

# File PDF đã bị chỉnh sửa

Chữ ký số không còn hợp lệ.

## BÁO CÁO

**Môn:** An toàn và Bảo mật thông tin

**Chủ đề:** Chữ ký số trong file PDF

**Giảng viên:** ThS. Đỗ Duy Cốp

**Sinh viên thực hiện:** Nguyễn Tiến Thắng

**Lớp:** 58KTPM

**Thời điểm nộp:** 31/10/2025

### I. MÔ TẢ CHUNG

Báo cáo phân tích và hiện thực việc nhúng, lưu trữ và xác thực chữ ký số trong file PDF. Mục tiêu: trình bày cấu trúc PDF liên quan chữ ký, các vị trí lưu thông tin thời gian ký, trình tự kỹ thuật để tạo/chèn chữ ký PKCS#7 vào PDF (đã có private RSA), và quy trình xác thực chữ ký trên PDF đã ký. Ứng dụng tham chiếu: PDF 1.7 / PDF 2.0 (ISO 32000-2) và PAdES (ETSI EN 319 142).

**Công cụ thực thi đề xuất:** OpenSSL (CA/PKI), iText7 hoặc BouncyCastle (Java/.NET), PyPDF2/pikepdf/reportlab (Python) để thao tác PDF, và openssl/rfc3161token để lấy timestamp.

### II. PHÂN TÍCH KỸ THUẬT

1) Cấu trúc PDF liên quan chữ ký (tóm tắt)

**Catalog:** điểm vào tài liệu. Thông qua /AcroForm có thể truy cập các trường chữ ký.

**Pages tree:** cây trang. Mỗi **Page object** chứa /Resources, /Contents (content stream) và thông tin vị trí.

**Resources:** fonts, XObject, v.v.

**Content streams:** luồng hiển thị nội dung trang.

**AcroForm:** chứa form-level fields; chữ ký thường là SigField (field type /Sig).

**Signature field (widget):** object trong AcroForm đại diện cho vị trí hiển thị chữ ký; liên kết tới Signature dictionary.

**Signature dictionary (/Sig):** chứa các khóa quan trọng như /Type, /Filter, /SubFilter, /Contents, /ByteRange, /M, /Name.

**/ByteRange:** chỉ định các offset trong file PDF để tính băm (loại trừ phần /Contents đã dành sẵn).

**/Contents:** vùng lưu blob chữ ký PKCS#7 (DER, thường encode hex hoặc binary), kích thước cố định (ví dụ reserve 8192 bytes).

[!] CẢNH BÁO: Đây KHÔNG PHẢI là bản PDF gốc đã ký.

**Incremental updates:** cơ chế viết thêm (append) ghi chữ ký mà không làm thay đổi nội dung byte gốc trước đó — quan trọng để phát hiện sửa đổi.

**DSS (Document Security Store)** trong PAdES: vùng lưu chứng chỉ, OCSP/CRL, VRI để hỗ trợ LTV (Long Term Validation).

Sơ đồ object (minh họa)

Catalog → Pages → Page → /Contents

Catalog → /AcroForm → SigField → SigDict (/Contents, /ByteRange, /M)

Các object refs quan trọng: Catalog (entry point), AcroForm (form-level), SigField (widget), SigDict (signature metadata), ByteRange (hash boundaries), Contents (PKCS#7 blob), DSS (PAdES store).

2) Thời gian ký được lưu ở đâu?

**/M trong Signature dictionary:** trường text chứa thời điểm ký (ví dụ D:20251024...Z). *Không có giá trị pháp lý mạnh* vì dễ bị chỉnh sửa (nằm trong phần khả dĩ bị appended nếu attacker tự thêm incremental update).

**Timestamp token (RFC 3161):** nằm trong PKCS#7 như attribute timeStampToken. Đây là bằng chứng của bên thứ ba (TSA) và mang tính pháp lý hơn vì do TSA đóng dấu bằng khóa riêng của họ.

**Document timestamp object (PAdES):** PAdES định nghĩa cách nhúng document-level timestamp để chứng thực trạng thái tài liệu tại một thời điểm.

**DSS (Document Security Store):** có thể lưu timestamp và dữ liệu xác minh kèm OCSP/CRL để hỗ trợ LTV.

**Khác biệt chính:** /M là metadata nằm trong PDF, có thể bị giả mạo; token RFC3161 là bằng chứng độc lập do TSA ký, khó giả mạo nếu TSA được tin tưởng.

### III. QUY TRÌNH TẠO VÀ LƯU CHỮ KÝ TRONG PDF (KHI ĐÃ CÓ PRIVATE RSA)

**Mục tiêu:** Tạo chữ ký PKCS#7 (detached/CAdES), nhúng vào /Contents bằng incremental update, tuân thủ PAdES nếu cần LTV.

Tóm tắt bước kỹ thuật (phiên bản thực thi)

**Chuẩn bị file PDF gốc** — original.pdf (không thay đổi nội dung gốc sau khi bắt đầu).

**Tạo Signature field (AcroForm)** — thêm field widget và reserve vùng trong /Contents (ví dụ 8192 bytes): viết </Contents <0000...>> kích thước cố định.

[!] CẢNH BÁO: Đây KHÔNG PHẢI là bản PDF gốc đã ký.

**Xác định /ByteRange** — phần offset loại trừ vùng /Contents. Thông thường: /ByteRange [0 <off1> <off2> <off3>].

**Tính hash** — đọc phần byte theo ByteRange, hash bằng SHA-256 hoặc SHA-512.

**Tạo PKCS#7/CMS (detached) hoặc CAdES:**

Bao gồm attribute: messageDigest (hash), signingTime, contentType.

Chèn certificate chain (end-entity → intermediate → ...).

(Tùy chọn) Gửi digest lên TSA để nhận RFC3161 timeStampToken và thêm vào PKCS#7.

Dùng RSA padding: PKCS#1 v1.5 hoặc RSA-PSS (khuyến cáo RSA-PSS nếu hệ thống hỗ trợ).

**Chèn blob DER PKCS#7 vào /Contents** — đảm bảo kích thước blob  $\leq$  vùng reserve; nếu nhỏ hơn, pad với 6. **\*\*Chèn blob DER PKCS#7 vào /Contents\*\*** — đảm bảo kích thước blob  $\leq$  vùng reserve; nếu nhỏ hơn, pad với `.

**Ghi incremental update** — append phần cập nhật vào cuối file PDF, cập nhật xref và trailer mới.

**(LTV) Cập nhật DSS** — thêm chứng chỉ, OCSP/CRL, và VRI cho phép xác thực dài hạn.

Các thông số quan trọng cần nêu trong báo cáo

**Hash alg:** SHA-256 hoặc mạnh hơn.

**RSA key size:**  $\geq$  2048-bit (2048, 3072, 4096).

**Padding:** PKCS#1 v1.5 hoặc RSA-PSS (nêu ưu/nhược).

**Vị trí lưu trong PKCS#7:** certificate chain trong certs set; messageDigest trong signedAttributes; timeStampToken trong unsignedAttributes (RFC3161 token).

Ví dụ flow OpenSSL (tóm tắt)

Tạo digest file:

openssl dgst -sha256 -binary -out digest.bin data\_to\_sign.bin

Tạo PKCS#7 (detached signature):

[!] CẢNH BÁO: Đây KHÔNG PHẢI là bản PDF gốc đã ký.

`openssl cms -sign -binary -in data_to_sign.bin -signer cert.pem -inkey key.pem -outform DER -nodetach -out sig.der -certfile chain.pem`

(TSA) Đặt timestamp (gửi digest tới TSA) và chèn timeStampToken vào unsignedAttributes.

#### IV. QUY TRÌNH XÁC THỰC CHỮ KÝ TRONG PDF ĐÃ KÝ

**Đọc Signature dictionary** — trích /Contents (blob PKCS#7) và /ByteRange.

**Tách PKCS#7** — kiểm tra cấu trúc DER/PEM, xác định signedAttributes/unsignedAttributes.

**Tính hash** — tính hash theo ByteRange và so sánh với messageDigest trong signedAttributes.

**Verify signature** — xác minh chữ ký bằng public key trong chứng chỉ (kiểm tra signedAttributes digest được ký đúng bằng khóa private).

**Kiểm tra chain** → **root trusted CA** — xây dựng chain từ cert trong PKCS#7; kiểm tra mỗi cert đến root uy tín.

**Kiểm tra OCSP/CRL** — nếu có OCSP/CRL trong DSS hoặc trong signed data, kiểm tra trạng thái revocation.

**Kiểm tra timestamp token (RFC3161)** — xác minh token bằng public key của TSA; token chỉ ra thời gian tồn tại chữ ký.

**Kiểm tra incremental update** — nếu có incremental updates sau chữ ký, xác định thay đổi → báo tampered.

**Ghi log kiểm thử:** cần in ra các bước: ByteRange values, digest hex, messageDigest from PKCS#7, verification result, chain path, OCSP/CRL responses, timestamp token validity.

#### V. RỦI RO BẢO MẬT & BIỆN PHÁP GIẢM THIỂU

Rủi ro chính

**Lộ private key:** phá vỡ toàn bộ tính toàn vẹn/không thể phủ nhận.

**Padding oracle / side-channel:** tấn công vào thuật toán chữ ký (đặc biệt PKCS#1 v1.5).

**Replay / timestamp spoofing:** sử dụng timestamp giả nếu TSA không tin cậy.

**Tampering incremental update:** kẻ tấn công thêm nội dung sau chữ ký ở phần incremental update hoặc thay thế xref.

Biện pháp giảm thiểu

**Bảo quản private key** trên HSM hoặc module an toàn (PKCS#11).

**Dùng RSA-PSS** nơi khả dụng, giảm rủi ro padding oracle.

**Sử dụng RFC3161 TSA** đáng tin cậy cho timestamp.

**Lưu chứng chỉ & OCSP/CRL vào DSS** để hỗ trợ LTV.

**Kiểm tra incremental updates** trong quá trình verify để phát hiện sửa đổi.

[!] CẢNH BÁO: Đây KHÔNG PHẢI là bản PDF gốc đã ký.

Đã chỉnh sửa ngày: 31/10/2025 23:28:11