E14. Image-provenance examination

Ilja Krasnjanski
24 October 2026 · Edited evidence report
Historical evidence through 24 July 2026; later engineering checks identified separately.

This report synthesizes the cited case records. Exhibits distinguish retained artifact excerpts from secondary analytical records and claims whose original inputs are unavailable. Excerpts may be redacted or reformatted; extraction methods and limitations are stated locally. Fingerprints identify inputs, not the truth of their claims. Bracketed S labels identify the analysis records listed at the end.

Examined material

The ECHOES poster, identified in the case as `Post.png`, was examined through PNG metadata and embedded provenance structures. The recorded method used metadata extraction and manual JUMBF, CBOR and certificate parsing. This report reviews the saved examination; it does not perform a new cryptographic validation. [S1]

Recorded provenance fields

The record identifies a C2PA structure in the PNG caBX chunk. Reported values include claim generator ChatGPT, software agent GPT-4o, source type `trainedAlgorithmicMedia`, and certificate subject fields referring to an OpenAI/Sora signing pipeline. Creation, conversion and ingredient-opening actions are reported across two manifests. These are descriptions of embedded assertions, without independent authentication established by this exhibit set. [S1]

The later manifest (B), which references manifest A as an ingredient, contains an empty redacted-assertions array; that field is absent from A. The note reports no prompt and no user identifier. Absence of those fields is not evidence that they were deliberately stripped. The embedded thumbnail was described as a lower-resolution version of the same poster; no appended payload was found after the PNG end marker in the recorded examination.

Timestamp and signature limitations

The recorded leaf certificate has a validity interval from 15 April 2025 to 15 April 2026. The analysis also reports no trusted timestamp token. Certificate validity alone therefore cannot establish the actual generation or signing date. The earlier inference that the image must have been generated within that interval is not adopted here. [S1]

The source states that the signature validated, but no independent verifier output, certificate-status check or trust-list result is retained. An embedded ingredient validation assertion about manifest A is not independent verification and does not validate manifest B. Provider identifiers and manifest instance values are omitted. Readers should distinguish a reported validation result from independently reproduced verification.

Assessment

The recorded metadata supports use of an AI-image generation pipeline in preparation of the lure artwork. It does not identify the person who generated, obtained or deployed the image. Other portraits lacking C2PA metadata could have been re-encoded, but absence alone does not prove intentional provenance removal. Image metadata also cannot establish nationality, account ownership or responsibility for the intrusion.

Exhibits

All exhibits were produced by reading bytes only. The two helper scripts (`tools/pngchunks.py`, `tools/c2pa_static.py`, kept beside this file) walk PNG chunk headers and JUMBF boxes, decode CBOR with a small built-in decoder, and parse certificate DER with the `cryptography` library. They perform no signature verification, no network access and no image decoding. The image was not uploaded anywhere. Provider manifest identifiers, instance IDs and the leaf-certificate serial are redacted, consistent with E14's statement that they are omitted.

Exhibit E14-1. Poster chunk layout: C2PA chunk present, nothing after IEND

Artifact: ECHOES poster `Post.png` · 3698494 bytes · SHA-256
`d213b4184b93ccf59449237b071ac2e41ff71b83213b2e338917b26696a7b520`
Location: PNG chunk headers, byte offsets 8–3698494 (whole file)
Extraction: `python3 -I tools/pngchunks.py Post.png`
Supports: "a C2PA structure in the PNG caBX chunk"; "no appended payload was found after the PNG end marker"

```
file_size=3698494
offset    type  length
8         IHDR  13
33        caBX  103419
103464    IDAT  3595006
3698482   IEND  0
iend_end_offset=3698494 trailing_bytes_after_IEND=0
```

The `caBX` chunk (the C2PA chunk type for PNG) sits directly after IHDR and holds 103,419 bytes; IEND ends exactly at end-of-file, so zero bytes follow it. There is no tEXt/iTXt/XMP chunk.

Exhibit E14-2. Character portraits: no caBX chunk

Artifact: portrait `Lyra.png` · 862883 bytes · SHA-256
`f7d17f0c0f035b67c96b8f37a4e55e27141df0c4d69b534a309191f398a66a47`
Artifact: portrait `Solace.png` · 605016 bytes · SHA-256
`6e5e8ecf4b38580ed494556e0cd12150790ce0bf790b3da8588df91298aeadce`
Artifact: portrait `Kael.png` · 816478 bytes · SHA-256
`8a0d38a6089e5c5d276820bb13089986798f5ca3e3136cd028a41e37da1892c0`
Location: PNG chunk headers (whole files)
Extraction: `for f in Lyra.png Solace.png Kael.png; do python3 -I tools/pngchunks.py "$f"; done`
Supports: "Other portraits lacking C2PA metadata could have been re-encoded"

```
== Lyra.png
8         IHDR  13
33        pHYs  9
54        gAMA  4
70        cHRM  32
114       IDAT  862745
862871    IEND  0
iend_end_offset=862883 trailing_bytes_after_IEND=0
== Solace.png
8         IHDR  13
33        pHYs  9
54        gAMA  4
70        cHRM  32
114       IDAT  604878
605004    IEND  0
iend_end_offset=605016 trailing_bytes_after_IEND=0
== Kael.png
8         IHDR  13
33        pHYs  9
54        gAMA  4
70        cHRM  32
114       IDAT  816340
816466    IEND  0
iend_end_offset=816478 trailing_bytes_after_IEND=0
```

The `file_size`/header lines of each run are omitted. All three share an identical pHYs/gAMA/cHRM profile and carry no caBX chunk. As E14 says, this is consistent with re-encoding but does not by itself show deliberate provenance removal.

Exhibit E14-3. JUMBF manifest-store layout inside caBX

Artifact: ECHOES poster `Post.png` · 3698494 bytes · SHA-256
`d213b4184b93ccf59449237b071ac2e41ff71b83213b2e338917b26696a7b520`
Location: caBX chunk data, absolute byte offsets 41–103460
Extraction: `python3 -I tools/c2pa_static.py Post.png labels`
Supports: "Creation, conversion and ingredient-opening actions are reported across two manifests";
embedded thumbnail

```
    41 jumb  c2pa
    79   jumb  urn:c2pa:[REDACTED: manifest id A]
   158     jumb  c2pa.assertions
   207       jumb  c2pa.actions.v2
   256         cbor  len=180
   436       jumb  c2pa.hash.data
   484         cbor  len=123
   607     jumb  c2pa.claim.v2
```

```
    654       cbor  len=447
   1101     jumb  c2pa.signature
   1149       cbor  len=12999
  14148   jumb  urn:c2pa:[REDACTED: manifest id B]
  14227     jumb  c2pa.assertions
  14276       jumb  c2pa.thumbnail.ingredient
  14359         bfdb  len=20
  14379         bidb  len=73443
  87822       jumb  c2pa.ingredient.v3
  87874         cbor  len=1458
  89332       jumb  c2pa.actions.v2
  89381         cbor  len=155
  89536       jumb  c2pa.hash.data
  89584         cbor  len=125
  89709     jumb  c2pa.claim.v2
  89756       cbor  len=657
  90413     jumb  c2pa.signature
  90461       cbor  len=12999
```

Two manifests, each with its own claim and signature. Manifest B (last in the store, hence the active manifest under C2PA ordering) carries the thumbnail and an ingredient assertion. The thumbnail payload (bidb box minus its 8-byte header, 73,435 bytes, `ffd8ffe0…` JPEG, 333×500 px) hashes to `fee5e557f82c8a500818b917f4861e781eee664315d27e0d99f1683957f13e57`, identical to the case's extracted `c2pa_thumb_0.jpg`; the poster IHDR is 1024×1536, the same 2:3 aspect.

Exhibit E14-4. Decoded claim and action assertions

Artifact: ECHOES poster `Post.png` · 3698494 bytes · SHA-256 `d213b4184b93ccf59449237b071ac2e41ff71b83213b2e338917b26696a7b520`
Location: CBOR boxes at offsets 256 (actions A), 654 (claim A), 89381 (actions B), 89756 (claim B)
Extraction: `python3 -I tools/c2pa_static.py Post.png claims`
Supports: claim generator ChatGPT, software agent GPT-4o, `trainedAlgorithmicMedia`, actions, empty redacted-assertions array

```
## c2pa/urn:c2pa:[REDACTED: id A]/c2pa.assertions/c2pa.actions.v2
   "action": "c2pa.created",
   "softwareAgent": {
    "name": "GPT-4o"
   },
   "digitalSourceType": "hxxp://cv[.]iptc[.]org/newscodes/digitalsourcetype/\
trainedAlgorithmicMedia"
   "action": "c2pa.converted"
## c2pa/urn:c2pa:[REDACTED: id A]/c2pa.claim.v2
 "instanceID": "xmp:iid:[REDACTED: provider instance id]",
 "claim_generator_info": {
  "name": "ChatGPT",
  "org.contentauth.c2pa_rs": "0.67.1"
 "dc:title": "image.png",
## c2pa/urn:c2pa:[REDACTED: id B]/c2pa.assertions/c2pa.actions.v2
   "action": "c2pa.opened",
     "url": "self#jumbf=c2pa.assertions/c2pa.ingredient.v3",
## c2pa/urn:c2pa:[REDACTED: id B]/c2pa.claim.v2
 "claim_generator_info": {
  "name": "ChatGPT",
  "org.contentauth.c2pa_rs": "0.67.1"
 "dc:title": "image.png",
 "redacted_assertions": [],
 "alg": "sha256"
```

Selected lines from the JSON rendering of the CBOR (structural braces and hash fields between them omitted; one line hard-wrapped with `\`; the IPTC vocabulary URL defanged). Note that `redacted_assertions: []` appears only in claim B; claim A has no such field. Neither claim, nor any other assertion, contains a prompt or user identifier field.

Exhibit E14-5. Signing certificate chain and absence of a timestamp token

Artifact: ECHOES poster `Post.png` · 3698494 bytes · SHA-256 `d213b4184b93ccf59449237b071ac2e41ff71b83213b2e338917b26696a7b520`
Location: COSE_Sign1 in c2pa.signature CBOR at offsets 1149 (A) and 90461 (B); x5chain = header 33
Extraction: `python3 -I tools/c2pa_static.py Post.png cert; python3 -I tools/c2pa_static.py Post.png protected`
Supports: certificate subject referring to OpenAI/Sora; validity 15 Apr 2025 – 15 Apr 2026; "no trusted timestamp token"

```
## c2pa/urn:c2pa:[REDACTED: id A]/c2pa.signature protected-alg -7
cert[0] subject=CN=Truepic Lens CLI in Sora,OU=Sora,O=OpenAI,C=US
```

```
cert[0] issuer=C=US,O=Truepic,OU=Lens,CN=WebClaimSigningCA
cert[0] serial=[REDACTED: leaf certificate serial]
cert[0] not_before=2025-04-15T15:09:05+00:00
cert[0] not_after=2026-04-15T15:09:04+00:00
cert[1] subject=C=US,O=Truepic,OU=Lens,CN=WebClaimSigningCA
cert[1] issuer=C=US,O=Truepic,OU=Lens,CN=RootCA
cert[1] not_before=2021-12-09T20:39:46+00:00
cert[1] not_after=2026-12-08T20:39:45+00:00
unprotected-keys ['pad']
## c2pa/urn:c2pa:[REDACTED: id A]/c2pa.signature tag 18 protected-keys ['1', '33'] \
unprotected-keys ['pad']
## c2pa/urn:c2pa:[REDACTED: id B]/c2pa.signature tag 18 protected-keys ['1', '33'] \
unprotected-keys ['pad']
```

Manifest B's chain is identical (same leaf, same validity; its cert lines and the intermediate serial are omitted here). Algorithm −7 is ES256. The only COSE header keys are `alg` (1), `x5chain` (33) and padding: there is no `sigTst`/timestamp header, so nothing in the file dates the signing event. These are decoded fields only; no chain or revocation validation was run.

Exhibit E14-6. The "validated" result is a record embedded at signing, not a fresh check

Artifact: ECHOES poster `Post.png` · 3698494 bytes · SHA-256
`d213b4184b93ccf59449237b071ac2e41ff71b83213b2e338917b26696a7b520`
Location: c2pa.ingredient.v3 CBOR in manifest B, offset 87874, 1458 bytes
Extraction: `python3 -I tools/c2pa_static.py Post.png ingredient`
Supports: "The source states that the signature validated, but this report does not supply a fresh verifier output"

```
  "relationship": "parentOf",
  "dc:title": "image.png",
  "dc:format": "png",
  "validationResults": {
   "activeManifest": {
    "success": [
      "code": "claimSignature.insideValidity",
      "url": "self#jumbf=/c2pa/urn:c2pa:[REDACTED: id A]/c2pa.signature",
      "explanation": "claim signature valid"
      "code": "claimSignature.validated",
      "code": "assertion.dataHash.match",
      "explanation": "data hash valid"
    "informational": [],
    "failure": []
  "activeManifest": {
   "url": "self#jumbf=/c2pa/urn:c2pa:[REDACTED: id A]",
```

Selected lines (braces and three intermediate entries omitted). Manifest B records manifest A as its `parentOf` ingredient, together with the validation results the signing software computed for A at that time. This is a self-reported result inside the file; it is not an independent verification and says nothing about B's own signature.

Claims without a retained primary artifact

- The statement that "the signature validated" (S1, `openssl x509` "verified"): no verifier output, OCSP response or trust-list check is retained in the workspace; the only validation data is the signer-embedded record in Exhibit E14-6.
- The visual description of the poster (malformed title kerning, cast match) and the claim that the thumbnail shows "the same poster": only byte identity of the thumbnail and matching aspect ratio were re-derived here; no visual comparison was recorded as an artifact.

Exhibit inputs

Retained inputs identified by size and SHA-256. These include artifacts and analytical records; each exhibit identifies its evidence type and extraction limitations. The files themselves are not distributed.

ECHOES poster Post.png · 3,698,494 bytes

`SHA-256 d213b4184b93ccf59449237b071ac2e41ff71b83213b2e338917b26696a7b520`

Portrait Lyra.png · 862,883 bytes

`SHA-256 f7d17f0c0f035b67c96b8f37a4e55e27141df0c4d69b534a309191f398a66a47`

Portrait Solace.png · 605,016 bytes

`SHA-256 6e5e8ecf4b38580ed494556e0cd12150790ce0bf790b3da8588df91298aeadce`

Portrait Kael.png · 816,478 bytes

`SHA-256 8a0d38a6089e5c5d276820bb13089986798f5ca3e3136cd028a41e37da1892c0`

Previously extracted C2PA thumbnail (compared to embedded bidb payload) · 73,435 bytes

`SHA-256 fee5e557f82c8a500818b917f4861e781eee664315d27e0d99f1683957f13e57`

Source records and fingerprints

SHA-256 identifies the reviewed source bytes, not acquisition time or every artifact described. Restricted originals are not reproduced.

[S1] Recorded poster-provenance examination · 3,956 bytes

`SHA-256 77e532b9c3d412823450d4982530fbb51d609aa838d47fbe2aa8a934c5cc1478`