

Do AI provenance marks survive common image, video and audio pipelines?

EasyxLab · EasyByte Hub S. Coop. Mad. · Study S3

Working draft Published 2026-10-02 · Updated 2026-10-03

Abstract

Three legal regimes now require synthetic media to carry machine-readable marks: the EU AI Act (Art. 50(2), applicable from 2 August 2026), California's AI Transparency Act (SB 942 as amended by AB 853, operative 2 August 2026) and China's labelling rules (GB 45438-2025, in force since 1 September 2025). In practice those marks take three forms:

- a C2PA manifest whose actions declare an IPTC `digitalSourceType` of `trainedAlgorithmicMedia` ;
- the same IPTC term written into XMP;
- the Chinese AIGC XMP field.

A mark only does its job if it is still there when the file reaches a reader. Most websites and apps, however, re-encode what they publish. We built a reproducible local lab and ran signed test files through 41 transformations in the libraries most web stacks use: Pillow, sharp/libvips (including a replica of the sharp call chain used by the Next.js `next/image` optimizer), ImageMagick, ExifTool and ffmpeg. That gave 154 output files (including 8 unchanged identity controls) and 474 output × mark observations.

What we found:

- **Every pixel or container rewrite removed the embedded C2PA manifest (124/124).** That includes every "keep metadata" option we tested.
- **Editing a single XMP field after signing left the manifest present but invalid (6/6).** A check that only asks "is there a manifest?" gets this wrong.
- **The two XMP marks behaved identically in all 115 paired observations.** They survived only where the pipeline explicitly kept XMP, which is not the default in sharp, `next/image` , Pillow or ffmpeg.
- **The only tested route to a valid credential on a processed file was to re-sign it after processing,** with the original as an ingredient (4/4). But in all four the AI declaration survived only inside the ingredient: the new active manifest did not repeat it.

1. Why this matters now

The obligations are in force and are worded around detectability:

- **EU AI Act, Art. 50(2):** providers must ensure synthetic outputs "are marked in a machine-readable format and detectable as artificially generated or manipulated", with solutions that are "effective, interoperable, robust and reliable as far as this is technically feasible".

- **California SB 942:** a covered provider "shall include a latent disclosure in AI-generated image, video, or audio content" that conveys, among other things, "(A) The name of the covered provider" and "(B) The name and version number of the GenAI system that created or altered the content", and the disclosure must be "permanent or extraordinarily difficult to remove, to the extent it is technically feasible".
- **California AB 853** adds a duty for distributors, effective 1 January 2027 (§22757.3.1(b)): "A large online platform shall not, to the extent technically feasible, knowingly strip any system provenance data or digital signature that is compliant with widely adopted specifications adopted by an established standards-setting body from content uploaded or distributed on the large online platform."

The weak link sits between the generator, which adds the mark, and the reader or regulator, who looks for it. That link is the ordinary publishing pipeline: thumbnails, format conversion to WebP or AVIF, CMS uploads and video transcoding. Practitioners widely assume that "metadata is stripped". We wanted the precise, per-library answer, with each kind of mark separated out.

2. Marks under test

MARK	WHAT IT IS	WHERE IT LIVES
C2PA	signed manifest; its <code>c2pa.created</code> action declares <code>digitalSourceType = .../trainedAlgorithmicMedia</code>	JUMBF box (JPEG/PNG/WebP), <code>uuid</code> box (MP4/M4A), RIFF/ID3 (WAV/MP3)
XMP-DST	IPTC Photo Metadata <code>Iptc4xmpExt:DigitalSourceType</code> with the same IPTC term	XMP packet
AIGC	Chinese implicit-label field <code>TC260:AIGC</code> (JSON with Label, ContentProducer, ProduceID...)	XMP packet
Remote reference	<code>dcterms:provenance</code> URL that points to a manifest stored elsewhere	XMP packet

3. Method

The full method is in [`METHOD.md`](METHOD.md); this is a summary.

Corpus. One deterministic synthetic image (800×600, seed 42), produced as JPEG, PNG and WebP, each carrying all four marks. The XMP marks are written first and the C2PA manifest after them, so the claim hashes the XMP. We also made a 2-second MP4, a 3-second M4A carrying the XMP marks plus C2PA, and a WAV and an MP3 carrying C2PA only. Signing uses the ES256 test certificate shipped with `c2patool` 0.27.22, without a timestamp authority. As an external control we used one public C2PA sample, `CA.jpg`, from the `c2pa-rs` test fixtures (MIT OR Apache-2.0).

Pipelines. 41 transformations, each library at a single pinned version:

LIBRARY	VERSION	TRANSFORMATIONS
Pillow	12.3.0	re-save, thumbnail, WebP/AVIF conversion, each with and without the <code>exif=</code> / <code>xmp=</code> save parameters; JPEG conversion (no keep variant); PNG with an explicit <code>XML:com.adobe.xmp</code> iTXt chunk
sharp	0.35.5 (libvips 8.18.7)	resize and format conversion, with the default, <code>keepMetadata()</code> , <code>withMetadata()</code> , <code>keepXmp()</code> and <code>keepExif()</code>
<code>next/image</code> (replica)	next 16.3.8 call chain on sharp	<code>rotate().resize(640,</code> <code>{withoutEnlargement:true}).webp({quality:75})</code> and the AVIF equivalent: the sharp call chain of <code>optimizeImage()</code> , minus its timeout and input limits, which do not touch metadata
ImageMagick	7.1.2-32 (official WASM build)	re-save, resize, <code>strip()</code> , <code>strip</code> + resize, conversion to WebP
ExifTool	13.55	edit one XMP field after signing; <code>-all=</code> ; restore with <code>-</code> <code>tagsFromFile <original> -all:all</code>
ffmpeg	8.1.2	<code>remux(-c copy), +faststart, -map_metadata -1,</code> <code>use_metadata_tags, -export_xmp 1</code> , re-encode, scale, WebM, MP3, M4A
c2patool	0.27.22	remediation: re-sign the sharp output with the original as parent ingredient (<code>c2pa.resized</code>)

Classification. Every output is read with c2patool and ExifTool.

- A C2PA manifest is classified as one of:
 - **intact:** the only failures are the ones expected with a test certificate (`signingCredential.untrusted`, OCSP or TSA skipped);
 - **broken:** any other validation failure;
 - **removed:** no manifest;
 - **removed with residue:** no readable manifest, but signature or claim bytes are still present.
- An XMP mark is classified as **survived** if it parses, **residue** if its namespace or property string is in the bytes but does not parse, and **removed** otherwise.
- A mark is only evaluated on outputs whose input actually carries it, which we checked with an identity control for every input.

Nothing was uploaded to any platform or CDN. The only remote URL in the corpus is under `.invalid` (RFC 2606), so no request could leave the machine.

4. Results

4.1 The embedded C2PA manifest does not survive processing

C2PA	OUTCOME	OUTPUTS	WHERE
	removed	136	all 124 Pillow, sharp, <code>next/image</code> , ImageMagick and ffmpeg outputs, plus the 12 ExifTool strip/restore outputs
	broken (present but invalid)	6	ExifTool <code>-XMP-dc:Title=...</code> after signing: JPEG, PNG, WebP, MP4, M4A and the public sample
	intact	12	the 8 identity controls and the 4 re-signed outputs

No output fell into the "removed with residue" class (0/136): when the manifest went, it went entirely. No "keep metadata" option we tested preserved C2PA, and none can be expected to. The manifest's hard binding is a hash over the file's bytes, so re-encoding invalidates it even when a library copies the box, and in practice libraries do not copy it at all. ExifTool's `-tagsFromFile` restored both XMP AI marks, but neither the manifest nor the remote-reference pointer.

4.2 The silent failure: "present but invalid"

Editing a single XMP field after signing, which is what DAMs, CMS caption editors and SEO plugins do, left the manifest in the file with a failed data-hash assertion in all six cases. Any check that only detects the presence of a manifest, or that reads its assertions without validating them, reports these files as marked. **Validation must be part of the check.**

4.3 XMP marks: all-or-nothing, decided by the pipeline

XMP-DST and AIGC had identical outcomes in all 115 paired observations. The remote-reference URL followed them too, with one exception: `exiftool -tagsFromFile` restored XMP-DST and AIGC but not `dc:terms:provenance` (3/3). Their survival does not depend on the standard you choose. It depends on whether the pipeline keeps the XMP packet. Across the 110 non-identity outputs whose input carried XMP-DST, it survived in 48.

LIBRARY	DEFAULT BEHAVIOUR	TO KEEP THE XMP MARKS
sharp	removed (resize, WebP, AVIF)	<code>keepMetadata()</code> , <code>withMetadata()</code> or <code>keepXmp()</code> ; <code>keepExif()</code> is not enough
<code>next/image</code>	removed (WebP and AVIF)	none inside the optimizer, which calls sharp without any metadata option; <code>unoptimized</code> serves the original file (not tested)
Pillow	removed	pass <code>xmp=im.info["xmp"]</code> on save: kept the marks in JPEG, WebP and AVIF (the handbook documents <code>xmp</code> for WebP and AVIF; the JPEG writer also accepts it). In PNG <code>xmp=</code> is not a save option and is ignored; only an explicit iTXt <code>XML:com.adobe.xmp</code> chunk kept the marks
ImageMagick	kept on re-save, resize and WebP	removed by <code>strip()</code>
ExifTool	kept when editing; removed by <code>-all=</code>	restored by <code>-tagsFromFile ... -all:all</code>
ffmpeg	removed in every MP4/M4A configuration	none found; <code>-export_xmp 1</code> left an unreadable residue

4.4 Remote manifests are a pointer, not a credential

When provenance is stored externally, the in-file pointer (`dcterms:provenance` in XMP) survives where XMP survives, except after `-tagsFromFile` (§4.3). In a separate follow-up we signed files with a sidecar `.c2pa` manifest (nothing embedded, no XMP pointer), resized them with sharp `keepMetadata()`, and validated each derivative against the original's sidecar, offline. In all three cases (JPEG, PNG and WebP, `data/remote_followup.csv`) the original validated and the derivative failed with `assertion.dataHash.mismatch`. A surviving reference tells a reader where provenance can be found. It does not make the processed file carry a valid credential.

4.5 The only route that worked: re-sign after processing

Resizing with sharp `keepMetadata()` and then re-signing with `c2patool`, declaring the original as parent ingredient and the action `c2pa.resized`, produced a hash-valid manifest in all four cases (JPEG, PNG, WebP and the public sample), and kept both XMP marks in the three files that carried them.

There is a catch we did not anticipate. In all four files the new active manifest does **not** contain `trainedAlgorithmicMedia`: the AI declaration survives only inside the ingredient (the original's manifest), because our re-signing manifest declared only `c2pa.resized`. A reader that checks only the active

manifest will not see that the content is synthetic. The remediation pattern is therefore **process, then sign, and repeat the AI declaration in the new manifest**. We did not test that last variant.

4.6 Side observation: manifest size

With c2patool's default embedded thumbnail, a small WebP grew from about 18 KB to about 278 KB once signed. Size pressure may be one reason pipelines and CDNs discard the manifest.

5. What platforms and CDNs say (documented, not measured)

We did not upload anything anywhere. The following quotes are copied literally from official pages fetched on 2026-10-02 and are kept apart from the measurements above.

- **Cloudflare Images, "Preserve Content Credentials"**: "When Content Credentials are preserved in a transformation, Cloudflare will keep any existing Content Credentials embedded in the source image and automatically append and cryptographically sign additional actions. When this setting is disabled, any existing Content Credentials will always be discarded." The page states that the toggle is set per zone, that its behaviour "is determined by the metadata parameter for each transformation", and that it applies when optimizing images stored in remote sources. The `metadata` parameter documents `keep` ("Preserves most of EXIF metadata, including GPS location, if present") and `none` ("Discards all invisible EXIF metadata"). This is consistent with our remediation finding: the provider-side route is to re-sign.
- ****WordPress core (Imagick editor), `class-wp-image-editor-imagick.php` :** when stripping metadata it keeps a list of protected profiles, `$protected_profiles = array( 'icc', 'icm', 'iptc', 'exif', 'xmp', )`, so XMP is kept and other profiles are removed. We did not run WordPress, and the GD editor was not examined.
- **Meta (February 2024)**: "We're building industry-leading tools that can identify invisible markers at scale – specifically, the "AI generated" information in the C2PA and IPTC technical standards".
- **TikTok (May 2024)**: "we're expanding auto-labeling to AIGC created on some other platforms by launching the ability to read Content Credentials".

Platforms that read marks still depend on those marks arriving intact. Sections 4.1-4.3 show how easily the publisher's own stack removes them before upload.

6. Practical recommendations

1. **Check signatures, not presence.** Treat "manifest present but invalid" as a failure in its own right (§4.2).
2. **Decide what you need to keep, then configure for it explicitly.** XMP marks survive with sharp `keepMetadata()` / `withMetadata()` / `keepXmp()`, Pillow `xmp=` (except PNG), ImageMagick without `strip()`, and WordPress Imagick. Nothing in ffmpeg kept them in our tests.
3. **For C2PA, re-sign after processing**, with the original as ingredient, **and repeat the AI declaration in the new manifest**, or use a provider that re-signs (Cloudflare documents this behind a per-zone toggle). Check that the declaration is visible in the active manifest, not only in an ingredient.

4. **Test your own pipeline with before/after pairs.** Library defaults differ, and a one-line change (`keepExif()` in place of `keepMetadata()`) silently drops the XMP marks. The fixture pairs in `fixtures/derived/` exist for this purpose, and `ai-mark-lint --before/--after` (EasyByte Lab) automates the comparison.

7. Limitations

- **Synthetic corpus.** One image design, one short video and one short audio file. Results speak to container and metadata behaviour, not to image content.
 - **Pinned versions.** Library behaviour can change between releases; `data/versions.json` records the exact build.
 - **ImageMagick** was tested through its official WASM build, which uses the same 7.1.2-32 core but is not the CLI or PHP Imagick.
 - **Not tested:** PHP GD, WordPress Imagick (documented in §5, not run), a running WordPress or Next.js server, CDNs and social platforms (excluded by design), re-signing that repeats the AI declaration, HEIC, JPEG XL, GIF and TIFF, XMP in WAV/MP3, and non-XMP carriers of the AIGC label.
 - **Soft bindings** (invisible watermarks, fingerprints) are out of scope. The EU Code of Practice expects a multi-layer approach in which they matter.
 - ****The TC260:AIGC serialisation**** follows open-source implementations, not the official text of GB 45438-2025.
 - **"Intact" means hash-valid under a test certificate**, never "trusted".
-

8. Automation and review

This study was run by AI agents supervised by EasyByte. An LLM-based agent wrote the corpus generator, the pipelines and the classifier, and ran them. A second agent drafted this report from `data/matrix.csv` and the literal quotes. A third, independent agent reviewed every figure against the data, checked each quote against the fetched sources and re-ran `scripts/run.sh` (byte-identical matrix). All results are mechanical and reproducible: anyone can re-run the matrix and compare.

9. Data and reproducibility

- `scripts/run.sh` regenerates the corpus, all 41 transformations and the matrix in about 15 seconds. Two further runs from scratch, one by an independent reviewer, produced a byte-identical `matrix.csv`. Note that it also regenerates `fixtures/`: the signed files change bytes on every run (new signatures), while every classification stays the same.
- `data/matrix.csv` has one row per output × mark (474 rows); `data/summary.md` and `data/summary_compact.md` are the generated pivots.
- `fixtures/` is a small reusable corpus (under 3 MB) documented in `fixtures/README.md`.

Code: Apache-2.0. Data, fixtures and text: CC BY 4.0.

References

- Regulation (EU) 2024/1689 (AI Act), Art. 50, as reproduced at artificialintelligenceact.eu (EUR-Lex blocked automated retrieval), retrieved 2026-10-02.
- California SB 942 (2024), Business and Professions Code §22757 et seq., [leginfo.legislature.ca.gov](https://leginfo.ca.gov), retrieved 2026-10-02.
- California AB 853 (2025), §22757.3.1, [leginfo.legislature.ca.gov](https://leginfo.ca.gov), retrieved 2026-10-02.
- C2PA Technical Specification 2.2; IPTC Digital Source Type vocabulary.
- Cloudflare Images documentation: "Preserve Content Credentials" (last updated 2026-05-26) and "Transform via URL", retrieved 2026-10-02.
- WordPress/wordpress-develop, `src/wp-includes/class-wp-image-editor-imagick.php` (trunk), retrieved 2026-10-02.
- Meta Newsroom, "Labeling AI-Generated Images on Facebook, Instagram and Threads" (February 2024); TikTok Newsroom (May 2024).
- Pillow handbook, "Image file formats"; sharp API documentation; Next.js 16.3.8 `dist/server/image-optimizer.js`.

Cite this study

CITATION

EasyxLab (2026). Do AI provenance marks survive common image, video and audio pipelines? Study S3. EasyByte Hub S. Coop. Mad.
<https://github.com/easybytehub/easyxlab/tree/main/studies/s3-ai-marks-survival>

BIBTEX

@techreport{easyxlab_s3,
 title = {Do AI provenance marks survive common image, video and audio pipelines?},
 author = {{EasyxLab}},
 institution = {EasyByte Hub S. Coop. Mad.},
 number = {S3},
 year = {2026},
 url = {https://github.com/easybytehub/easyxlab/tree/main/studies/s3-ai-marks-survival}
}