BÀI TẬP SỐ 2

MÔN: AN TOÀN VÀ BẢO MẬT THÔNG TIN

Sinh viên: Vũ Bảo Khánh – MSSV: K225480106028

Lớp: K58KTPM

Nội dung: Tệp PDF này dùng để thử nghiệm quy trình tạo chữ ký số (8 bước) theo yêu cầu của đề bài môn An toàn và Bảo mật thông tin. Báo cáo mô tả cấu trúc PDF liên quan chữ ký, cách lưu thời gian ký, và các rủi ro bảo mật, dựa trên ISO 32000-1 và PAdES. Minh họa qua file original.pdf (gốc), signed.pdf (đã ký), tampered.pdf (bị chỉnh sửa).

1) Cấu trúc PDF liên quan chữ ký (Nghiên cứu)

Chữ ký số trong PDF được lưu dưới dạng các object trong cấu trúc PDF **và liên kết chặt chẽ thông qua** Catalog → AcroForm → Signature Field → Signature Dictionary**.**

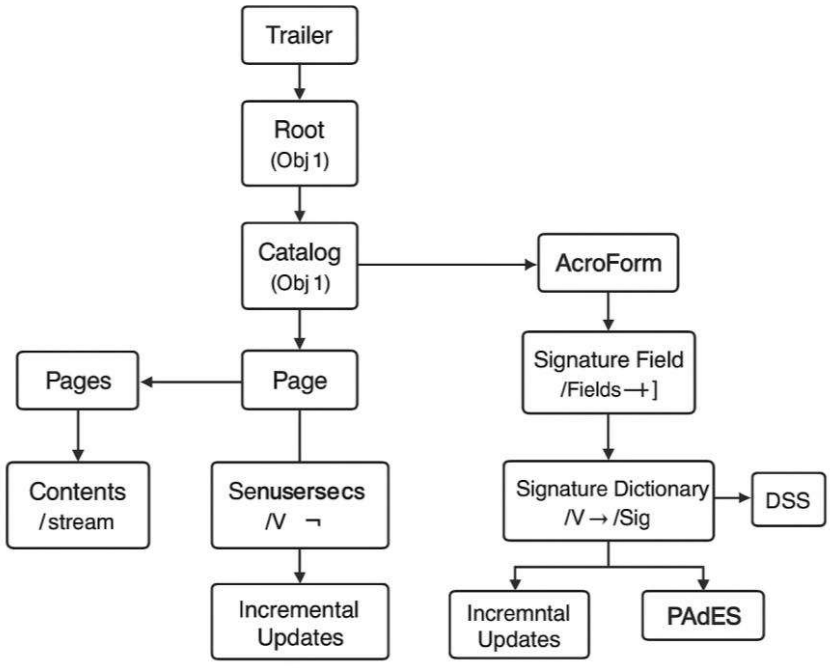
PDF cho phép ký theo dạng "incremental update"**: mỗi lần ký, file không bị ghi đè mà thêm một lớp cập nhật mới, đảm bảo** toàn vẹn và truy xuất được lịch sử ký**.**

Các thành phần chính:

Thành phần	Vai trò	Object refs / từ khóa
Catalog	Gốc của tài liệu, tham chiếu đến tất cả cấu trúc như Pages và AcroForm.	**/Root** trong trailer
Pages Tree	Cây phân cấp quản lý tất cả các trang của PDF.	**/Pages, /Kids, /Count**
Page Object	Đại diện cho từng trang, liên kết tới nội dung hiển thị.	**/Contents, /Resources**
Resources	Danh sách tài nguyên (phông, hình, XObject...) dùng trong trang.	**/Resources**
Content Streams	Chuỗi lệnh vẽ nội dung trang (văn bản, hình ảnh...).	**/Contents → stream**
XObject	Các đối tượng đồ họa hoặc hình ảnh tái sử dụng.	**/Xobject** trong **/Resources**
AcroForm	Biểu mẫu tương tác, chứa danh sách các trường (fields) — bao gồm trường chữ ký.	**/AcroForm**
Signature Field (Widget Annotation)	Trường biểu mẫu chứa chữ ký số (vị trí ký hiển thị).	**/FT /Sig, /V** trỏ đến **/Sig**
Signature Dictionary (/Sig)	Nơi lưu dữ liệu chữ ký (hash, người ký, thời gian...).	**/Type /Sig, /Filter, /SubFilter, /ByteRange, /Contents, /M**

/ByteRange	Chỉ định các đoạn byte trong file được bao phủ bởi chữ ký (phần đã ký và chưa ký).	**[start1 length1 start2 length2]**
/Contents	Chứa dữ liệu chữ ký số (thường là PKCS#7/CMS dạng hex).	**/Contents**
Incremental Updates	Cơ chế thêm phần ký mới mà không làm mất tính toàn vẹn các phần trước đó.	Mỗi lần ký thêm một **xref, trailer,** và **/Sig** mới
DSS (Document Security Store – PAdES)	Theo chuẩn PAdES, lưu metadata bảo mật: chứng thư, OCSP, CRL, timestamp... phục vụ xác minh lâu dài (LTV).	/DSS trong **Catalog hoặc SigDict**

Sơ đồ liên kết:



2) Thời gian ký được lưu ở đâu?

Các vị trí có thể lưu thông tin thời gian:

Vị trí	Mô tả	Đặc điểm
1. /M trong Signature Dictionary (/Sig)	**Là** chuỗi text **lưu thời gian ký mà phần mềm ký ghi vào. Ví dụ: /M (D:20251030T103000+07'00').**	**- Chỉ mang tính** thông tin hiển thị.- Không được bảo vệ bởi chữ ký.**- Có thể bị chỉnh sửa mà không làm sai chữ ký (nên** không có giá trị pháp lý**).**
2. Timestamp Token (RFC 3161) trong chữ ký PKCS#7	**Là** thuộc tính (attribute) **trong cấu trúc CMS/PKCS#7 – cụ thể là** timeStampToken**. Được cấp bởi** Time Stamping Authority (TSA)**.**	**- Được** ký số bởi TSA **nên có** giá trị pháp lý.**- Dùng để chứng minh chữ ký được tạo** tại hoặc trước **thời điểm timestamp.**
3. Document Timestamp (PAdES)	**Là** một loại chữ ký đặc biệt**, không gắn với người ký mà với toàn bộ tài liệu.**	**- Bảo vệ toàn bộ file tại một thời điểm nhất định.- Dùng trong** PAdES-LTV **để đảm bảo tính tồn tại lâu dài của chữ ký.**
4. DSS (Document Security Store)	**Theo chuẩn** PAdES **– lưu trữ các thông tin phục vụ xác**	**- Có thể chứa** timestamp bổ sung **(cho chữ ký hoặc**

	minh lâu dài như OCSP, CRL, timestamp**.**	**toàn tài liệu).- Dùng để xác minh về sau ngay cả khi TSA/CA gốc đã hết hạn.**

Sự khác biệt giữa /M và Timestamp (RFC 3161):

Tiêu chí	**/M (Signature Dictionary)**	**Timestamp (RFC 3161 trong PKCS#7)**
Nguồn gốc	Do phần mềm ký (signing application) tự ghi.	Do TSA (Time Stamping Authority) phát hành và ký.
Định dạng	Chuỗi text PDF kiểu /M (D:YYYYMMDDHHmmss+TZ)	Cấu trúc nhị phân ASN.1 trong gói PKCS#7 (timeStampToken)
Được bảo vệ bởi chữ ký?	Không (nằm ngoài vùng băm /ByteRange)	Có (nằm trong vùng được ký bởi TSA)
Giá trị pháp lý	Tham khảo (chỉ mô tả lúc phần mềm tạo chữ ký)	Có giá trị chứng thực thời gian ký (theo chuẩn RFC 3161)
Mục đích chính	Hiển thị trong giao diện xem chữ ký	Chứng minh thời điểm ký số là có thực và hợp lệ

3) Rủi ro bảo mật

Rủi ro chính (tóm tắt):

1. **Tamper nội dung (/Contents hoặc /ByteRange)**
    - Mô tả: sửa text/hình trên trang sau khi ký (hoặc sửa trực tiếp /Contents/objects), khiến nội dung hiển thị khác so với vùng được băm.
    - Hậu quả: chữ ký báo *invalid* (mất integrity) — nhưng nếu tấn công tinh vi (object injection / incremental abuse) có thể che dấu.
    - Phát hiện: verify kiểm tra ByteRange vs file bytes → thất bại nếu băm khác.
    - Tham khảo: chuẩn kiểm tra incremental / ByteRange theo PAdES/ETSI.
2. **Replay attack (ký lại SigDict với timestamp cũ bằng incremental updates)**
    - Mô tả: dùng incremental update để thêm một SigDict mới hoặc sửa metadata thời gian, dùng timestamp cũ để chứng minh "ký trước" khi thực tế không phải vậy.
    - Hậu quả: chối bỏ thời gian, giả tạo lịch sử ký.
    - Giảm rủi ro: bắt buộc *RFC3161 timestamp token* từ TSA và lưu token/validation data vào DSS (PAdES-LTV). RFC3161 mô tả token & cách verif.
3. **Cert revocation không được kiểm tra (CRL/OCSP)**
    - Mô tả: chứng thư signer đã bị thu hồi/expire nhưng verifier không kiểm tra OCSP/CRL — hoặc không có dữ liệu validate offline.
    - Hậu quả: chữ ký "hợp lệ" về mặt cryptography nhưng thực tế signer đã bị thu hồi → rủi ro pháp lý.
    - Giảm rủi ro: nhúng CRL/OCSP responses vào **DSS** để cho phép xác minh offline/LTV.

4. **Lộ private key / side-channel**
    - Mô tả: private key (file .pem/.pfx) bị leak hoặc tấn công side-channel.
    - Hậu quả: forge chữ ký (ký giả mạo).
    - Giảm rủi ro: HSM / smartcard, khóa truy cập chặt, offline key usage policy.
5. **Incremental updates lạm dụng / object injection**
    - Mô tả: thêm nhiều lớp update, lớp sau che lớp trước (content overlay, form filling) — viewer yếu/kém có thể hiển thị lớp cuối mà không cảnh báo thay đổi lớp trước.
    - Hậu quả: thay đổi hiển thị mà chữ ký trên lớp cũ vẫn được coi là "valid" nếu verifier không kiểm tra modification level.
    - Phát hiện & giảm rủi ro: dùng công cụ kiểm tra modification_level / incremental diff (ví dụ pyHanko có phân tích incremental updates, cho biết modification_level).