A signed document, and how it was signed

This file carries a digital signature. It covers every byte of the file but the place it is written in: one byte changed anywhere breaks it, and reading software says so.

The certificate is a demonstration one: it was made for this example, and it names no person and no company. The signature proves the file has not changed; it proves nothing about who signed it, since nobody vouches for this certificate.



How it was signed

1. The key and its certificate were made once, from the repository's root:

```
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. The library is handed the certificate as DER, which this writes:

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

3. The library leaves room in the file for the signature, takes the digest of every byte around it, and hands the program the attributes to sign. The key signs them, and never reaches the library:

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

4. The library wraps that signature and the certificate into the structure a PAdES signature carries, and writes it into the room it left.

Checking it without this library

The file's /ByteRange names the two runs of bytes the signature covers, and /Contents holds the structure between them, in hexadecimal. Join the two runs into covered.bin, decode the structure into signature.der, and run:

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