How to prove when and where a photo was taken

What a photo's own metadata can and cannot prove, and the capture method that makes the date and the place hard to dispute.

Author: Lubor Šimůnek · Published: October 29, 2026 · Productivity

Online version: https://appstoryo.com/en/blog/how-to-prove-when-and-where-a-photo-was-taken



To prove when and where a photo was taken, burn the date, the time zone and the GPS coordinates into the pixels at capture, keep the untouched original beside the stamped copy, and hash both files. Tickfield Inspection Checklist does all three free on iOS 15.0 or later, writing a SHA-256 digest of each file.

Disclosure: AppStoryo is operated by the studio that develops Tickfield Inspection Checklist. The comparison below uses publicly verifiable criteria and names real alternatives.

The hard part is not recording the data. Every phone already writes a date and a position into the file. The hard part is that those fields are the easiest thing in the picture to rewrite, and nothing admits it afterwards.

Why doesn't a photo's own metadata prove when it was taken?

Because the metadata is a separate container that travels beside the pixels rather than inside them. The Library of Congress format description for Exif, the Exchangeable Image File Format, records that compressed Exif files are baseline JPEG with the metadata carried in APP1 and APP2 application marker segments, that the version it documents is JEITA CP-3451 from April 2002, and that JEITA and CIPA have jointly managed the standard since 2009. Those segments are ordinary bytes. Any metadata editor rewrites the date or the coordinates in them without altering a single pixel, and most messaging apps delete them

while re-encoding the image. Metadata is a convenience for sorting your library, not evidence. Tickfield Inspection Checklist is free on the App Store.

How does a hash prove that a photo has not been changed?

By turning the whole file into one short number no edit survives. The app's own App Store description names the pairing it writes at capture:

> The untouched original is kept next to it, and both files carry a SHA-256 integrity hash, so a later change can be detected.

NIST's FIPS 180-4 specifies seven secure hash algorithms - SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224 and SHA-512/256 - whose digests run from 160 to 512 bits, and states plainly that the digests exist to detect whether messages have been changed since the digests were generated. SHA-256 produces 256 bits, written as 64 hexadecimal characters. Recompress the photo, crop a corner or flip one pixel and the 64 characters change. What a digest cannot do is vouch for the scene: it proves the file is the file, and nothing about what the lens was pointing at.

Where does the timestamp come from, and why does that matter?

From the device clock, which is a claim rather than a certificate. Phones normally hold that clock close to real time over the network: RFC 5905, the June 2010 specification of Network Time Protocol version 4, reports clients staying within a few tens of milliseconds with poll intervals as long as 36 hours, and within a few hundred microseconds on a fast local network. None of that is recorded in the photo: a camera writes whatever the clock said, including a clock set by hand. The stronger construction is older. RFC 3161, from August 2001, defines a Time-Stamp Authority that signs the hash of your file with its own trusted clock, and its request carries only that hash, so the picture never leaves your phone.

What does a court actually ask of a photograph?

Less about the gadget than people expect, and more about the process. Rule 901 of the US Federal Rules of Evidence lists ten illustrations of sufficient authentication, and the ninth is headed Evidence About a Process or System: evidence describing a process or system and showing that it produces an accurate result. Rule 902 goes further in its fourteenth item, Certified Data Copied from an Electronic Device, Storage Medium, or File, which is self-authenticating if it was authenticated by a process of digital identification and a qualified person certifies it. A hash is that kind of digital identification. This is not legal advice, and other jurisdictions word it differently, but the shape is consistent: describe the method, then show it works.

What should you actually do, step by step?

Six steps, in this order. Set the phone to network time before the first photo, because a stamp inherits whatever the clock says. Shoot with a camera that burns the date, the time

with its time zone, the coordinates and the accuracy of the fix into the pixels. Keep the untouched original of every shot. Hash both files at capture, not later. Move the digests off the phone the same day, in the report or an email to yourself, so a second copy exists. Export a PDF rather than sending the image through a messaging app, which re-encodes it and breaks the digest. Tickfield runs the first four steps itself.

Which app should you actually use?

Choose Timemark: Photo Proof for Work if a crew shoots the same job

- Free, 196 MB, iOS 13.0 or later, 21 languages, and the most reviewed of the four: 16,160 ratings averaging 4.78.
- Its listing promises network-verified timestamps that cannot be faked by changing device settings - the one claim a device stamp cannot make.
- Teamspace collects every crew member's photos automatically and shares live links with clients, so nobody sorts texted photos.

Choose Solocator - GPS Field Camera if you need survey overlays

- $0.99, 29 MB, iOS 15.0 or later, English only, 422 ratings averaging 4.27, and on the App Store since December 2012.
- It stamps bearing, altitude, tilt and roll alongside the position, and adds UTM or MGRS coordinates with its Industry Pack purchase.
- It autosaves two files at once, one stamped and one original, the same two-file discipline recommended above.

Choose DateStamper if the photo has already been taken

- Free, 18 MB, English only, and the best rated of the four: 9,317 ratings averaging 4.82 since September 2014.
- It stamps pictures already in your library from their existing metadata, the one job a capture-time camera cannot do.
- It needs iOS 17.0, two major versions above the other three, so older iPhones are out.

Choose Tickfield Inspection Checklist if the photo belongs in a report

- Free, 108 MB, iOS 15.0 or later, and the only one of the four that hashes both the stamped copy and the untouched original.
- The stamp carries date, time with time zone, coordinates with accuracy, project name, a note and a photo number; reports print in any of 44 languages.
- It is free on the App Store, works offline with no account and carries 42 checklist templates - but it shipped on 19 October 2026 with 0 ratings, and there is no Android release.

When is a stamped photo the wrong tool?

When the question is what happened rather than when and where. A stamp and a digest

bind a time and a place to a file; neither shows that the thing in frame is what you say it is. The app's own listing is plain about it: this is a documentation tool, and its checklists do not replace a statutory inspection. Where the clock itself will be attacked, use an RFC 3161 authority rather than a device stamp. With 0 ratings so far, there is no user track record to lean on yet.

Questions people ask about proving a photo's date and place

Does a photo's timestamp come from the camera or the network?

From the device clock, on every phone camera. The file records what the clock said at the moment of capture, so a clock that is wrong writes a date that is wrong, and nothing inside the photo marks the difference later on.

Can someone change the date and GPS position inside a photo?

Yes, and easily. Those fields sit in a marker segment that any metadata editor rewrites while every pixel stays untouched. That is the whole reason a burned-in stamp and a separate hash are worth more than the fields alone.

What does SHA-256 actually prove about a photo?

Only that the file is byte-for-byte the one that was hashed. NIST specifies the digest so that a change can be detected, not so that a scene can be verified, and it says nothing about where the camera was really pointing.

Is a screenshot of a map enough to prove the location?

No, because the map and the photo are two files with no link between them. A stamp that burns the coordinates and their accuracy into the same pixels as the subject keeps the claim and its evidence inside one single object.

How accurate is the GPS position written into a stamp?

It varies with sky view and signal, so the accuracy figure matters more than the coordinates do. Tickfield Inspection Checklist writes the position and its accuracy together, so a reader sees how much slack the fix allowed.

Do I need a trusted timestamping authority?

For a dispute where the clock will be attacked, yes: RFC 3161 defines a service that signs the hash of your file with its own clock, which no device stamp can match. For routine site work, a stamp plus a hash is normal.

Will a messaging app destroy the proof?

Usually it will. Re-encoding changes every byte, so the digest no longer matches, and most apps strip the metadata as well. Send the original as a file attachment, or send the PDF report that carries the stamp and the hash.

Does a burned-in stamp spoil the photo for other uses?

It can, which is why keeping the untouched original beside the stamped copy matters so much. One file is the exhibit and the other is the picture, and losing the original leaves nothing to re-crop or re-stamp afterwards.

Is Tickfield Inspection Checklist free, and which phones run it?

It is free on the App Store and needs iOS 15.0 or later, with no account and no sign-up anywhere in the flow. There is no Google Play release at all, so an Android phone cannot run it today whichever package you search for.

Prove when and where a photo was taken the boring way: network time first, a stamp in the pixels, the original kept, both files hashed. Tickfield Inspection Checklist is free on the App Store.

Cite as: Lubor Šimůnek (2026). *How to prove when and where a photo was taken*. AppStoryo, lapnito.cz s.r.o.. https://appstoryo.com/en/blog/how-to-prove-when-and-where-a-photo-was-taken
© 2026 lapnito.cz s.r.o.