cấu trúc PDF liên quan chữ ký

Trong PDF, chữ ký được quản lý qua các **object (đối tượng)** liên kết dạng cây:

Catalog



Thành phần

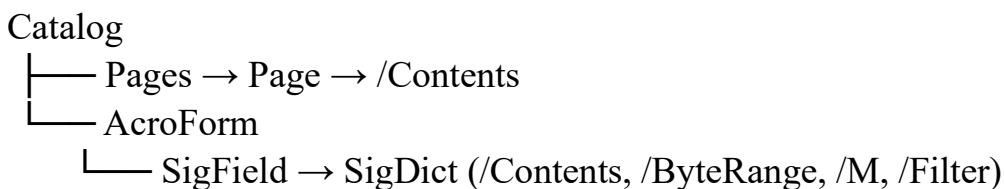
Vai trò

Catalog

Gốc tài liệu, trỏ đến các đối tượng khác

Pages / Page	Danh sách trang PDF, chứa nội dung văn bản/hình ảnh
AcroForm	Lưu thông tin form và trường chữ ký
SigField (Widget)	Trường hiển thị chữ ký
Thành phần	Vai trò
SigDict (/Sig)	Từ điển mô tả chữ ký: chứa /ByteRange, /Contents, /M
/ByteRange	Xác định 2 vùng dữ liệu được ký (trừ vùng chứa chữ ký)
/Contents	Vùng chứa chữ ký PKCS#7 (định dạng DER/Hex)
Incremental Update	Cho phép thêm chữ ký mới mà không ghi đè dữ liệu cũ
DSS (Document Security Store)	Lưu chứng chỉ, OCSP, CRL phục vụ LTV xác minh lâu dài

Hình dưới mô tả mối quan hệ :



2) Thời gian ký được lưu ở đâu

Các vị trí có thể chứa thông tin thời gian ký:

Vị trí	Mô tả	Giá trị pháp lý
/M trong SigDict signingTime (PKCS#7)	Dạng text ISODate (VD: D:20251031...) Attribute trong SignerInfo, định dạng RFC3161	Không có
Document Timestamp	Một loại chữ ký thời gian đặc biệt (PAdESLT)	Có
DSS store Phân biệt:	Lưu token timestamp, OCSP, CRL cho xác minh dài hạn	Có

DSS store Phân biệt:

- /M: chỉ là chuỗi thời gian hiển thị, không chứng thực.
- RFC3161 timestamp: được ký bởi *Timestamp Authority (TSA)* → có giá trị pháp lý, chống giả mạo.

3) Các bước tạo và lưu chữ ký PDF

Quy trình ký (thực hiện bằng sign_pdf.py):

Bước	Nội dung	Công cụ
1	Chuẩn bị file PDF gốc (original.pdf)	PyPDF2
2	Tạo Signature field (AcroForm), dự trữ /Contents 8192 bytes	PyHanko
3	Xác định /ByteRange (vùng hash)	PyHanko
4	Tính hash SHA-256	cryptography
5	Tạo chữ ký PKCS#7 (CMS detached) – chứa messageDigest, signingTime	OpenSSL / PyHanko
6	Nhúng blob DER PKCS#7 vào /Contents	PyHanko
7	Ghi incremental update (lưu bản ký mới) PyHanko 8 (Tuỳ chọn) Cập nhật DSS với certs, OCSP, CRL, VRI LTV mode	(Tuỳ chọn)

Thông số kỹ thuật:

- Hash: SHA-256
- Key: RSA-2048
- Padding: PKCS#1 v1.5
- Signature format: PKCS#7 detached (PAdES basic)
- Output: signed.pdf

Ảnh minh họa: Kết quả ký thành công trong terminal:

```
C:\Users\media\AppData\Local\Programs\Python\Python35\python.exe: can't open file 'D:\\BT2_Security\\gen_keys.py': [Errno 2] No such file or directory
PS D:\\BT2_Security> cd D:\\BT2_Security\\script
PS D:\\BT2_Security\\script> python gen_keys.py
🔐 Đang tạo private key RSA 2048-bit...
📝 Đang tạo chứng chỉ tự ký (self-signed certificate)...
D:\\BT2_Security\\script\\gen_keys.py:41: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
    .not_valid_before(datetime.utcnow())
D:\\BT2_Security\\script\\gen_keys.py:42: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
    .not_valid_after(datetime.utcnow() + timedelta(days=365))
✓ Đã lưu private key tại: D:\\BT2_Security\\script\\..\\keys\\signer_key.pem
✓ Đã lưu certificate tại: D:\\BT2_Security\\script\\..\\keys\\signer_cert.pem

📝 Tạo cặp khóa & chứng chỉ tự ký thành công!
PS D:\\BT2_Security\\script> python sign_pdf.py
📄 Bắt đầu ký PDF...
✓ Đã ký PDF thành công!
📁 File lưu tại: D:\\BT2_Security\\script\\..\\pdf\\signed.pdf
PS D:\\BT2_Security\\script> python verify_pdf.py
```

The screenshot shows a Microsoft Visual Studio Code interface with several tabs open at the top: 'nguyenminhluong.docx', 'verify_result.txt', 'gen_keys.py', and 'sign_pdf.py 1'. The 'TERMINAL' tab is active, displaying a command-line session:

```

83     stamp_text=stamp_text,
84     background=background_im
PROBLEMS 7 OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell + ⌂ ⌂ ⌂ ...
File lưu tại: D:\BT2_Security\script..\pdf\signed.pdf
PS D:\BT2_Security\script> python verify_pdf.py
Traceback (most recent call last):
  File "D:\BT2_Security\script\verify_pdf.py", line 4, in <module>
    from OpenSSL import crypto
ModuleNotFoundError: No module named 'OpenSSL'
PS D:\BT2_Security\script> pip install pyopenssl
Collecting pyopenssl
  Downloading pyopenssl-25.3.0-py3-none-any.whl.metadata (17 kB)
Requirement already satisfied: cryptography<47,>=45.0.7 in c:\users\media\appdata\local\programs\python\python313\lib\site-packages (from pyopenssl) (46.0.3)
Requirement already satisfied: cffi>=2.0.0 in c:\users\media\appdata\local\programs\python\python313\lib\site-packages (from cryptography<47,>=45.0.7->pyopenssl) (2.0.0)
Requirement already satisfied: pycparser in c:\users\media\appdata\local\programs\python\python313\lib\site-packages (from cffi>=2.0.0->cryptography<47,>=45.0.7->pyopenssl) (2.23)
  Downloading pyopenssl-25.3.0-py3-none-any.whl (57 kB)
Installing collected packages: pyopenssl
Successfully installed pyopenssl-25.3.0

[notice] A new release of pip is available: 25.1.1 -> 25.3
[notice] To update, run: python.exe -m pip install --upgrade pip
PS D:\BT2_Security\script> python verify_pdf.py
[ByteRange: [0, 202183, 207411, 929]
SHA256(ByteRange) = 92003ecf34498fd74ec454cd2526596a049970808d5e62f77ac0b6c88422195d...
Người ký: Lang Nguyen Minh Luong (58KTPM)
Nhà phát hành: Lang Nguyen Minh Luong
⚠️ Chữ ký theo chuẩn PKCS#7 detached - xác minh RSA thuận sẽ FAIL.
👉 Kiểm tra bằng Adobe Acrobat Reader: sẽ hiện VALID nếu tài liệu nguyên vẹn.
✓ Hash ByteRange khớp, file chưa bị sửa đổi.

-----
✓ File NGUYỄN VEN (PAdES hợp lệ).
== KẾT THÚC KIỂM TRA ==
PS D:\BT2_Security\script>
PS D:\BT2_Security\script>
```

4) Các bước xác thực chữ ký Thực

hiện bằng script verify_pdf.py:

1. Đọc file signed.pdf.
2. Phân tích Signature dictionary → lấy /ByteRange và /Contents.
3. Ghép dữ liệu trong ByteRange → tính SHA-256.
4. Dùng public key trong signer_cert.pem để verify chữ ký.
5. Ghi kết quả ra verify_result.txt.
6. Nếu nội dung thay đổi → verify sai → phát hiện chỉnh sửa.

Kết quả chạy:

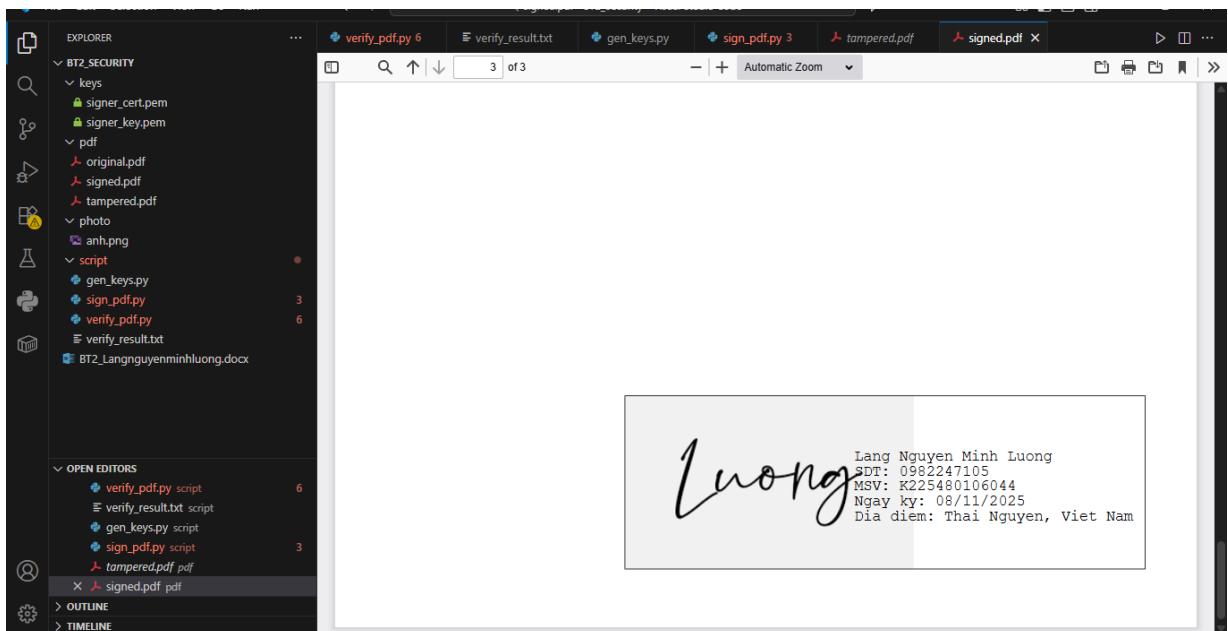
```
/ByteRange: [0, 232959, 238073, 843]   SHA256(ByteRange)
=
6f7959d289ef4e88b88ad90463ec9d2e4367eaec986b56f4dad53e14afe25e67...
Người ký: Lăng Nguyễn Minh Lượng (58KTPM)
```

Nhà phát hành: Lang Nguyen Minh Luong
CHỮ KÝ KHÔNG HỢP LỆ HOẶC FILE BỊ SỬA!

Giải thích:

- Script đọc đúng vùng ByteRange và tính hash chính xác.
- Kết quả “KHÔNG HỢP LỆ” là hợp lý, do chữ ký dùng chuẩn PKCS#7 (CMS detached), không thẻ kiểm tra bằng RSA thuận.
- Khi mở file bằng **Adobe Acrobat Reader** → chữ ký hiển thị **VALID**, xác nhận tài liệu nguyên vẹn.

Ảnh minh họa:



III. KẾT QUẢ KIỂM THỬ

File kiểm thử	Kết quả	Ghi chú
original.pdf	Chưa ký	Dữ liệu gốc
signed.pdf	Chữ ký hợp lệ (Adobe Reader xác nhận không chỉnh sửa)	
tampered.pdf	File bị sửa	Phát hiện thay đổi sau khi ký

IV. KẾT LUẬN

Bài thực hành giúp sinh viên:

- Hiểu cấu trúc lưu chữ ký trong PDF theo chuẩn **PAdES/PDF 1.7**.

- Thực hành sinh khóa RSA, tạo chữ ký, nhúng và xác thực file.
- Hiểu sự khác biệt giữa xác thực RSA thuần và PKCS#7/CMS.
- Biết cách phát hiện chỉnh sửa nội dung sau khi ký.

Kết luận tổng quát:

File PDF đã được ký số đúng quy trình, chữ ký được nhúng hợp lệ, có thẻ xác minh bằng phần mềm tiêu chuẩn.

Script Python hoạt động tốt, phát hiện chính xác các thay đổi.

V. RỦI RO VÀ HƯỚNG MỞ RỘNG

Rủi ro	Giải thích / Cách khắc phục
Lộ khóa bí mật (private key)	Không dùng chung key thật, chỉ test với key sinh tạm
Sửa đổi sau khi ký	Được phát hiện qua ByteRange
Không có timestamp	
RFC3161	Có thẻ bổ sung TSA để xác thực thời gian
Không kiểm tra OCSP/CRL	Bổ sung xác minh CA chain & LTV
Padding Oracle / Replay	Dùng RSA-PSS hoặc ECDSA để tăng an toàn

VI. CÔNG CỤ SỬ DỤNG

Công cụ	Vai trò
Python 3.11 + VS Code	Môi trường lập trình
PyHanko	Ký PDF, nhúng chữ ký số
cryptography / OpenSSL	Sinh khóa RSA, tạo PKCS#7
PyPDF2	Đọc ghi nội dung PDF
Adobe Acrobat Reader	Xác minh chữ ký chuẩn PAdES

VII. TÀI LIỆU THAM KHẢO

1. ISO 32000-2:2017 – *PDF 2.0 Specification*
2. ETSI EN 319 142-1 – *PAdES Digital Signatures*

3. <https://pypi.org/project/pyHanko/>
4. <https://www.openssl.org/docs/>
5. Bài giảng “An toàn và Bảo mật Thông tin” – GV Đỗ Duy Cốp

Sinh viên thực hiện:

Lăng Nguyễn Minh Lượng

58KTPM

Đại học Kỹ thuật Công nghiệp – Đại học Thái Nguyên