How to verify an instantproof evidence package

For lawyers, court experts and opposing parties. Every check below uses standard public tools and works without contacting instantproof.

> **What you can establish.** That each file is identical to what instantproof captured; that the record was sealed by THINK DATA with a qualified electronic seal; and that each principal file existed, unchanged, at the time stated by an independent qualified time-stamping authority (InfoCert S.p.A.). It does not establish the truth of the captured content or who authored it.

The files you need

File	Role
`certificate-<number>.pdf`	Human-readable certificate, sealed with a qualified electronic seal (PAdES).
`manifest.json`	The record: hash value (SHA-256) of every captured file, times, addresses, software versions.
`manifest.eseal.p7s`, `eseal-chain.pem`	Qualified electronic seal over `manifest.json`, and the issuer's certificate chain (Namirial S.p.A.).
`manifest.sig`, `signing-public-key.pem`	Second, independent signature over `manifest.json` (Ed25519).
`<file>.tsr` (and `.tsq`)	RFC 3161 time-stamp token for the file of the same name, e.g. `recording.mp4.tsr` covers `recording.mp4`.
`network/…`	Large network captures (complete bundle only), listed with their hash values in `manifest.json` under `notary.externalFiles`.
`US-court-declaration-FRE-902.pdf`	Signed declaration under US Federal Rules of Evidence 902(13) and 902(14). Its hash is in `manifest.json` under `legalDeclarations`.

Files are in the `evidence` folder of the downloaded bundle. Commands below are run from that folder on macOS or Linux (Windows: use WSL). OpenSSL 3 is required for step 3.

Step 1 · The certificate PDF (no tools needed)

1. Open the certificate PDF in Adobe Acrobat or Acrobat Reader and open the Signatures panel.
2. Expected: **signed by THINK DATA, signature valid, document not modified**. The signer chains to "Namirial EU Qualified CA", which Acrobat trusts through the EU Trusted List (on by default: Preferences, Trust Manager, "Load trusted certificates from an Adobe EUTL server").

Command-line alternative (Poppler): `pdfsig certificate-<number>.pdf` should report "Signature is Valid". Poppler does not load the EU Trusted List, so it may add "issuer isn't trusted"; establish trust with step 2 instead.

Step 2 · The qualified seal on the manifest

```
openssl cms -verify -inform DER -in manifest.eseal.p7s -content manifest.json \
  -binary -CAfile eseal-chain.pem -purpose any -out /dev/null
```

Expected: **CMS Verification successful**. To confirm the issuer independently, compare the root of `eseal-chain.pem` ("Namirial EU Qualified CA", Italy) with the EU Trusted List browser at eidas.ec.europa.eu.

Step 3 · The Ed25519 signature on the manifest

```
base64 -d -i manifest.sig > manifest.sig.bin
openssl pkeyutl -verify -pubin -inkey signing-public-key.pem -rawin \
  -in manifest.json -sigfile manifest.sig.bin
```

Expected: **Signature Verified Successfully**. (On Linux use `base64 -d manifest.sig` without `-i`.)

Step 4 · Every file matches the manifest

```
shasum -a 256 recording.mp4          # Linux: sha256sum recording.mp4
```

Compare the result with the value in `manifest.json`: under `video.sha256` for a recorded session, `file.sha256` for an uploaded file, and in the `sha256` fields for other entries such as `notary.externalFiles` and `legalDeclarations`. Identical values mean identical files; a change of one bit gives a different value.

Step 5 · The qualified time-stamps

Read the time and the authority:

```
openssl ts -reply -in recording.mp4.tsr -text | grep -E "Time stamp|TSA"
```

Verify that the token covers the file (repeat for each `.tsr`):

```
openssl ts -reply -in recording.mp4.tsr -token_out -out tok.der
openssl pkcs7 -inform DER -in tok.der -print_certs -out tsa.pem
openssl ts -verify -data recording.mp4 -in recording.mp4.tsr \
  -CAfile tsa.pem -untrusted tsa.pem -partial_chain
```

Expected: **Verification: OK**, with "InfoCert S.p.A." as the authority. The token embeds the time-stamping unit's certificate; its issuer, "InfoCert Time Stamping Authority 3", can be confirmed as a qualified time-stamping service on the EU Trusted List.

Step 6 · Network captures (complete bundle)

Hash each file in `network/` as in step 4 and compare with `notary.externalFiles`. The witness capture also carries its own qualified seal:

```
openssl cms -verify -inform DER -in notary-capture.pcap.eseal.p7s \
  -content network/notary-capture.pcap -binary -CAfile eseal-chain.pem \
  -purpose any -out /dev/null
```

The TLS session keys in `sslkeys.log` let tools such as Wireshark decrypt `network/capture.pcapng` and show the server certificate and the exact content the website transmitted. A decrypted summary is

included as `tls-decrypted.txt`.

Notes

- **Packages issued before 1 October 2026** stored `manifest.json` with one extra line break at the end. For those, remove the final byte before steps 2 and 3, for example `head -c -1 manifest.json > m.json` (macOS: `perl -pe 'chomp if eof' manifest.json > m.json`) and use `m.json`.
- **Trial certificates** are examples and carry no qualified time-stamp.
- **US proceedings:** the party offering the evidence under FRE 902(11)/(12)–(14) must give the adverse party reasonable written notice before trial and make the record and declaration available for inspection.
- Questions about a specific package: legal@instantproof.legal.