Een ondertekend document, en hoe dat ging

Dit bestand bevat een digitale handtekening. Die dekt elke byte van het bestand behalve de plek waar ze zelf staat: één gewijzigde byte maakt haar ongeldig, en leessoftware zegt dat.

Het certificaat is een demonstratiecertificaat: het is voor dit voorbeeld gemaakt, en het noemt geen persoon of bedrijf. De handtekening bewijst dat het bestand niet veranderd is; ze bewijst niets over wie tekende, want niemand staat in voor dit certificaat.



Hoe het ondertekend is

1. Sleutel en certificaat zijn één keer gemaakt, in de hoofdmap van de repository:

```
openssl req -x509 -newkey rsa:2048 -nodes -days 3650
  -subj "/CN=hqf-pdf demonstration signer"
  -keyout keys/demonstration_signer.key -out keys/demonstration_signer.crt
```

2. De bibliotheek krijgt het certificaat als DER, dat dit schrijft:

```
openssl x509 -in keys/demonstration_signer.crt -outform DER
```

3. De bibliotheek laat ruimte in het bestand voor de handtekening, berekent de hash van elke byte eromheen en geeft het programma de attributen om te ondertekenen. De sleutel ondertekent ze, en bereikt de bibliotheek nooit:

```
openssl dgst -sha256 -sign keys/demonstration_signer.key
```

4. De bibliotheek verpakt die handtekening en het certificaat in de structuur die een PAdES-handtekening bevat, en schrijft die in de vrijgelaten ruimte.

Controleren zonder deze bibliotheek

De /ByteRange noemt de twee bytereeksen die de handtekening dekt, en /Contents bevat de structuur ertussen, hexadecimaal. Voeg de twee reeksen samen tot covered.bin, decodeer de structuur naar signature.der, en voer uit:

```
openssl cms -verify -binary -inform DER -in signature.der -content covered.bin
  -CAfile keys/demonstration_signer.crt -purpose any
```