TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP KHOA ĐIỆN TỬ

BỘ MÔN CÔNG NGHỆ THÔNG TIN

BÀI TẬP VỀ NHÀ CHỮ KÍ SỐ TRONG FILE PDF

GIẢNG VIÊN : Th.S ĐỖ DUY CỐP

LỚP HỌC PHẦN : K58.KTP01

HỌ TÊN SINH VIÊN : DƯƠNG THỊ LY

MSSV : K225480106045

BÁO CÁO BÀI TẬP VỀ NHÀ SỐ 2 MÔN AN TOÀN VÀ BẢO MẬT THÔNG TIN

(CHỮ KÝ SỐ TRÊN FILE PDF)

I. MÔ TẢ CHUNG

Báo cáo này trình bày việc nhúng và xác thực chữ ký số trong file PDF, theo chuẩn PDF 1.7 / PDF 2.0 và PAdES/ETSI EN 319 142.

- **Chuẩn PDF**: ISO 32000-2 (PDF 2.0) – định nghĩa cấu trúc document, incremental update, AcroForm, XObject, Content streams.
- **Chuẩn chữ ký**: PAdES (PDF Advanced Electronic Signature) – mở rộng CMS/PKCS#7 cho PDF, hỗ trợ LTV, DSS, timestamp RFC3161.
- **Công cụ sử dụng**: Python (PyPDF2, Endesive), OpenSSL cho PKCS#7, và script Python để ký & verify.

II. CẤU TRÚC PDF LIÊN QUAN CHỮ KÝ

1. Các object quan trọng

Object	**Vai trò**
Catalog	Root của PDF, trỏ tới Pages tree & AcroForm
Pages tree	Quản lý tất cả trang, chứa Page objects
Page object	Chứa /Resources, /Contents (text, image, XObject)
XObject	Hình ảnh, Form XObject

AcroForm	Quản lý interactive form, chứa Signature fields
Signature field (widget)	Vùng chứa chữ ký hiển thị
Signature dictionary (/Sig)	Thông tin chữ ký, /ByteRange, /Contents
/ByteRange	Xác định offset dữ liệu để tính hash
/Contents	Blob DER PKCS#7/CMS
Incremental updates	Cho phép ghi chữ ký mà không sửa file gốc

III. LƯU THÔNG TIN THỜI GIAN KÝ

Vị trí	Nội dung	Ghi chú
/M trong SigDict	Dạng text, D:YYYYMMDDHHmmSS+TZ	Không có giá trị pháp lý, chỉ metadata
Timestamp token RFC3161 (PKCS#7)	signingTime + token từ TSA	Có giá trị pháp lý, chống sửa đổi/replay
Document timestamp object (PAdES)	Dùng trong PAdES-LTV	Cho LTV, tích hợp DSS
DSS	Lưu timestamp, chứng thư xác minh	Cho phép xác thực lâu dài

Khác biệt: /M chỉ là metadata text, timestamp RFC3161 là token ký số hợp lệ từ TSA.

IV. QUY TRÌNH TẠO CHỮ KÝ PDF

- Chuẩn bị file original.pdf có khung chữ ký. Tạo SigField, dành vùng /Contents ~ 8 KB.
- Xác định /ByteRange, loại trừ vùng ký.
- Hash (SHA-256) phần được ký.
- Sinh PKCS#7 detached: messageDigest, signingTime, contentType, chứng chỉ. Ký RSA-2048, PKCS#1 v1.5.
- Ghi blob DER vào /Contents → file signed.pdf. (Tuỳ chọn) thêm timestamp và DSS.

V. XÁC THỰC CHỮ KÝ

1. Đọc Signature dictionary /Contents và /ByteRange.
2. Tách PKCS#7 blob, kiểm tra định dạng.
3. Tính hash vùng ByteRange, so sánh messageDigest.
4. Verify signature bằng public key trong certificate.
5. Kiểm tra chain → root CA.
6. Kiểm tra OCSP/CRL.
7. Kiểm tra timestamp token (RFC3161).
8. Phát hiện incremental update (detect tampering).

Script verify (Python):

```
from endesive.pdf import verify
log = verify.verify('signed.pdf')
print(log)
```

Demo: original.pdf, signed.pdf, tampered.pdf.

VI. RỦI RO & AN TOÀN

- Padding oracle attack: với PKCS#1 v1.5 → nên dùng PSS.
- Replay attack: timestamp hợp lệ chống replay.
- Key leak: private key phải lưu an toàn.
- SHA-1 weak: dùng SHA-256 trở lên.
- Incremental update: phát hiện chỉnh sửa ngoài chữ ký.

Lưu ý: Không dùng key thương mại, sinh key riêng trong repo.

VII. KẾT QUẢ

1. Các bước tiến hành

- Tạo CA (Certificate Authority – Cơ quan chứng thực gốc)

  openssl genrsa -out ca.key 2048 openssl req -x509 -new -key ca.key -out ca.crt -days 365 subj"/C=VN/ST=ThaiNguyen/L=TNUT/O=TNUTCA/CN=TNUT Root CA"
- Tạo khóa cá nhân và CSR (Yêu cầu ký chứng chỉ)

  openssl genrsa -out duongthily.key 2048 openssl req -new -key duongthily.key -out duongthily.csr -subj "/C=VN/ST=ThaiNguyen/L=TNUT/O=K58KTPM/OU=EbookCoach/CN=Dương Thị Ly/emailAddress=duongthily@tnut.edu.vn" Ký chứng chỉ cá nhân bằng CA (dùng file v3.ext) openssl x509 -req -in duongthily.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out duongthily.cer -days 365 -sha256 -extfile v3.ext
- Tạo file .pfx (Trao đổi thông tin cá nhân)

openssl pkcs12 -export -out duongthily.pfx -inkey duongthily.key -in duongthily.cer -certfile ca.crt -password pass:123456

- Đây là tập tin tổng hợp chứa khóa riêng + chứng chỉ cá nhân + chứng chỉ CA , dùng cho Windows, ký hiệu PDF, IIS, vv

2. Kết quả kiểm thử

Hình 1. Chạy file Sign_pdf.py

Hình 2. Chạy file tamper_pdf.py

Hình 3. Chạy file Verify_pdf.py

Giải thích:

- Signature verification = True → signature trên PDF vẫn hợp lệ (chữ ký số khớp với messageDigest).
- messageDigest equality = True → hash tính lại từ ByteRange giống messageDigest trong PKCS#7.
- certificate_present = False → trong blob PKCS#7 trong /Contents không tìm thấy certificate chain (hoặc verifier của bạn không thấy vì nó nằm ở nơi khác như DSS).

3. Xác thực độ tin cậy