COMPUTER FORENSICS LAB

London digital forensics & e-discovery · Established 2007

Can File Timestamps Be Trusted?

What Each Timestamp Measures, When It Lies, and How to Tell

A file's dates look like facts: created on this day, modified on that. Lawyers plead them, witnesses swear by them, and cross-examiners dismantle them, because a timestamp is not a fact but a record: written by a particular system, from a particular clock, under rules that differ by file system, platform and operation, and changeable by anyone who knows how. Some timestamps are excellent evidence; some are noise; the difference lies in knowing which system wrote them, what event they record, whether the clock was right, and whether the value could have been set by hand. This guide gives UK lawyers the working knowledge: the timestamp families and their semantics across Windows, macOS, cloud platforms and documents; the innocent processes that rewrite dates wholesale; the corroboration that makes a timestamp reliable; the detection of manipulated stamps from the traces manipulation leaves; and the honest statement of what a date can and cannot prove.

⏱ Estimated reading time: 24 min read

§ ABOUT THE AUTHOR

PREPARED BY COMPUTER FORENSICS LAB E-DISCOVERY TEAM

Computer Forensics Lab is a London-based digital forensics and electronic disclosure practice established in 2007. Timestamp questions underlie a large share of its authenticity, chronology and attribution work: examiners read timestamps by system and semantics, calibrate clocks, corroborate across independent records, and detect manipulation from the traces it leaves: and report under CPR Part 35 where a date is the issue.

ESTABLISHED 2007 · LONDON | ISO 17025-ALIGNED PROCEDURES | TIMESTAMP & CLOCK ANALYSIS | MANIPULATION DETECTION | CPR PART 35 EXPERT REPORTS | FULL CHAIN-OF-CUSTODY DOCUMENTATION

§ CONTENTS

In this guide

§ 0 1 · ORIENTATION

Executive summary

THE HEADLINE POINT: A TIMESTAMP IS TRUSTWORTHY WHEN YOU KNOW WHICH SYSTEM WROTE IT, WHAT EVENT IT RECORDS, THAT THE CLOCK WAS RIGHT, AND THAT INDEPENDENT RECORDS AGREE: AND UNTRUSTWORTHY, OR AT LEAST UNPROVEN, WHENEVER ANY OF THOSE IS UNKNOWN

The families (§3): file-system timestamps (created, modified, accessed, and on NTFS the metadata-changed entry) are written by the operating system under rules that differ by file system and operation; document-internal timestamps (Office creation and last-saved dates, PDF creation and modification, image EXIF capture time) are written by applications and travel inside the file; platform timestamps (cloud version rungs, audit-log events) are written server-side by synchronised clocks; transmission timestamps (email headers per hop, message-server times) are written by the systems the item passed through. Each family measures a different event with a different clock and a different vulnerability. **Innocent rewrites (§4):** copying a file creates a new "created" date; moving across volumes may too; migrations, restores and sync clients rewrite dates wholesale; time-zone presentation shifts values by hours: the created-date cliff of guide 87 and the migration day of guide 96 Example 3 are routine, not sinister. **Clocks (§5):** a timestamp is only as good as the clock that wrote it: unsynchronised devices drift, misconfigured zones offset, and calibration against reference events (guide 96 §4) turns approximate stamps into bounded ones. **Manipulation (§6):** timestamps can be set by hand with freely available tools, by adjusting the system clock before saving, or by editing document metadata: and each method leaves traces: NTFS's second, hidden set of timestamps; the change journal; the mismatch between file-system and document-internal dates; the application-version anachronism; the platform record that disagrees. **Trustworthiness (§7):** a timestamp is pleaded at the grade its corroboration supports: server-side, synchronised, cross-checked stamps as reliable; single local stamps as indicative; stamps contradicted by independent records as suspect: with guide 96's timeline as the frame in which the grading is done.

- **Know the family:** file-system, document, platform, transmission: different events, different clocks, different vulnerabilities.
- **Copies reset dates innocently:** a created date after the modified date is usually a copy, not a lie.
- **Server-side beats local:** synchronised platform clocks and audit events are the reliable stratum; local file dates are indicative.
- **Manipulation leaves traces:** hidden timestamp sets, journals, anachronisms and platform disagreement expose the hand-set date.
- **Plead at the grade earned:** corroborated stamps as facts; single stamps as indications; contradicted stamps as questions.

§ 0 2 · FIRST PRINCIPLES

The problem in plain English: a timestamp is a record, not a fact

When a file's properties say "Created: 14 March 2023, 10:22", the user reads a fact about the document's origin. The system that wrote it recorded something narrower: at the moment this particular copy of the file appeared on this particular volume, this machine's clock read 10:22 on 14 March 2023 in its configured time zone. Whether the document was authored then, whether the clock was right, whether the value has since been edited, and whether the "copy" was the first or the fifth: the timestamp says nothing about any of it.

This does not make timestamps worthless; it makes them evidence to be interpreted, like a witness's recollection. Server-side stamps from synchronised clocks, cross-checked against independent systems, are among the most reliable evidence in digital forensics. A lone "modified" date on a file that has been copied, migrated and emailed is among the least. The lawyer's task is to know which kind is in front of them, and this guide's structure follows that task: what each stamp measures, what innocently rewrites it, what clock wrote it, how it can be faked, and how it is graded before it is pleaded.

§ 0 3 · THE FAMILIES

The timestamp families: file system, document, platform and transmission

- **File-system timestamps:** the operating system's record of a file's life on a volume: created (when this copy appeared here), modified (last content write), accessed (last read, often disabled or coarsened for performance), and, on NTFS, the entry-modified stamp recording changes to the file's metadata itself: with NTFS holding these in two places (the standard information attribute the user sees and the file-name attribute the user does not), a duplication that matters for §6.
- **Document-internal timestamps:** Office creation and last-saved fields, revision and editing-time counters, PDF creation and modification dates, EXIF capture time in images: written by the application from the machine's clock, carried inside the file through every copy, and editable with ordinary tools: guide 87's families, read for what the application recorded rather than what the file system did.
- **Platform timestamps:** cloud version-ladder rungs (guide 102), audit-log events (guide 105), DMS check-ins: written by the server's synchronised clock at the moment of the operation: the stratum least affected by client-side manipulation, and the one guide 96 builds its critical path on.
- **Transmission timestamps:** email header received-lines per hop, messaging-server delivery times, sending-client times: several clocks recording one message's journey, with the server hops as independent witnesses to the client's claimed time.
- **Semantics per family:** what each stamp means is established before any two are compared: guide 96 §4's semantic table, which this guide's §3 populates.

§ 0 4 · INNOCENT REWRITES

Innocent rewrites: copies, moves, migrations, sync and time zones

- **Copy resets created:** copying a file creates a new file-system entry with a fresh created date and the original's modified date: producing the "created after modified" pattern that alarms the untrained and means "this is a copy" to the examiner: guide 87's created-date cliff.
- **Moves, extracts and downloads:** moves within a volume preserve dates; moves across volumes behave like copies; extracting from an archive, downloading from the web or email, and restoring from backup each apply their own rules (sometimes preserving original dates, sometimes stamping the extraction time): the operation identified before the date is read.
- **Migrations and bulk operations:** server migrations, tenant moves, DMS upgrades and storage consolidations rewrite dates across whole libraries on a single day (guide 96 Example 3, guide 102 Example 2): the migration record is the explanation, and the pre-migration backup the original.
- **Sync clients:** cloud sync tools write local copies with dates that may reflect sync time, server time or original time depending on the client and version: the local stamp on a synced file is the client's decision, not the author's.
- **Time-zone presentation:** stored values are typically UTC; displayed values are local: a file "created at 01:15" in a British summer may have been created at 00:15 UTC and displayed to a New York reader at 20:15 the previous day: guide 96 §4's normalisation rule, applied before any comparison.

§ 0 5 · CLOCKS

Clocks: drift, misconfiguration, time zones and calibration

- **Synchronised versus free-running:** domain-joined and cloud-connected devices synchronise to time servers and keep good time; standalone machines, embedded devices, cameras and DVRs drift by minutes to hours per year: the clock's status is the first question about any local timestamp.
- **Misconfiguration:** wrong time zone, wrong date after a battery failure, daylight-saving settings disabled: systematic offsets that shift every stamp the device writes, detectable by comparing the device's stamps with external references.
- **Calibration:** reference events visible on both the questioned clock and a trusted one (an email received and saved, a sync event logged server-side, a photograph of a known event) establish the offset at a point in time: guide 96 §4's method, turning an untrusted clock into a bounded one.
- **Clock changes as evidence:** system event logs record time changes (Windows logs clock adjustments as events with the old and new values): a clock set back before a save, then forward again, is logged: §6's manipulation traced from the clock's own diary.
- **Precision and resolution:** stamps resolve to different precisions (NTFS to 100-nanosecond intervals, FAT to two seconds, some applications to the minute or day): comparisons claimed only at the coarser resolution, per guide 96 §10.

§ 0 6 · MANIPULATION

Manipulation: how timestamps are changed, and the traces it leaves

- **Direct timestamp editing:** freely available utilities set file-system dates to chosen values: but on NTFS they typically change the standard-information stamps and not the file-name attribute's copy, producing a mismatch between the two sets that the examiner reads directly; and the entry-modified stamp records the change itself.
- **Clock rollback:** setting the system clock back, saving or creating the file, and restoring the clock: producing internally consistent stamps: but leaving the clock-change events in the system log, the change journal's sequence out of order with its timestamps, and neighbouring files whose stamps expose the anomaly.
- **Document-metadata editing:** Office and PDF fields changed with property editors: leaving inconsistencies between the edited fields and the unedited ones (revision counts, editing time, application version), between the document and the file system, and between the document and any platform or transmission record: guide 87's manipulation signatures and guide 109's backdating tests.
- **The sequence tells:** file-system entries and journal records are ordered by the sequence in which they were written, independent of the timestamps they carry: a file whose stamps say 2021 but whose MFT entry and journal position sit among 2024 files is exposed by its neighbours.
- **Anachronisms:** the application build, template, font or feature that did not exist at the claimed date: guide 87 and 109's territory: the manipulator who set the date but could not unmake the tool.

§ 0 7 · TRUSTWORTHINESS IN PRACTICE

Trustworthiness in practice: corroboration, grading and pleading dates

- **Corroboration first:** a timestamp is checked against independent records before it is relied on: the platform ladder against the local file, the email header hops against the sending client, the audit log against the document's internal date: agreement across systems that could not have been jointly manipulated is what makes a stamp reliable.
- **The grading:** corroborated server-side stamps as reliable; single local stamps with a known-good clock as indicative; stamps from unknown or drifting clocks as approximate, bounded by calibration; stamps contradicted by independent records as suspect and investigated: guide 96 §5's confidence grades, applied per stamp.
- **Pleading dates:** "the document was created on 14 March" is pleaded only where the grade supports it; otherwise "the file's metadata records a creation date of 14 March, corroborated by [X] and consistent with [Y]" is the honest form: the difference between a fact and a defended assertion.
- **Challenging the opponent's dates:** the same checklist in reverse: which family, which clock, what corroboration, what innocent rewrite might explain the anomaly, and whether §6's traces exist: with guide 109 taking the backdating allegation forward where the traces are found.
- **Chain-of-custody protection:** the examiner's own handling must not rewrite what it examines: write-blocking, forensic imaging and hash verification (guide 1) preserve the stamps as found, and the acquisition record proves they were not altered in the laboratory: the timestamp's integrity is the examiner's first duty.

§ 0 8 · THE WIDER MAP

Source architecture: where else the evidence lives

Per this series' source-architecture discipline: no single timestamp should be trusted alone because no single timestamp needs to be. A document's date is recorded by its file system, inside itself, by the platform that stored it, by the systems that transmitted it, by the backups that captured it, and by the counterparties who received it: each with its own clock and its own exposure to manipulation. The discipline is to hold at least two independent records for every date the case depends on, chosen so that they could not have been jointly falsified: the local file's stamp against the server's audit event; the document's internal date against the email header that carried it; the claimed creation date against the earliest backup containing the file. When one record is missing, drifting or suspect, the table names the others. The timestamp that cannot be corroborated is not thereby false; it is thereby unproven, and pleaded as such.

QUESTION	FILE-SYSTEM STAMPS	DOCUMENT-INTERNAL STAMPS	PLATFORM RECORDS	TRANSMISSION RECORDS	BACKUPS + COUNTERPART COPIES	SYSTEM + CLOCK LOGS
When was it created	Indicative: this copy, this volume	Application's record; editable	Y: first rung, upload event	Earliest send brackets it	Y: earliest capture	n/a
When was it last changed	Modified stamp; copy-sensitive	Last-saved; editable	Y: last rung, edit event	Latest sent state	Latest capture	n/a
Was the clock right	n/a	n/a	Y: synchronised reference	Y: server hops	Backup-server time	Y: sync status, clock changes
Was the stamp altered	Y: dual-attribute mismatch, sequence	Field inconsistencies	Disagreement exposes it	Header disagreement exposes it	Prior state disagrees	Y: clock-change events, journal
Is the anomaly innocent	Copy / move patterns	Save-as resets	Migration events	n/a	Pre-migration originals	Sync and restore logs

Fallback strategy in one line: never plead a date on one clock: hold the server's record against the file's, the header against the document, the backup against the claim: and where only one survives, plead the indication, not the fact.

§ 0 9 · IN THE WILD

Worked examples

EXAMPLE 1 · THE CREATED DATE THAT WAS A COPY

A claimant alleged that a defendant's key policy document had been fabricated after the dispute began, pointing to a file-system created date three weeks after the letter of claim, later than its own modified date. The §4 reading was immediate: created-after-modified is the copy signature. The examiner traced the copy: the document had been moved from a departing employee's OneDrive to a shared library by IT during a leaver process (guide 103), the move across storage behaving as a copy; the original's platform ladder (guide 102) ran from two years earlier with forty rungs; the earliest backup containing the file predated the dispute by nineteen months; and the document's internal creation date matched the first rung. The §8 corroboration was unanimous; the anomaly was a leaver process. The allegation was withdrawn. A copy had been pleaded as a fabrication because nobody asked which event the stamp recorded.

EXAMPLE 2 · THE CLOCK THAT WAS SET BACK, AND THE LOG THAT NOTICED

A director produced a board resolution with file-system and internal dates in June 2022, supporting his account of a decision taken then. The §6 examination found the stamps internally consistent and the corroboration absent: no platform record (the file had lived only on his laptop), no transmission before 2024, no backup capture before 2024. The laptop's system event log then supplied the mechanism: a clock-change event on a date in 2024 setting the system time back to June 2022, a second event restoring it forty minutes later, and the change journal recording the resolution's creation between the two, with its journal sequence number among 2024 entries. The application build recorded in the document's internal metadata had been released in 2023. The date was consistent because it had been manufactured consistently; the clock's own diary and the journal's ordering exposed it. Guide 109 took the finding forward.

EXAMPLE 3 · THE TIMESTAMPS THAT DISAGREED AND THE ONE THAT WAS RIGHT

An insurance claim turned on when a photograph of storm damage had been taken. The image's EXIF capture time said the morning after the storm; the phone's file-system stamp said three days later; the cloud photo library's upload event said the same afternoon as the EXIF time; and the insurer's suspicion had been raised by the file-system date. The §5 analysis resolved it: the phone's file-system stamp recorded the date the image was exported from the photo library to the file system for emailing (an operation that writes a new file), not the capture; the EXIF time was written by the camera's clock, which the phone synchronised from the network; and the upload event, server-side and synchronised, agreed with EXIF to the minute. Three stamps, three events, one clock in doubt: the corroborated pair was right, the outlier was a different event, and the claim was paid.

§ 10 · WHERE IT GOES WRONG

Common mistakes and technical limitations

Common mistakes

- **Copies pleaded as fabrications:** Example 1's created-after-modified: the copy signature read as a lie.
- **Families mixed:** a file-system stamp compared with a document field as if they measured the same event: §3's semantics ignored.
- **Local stamps trusted alone:** Example 2's consistent-but-manufactured dates, accepted because nothing was checked against them.
- **Time zones unnormalised:** the hour that moves an event across a deadline: guide 96's first error, again.
- **Clock status unasked:** a free-running device's stamps pleaded to the minute.
- **The examiner's own handling:** files opened, copied or previewed before imaging, rewriting the accessed and sometimes modified stamps the case needed.
- **Manipulation alleged without traces:** "the dates must be false" without the dual-attribute, journal or anachronism evidence of §6.
- **Migration history unasked:** a whole library's rewritten dates read as suspicious rather than as a migration day.

Technical limitations

- **Some stamps are simply unreliable:** access times disabled or coarsened by policy; FAT volumes with two-second resolution and no zone; application fields on machines with generic configuration: stated as such.
- **Sophisticated manipulation can align the dual attributes:** tools exist that set both NTFS timestamp sets: the sequence, journal and external-record checks remain, and the report says which tests were run.
- **Platform records have retention:** the server-side corroboration expires with audit logs and version limits (guides 102, 105): harvested early or unavailable.
- **Calibration needs reference events:** a drifting clock with no cross-referenced events yields bounded uncertainty, not a corrected value.
- **Absence of corroboration is not proof of falsity:** an uncorroborated stamp is unproven; the report distinguishes "cannot be verified" from "shown to be false".

§ 11 · INTERROGATORIES & DRAFTING AIDS

Questions to ask · Suggested wording

Ask your client

- Which dates does the case depend on, and from which family and system does each come?
- Has the file been copied, moved, migrated, restored or synced since the claimed date: and are the platform original and earliest backup available?
- Was the device's clock synchronised, and has anyone opened the files since the dispute arose?

Ask your opponent

- Please identify, for each date relied on in respect of [document], the system that recorded it, the event it records, and the clock synchronisation status of that system, and disclose the file-system metadata (including all timestamp attributes), document-internal metadata, platform version and audit records, and transmission records for the document.
- Please disclose the system event logs of [device] for [period], including any clock-change events, and the file-system change journal so far as it survives.
- Please confirm whether [document] has been copied, moved, migrated, restored or synchronised since [date], with the records of any such operation.

Ask your eDiscovery / forensic provider

- Which family is each disputed stamp from, and what is its corroboration?
- Do the NTFS dual attributes, the journal sequence and the application build agree with the claimed date?
- Was the acquisition write-blocked and hashed so the stamps we are reading are the stamps as found?

SUGGESTED WORDING · INSTRUCTION FOR TIMESTAMP RELIABILITY EXAMINATION

> "You are instructed to examine the timestamps associated with [documents] and to report on their reliability. Please identify for each timestamp the recording system, the event recorded and the clock concerned; assess clock synchronisation and any offset by calibration against reference events; identify any copy, move, migration, restoration or synchronisation operations that may have rewritten timestamps; corroborate each material timestamp against independent records (platform version and audit records, transmission records, backups and counterpart copies); and examine for indications of manipulation including file-system dual-attribute inconsistencies, journal sequence anomalies, clock-change events, document-metadata inconsistencies and application or feature anachronisms. Please grade each timestamp's reliability, distinguish unverifiable from demonstrably false, and preserve the original evidence by write-blocked, hash-verified acquisition, reporting in a form suitable for CPR Part 35 [/ CrimPR Part 19]."

§ 12 · QUICK CONTROL

Checklist and red flags · When to involve a digital forensic expert

The timestamp checklist

- ☐ Evidence acquired write-blocked and hashed before any reading
- ☐ Each stamp's family, system and event identified
- ☐ Time zones normalised per guide 96 before comparison
- ☐ Clock synchronisation status established; drift calibrated
- ☐ Copy, move, migration, restore and sync history checked
- ☐ Platform, transmission and backup records collected for corroboration
- ☐ NTFS dual attributes and journal sequence examined
- ☐ System logs checked for clock-change events
- ☐ Document-internal fields checked for inconsistency and anachronism
- ☐ Each stamp graded; dates pleaded at the grade earned

Red flags

- Created after modified: a copy until proven otherwise: Example 1.
- Dates consistent everywhere but corroborated nowhere: Example 2's tell.
- Clock-change events near a disputed save.
- Journal or MFT sequence positions inconsistent with carried dates.
- Application builds newer than the claimed date.
- Whole libraries sharing one date: a migration, not a conspiracy.
- Files opened or copied by anyone after the dispute arose, before imaging.

When to involve a digital forensic expert

Before anyone touches the evidence, so the stamps are preserved as found; when a date matters, to identify its family, clock and corroboration before it is pleaded (Example 3's three stamps); when a date is challenged, to run the §4 innocent-rewrite checks before the §6 manipulation tests (Example 1's copy); and when manipulation is suspected, to find or exclude the traces and report under CPR Part 35 or CrimPR Part 19 (Example 2's clock diary), with guide 109 continuing where backdating is found. Through **cflab.uk** and **e-discovery.uk**, timestamp work runs from write-blocked acquisition to graded finding: because a date is only as good as the question "which clock, which event, what agrees?".

§ 13 · COMMON QUESTIONS

Frequently asked questions

The file says it was created after it was last modified. Is that proof of tampering?

Almost never: it is the signature of a copy, a cross-volume move, an extraction or a restore, which give the new file-system entry a fresh created date while preserving the content's modified date (§4): Example 1's leaver-process move. It becomes a question only when the copy cannot be traced and the other families disagree with the claimed history.

Which timestamps are most reliable?

Server-side stamps from synchronised platforms (version rungs, audit events, email server hops), corroborated by each other (§3, §7): they are written by clocks the user does not control and stored where the user cannot edit. Local file-system and document-internal stamps are indicative: useful, editable, and graded accordingly.

Can timestamps really be faked?

Yes, trivially, with free tools or a clock rollback: and the faking leaves traces: NTFS's second timestamp set, the journal's ordering, the logged clock change, the application build that postdates the claimed date, and the platform or transmission record that disagrees (§6): Example 2's resolution was exposed by all four. The forger controls the value, not the context.

Our device's clock was wrong. Are its timestamps useless?

No: a systematically wrong clock is calibrated against reference events (a received email, a server-logged sync, a dated photograph) to establish the offset, and the stamps corrected within stated bounds (§5, guide 96 §4). What is useless is a wrong clock nobody knew about; a known offset is just arithmetic.

The other side's chronology rests on file dates. How do we test it?

With §7's checklist in reverse: which family is each date from, which clock wrote it, what corroborates it, what innocent operation might explain any anomaly, and do §6's traces exist: then §11's requests for the platform, journal and clock records that answer. Chronologies built on single local stamps rarely survive the questions (guide 96 Example 3).

Does opening a file change its timestamps?

It can change the accessed stamp (where access-time updating is enabled), and some applications rewrite modified stamps or internal fields on open and auto-save: which is why evidence is imaged write-blocked before anyone reads it (§7), and why the "quick look" before instruction is the error guide 97 §3 warns against. The stamps as found are the evidence; handling rewrites them.

§ 14 · REFERENCE

Glossary

Accessed stamp
The file-system record of last read; often disabled or coarsened.

Calibration
Establishing a clock's offset from reference events.

Clock-change event
The system log's record of a manual time adjustment.

Created stamp
When this copy appeared on this volume; reset by copies.

Dual attributes (NTFS)
The two timestamp sets whose mismatch exposes editing.

Entry-modified stamp
NTFS's record of metadata changes, including timestamp edits.

Journal sequence
The order of file-system transactions, independent of carried dates.

Modified stamp
The file-system record of the last content write.

Reliability grade
Reliable, indicative, approximate or suspect, per corroboration.

Timestamp family
File-system, document-internal, platform or transmission origin.

Sources and authoritative references

The timestamp disciplines in this guide draw on materials published by our laboratory and e-discovery practice, referenced here as sources:

- cflab.uk, digital forensics services (timestamp and clock analysis, manipulation detection, CPR Part 35 reporting): cflab.uk/digital-forensics-services · guides library: cflab.uk/guides
- e-discovery.uk, EDRM Phase 04 · Collection and Phase 07 · Analysis (metadata preservation and chronology analysis as practised): e-discovery.uk/edrm/collection · e-discovery.uk/edrm/analysis
- e-discovery.uk, practice method and Knowledge Centre: e-discovery.uk · e-discovery.uk/knowledge
- Microsoft Learn, file times: learn.microsoft.com, file times · NIST SP 800-86 (file-system and timeline artefacts): csrc.nist.gov, SP 800-86
- CPR Part 35 and PD 35: justice.gov.uk, Part 35 · CrimPR Part 19: justice.gov.uk, CrimPR 19 · PD 57AD: justice.gov.uk, PD 57AD
- ISO/IEC 27037 (acquisition and preservation): iso.org, 27037 · ICO, UK GDPR guidance: ico.org.uk

§ HOW A SPECIALIST LABORATORY CAN ASSIST

Working with Computer Forensics Lab

Timestamp work at the laboratory starts with preservation, because the evidence is rewritten by handling: write-blocked, hash-verified acquisition before any stamp is read. Examination then follows this guide's order: each stamp's family, system and event identified; zones normalised and clocks calibrated per guide 96; the innocent rewrites of copy, move, migration and sync traced from their records (Example 1's leaver process); corroboration assembled from platform, transmission and backup layers (Example 3's agreeing pair); and manipulation tested from the dual attributes, the journal sequence, the clock's own log and the anachronisms (Example 2's forty minutes): with each date graded and the report distinguishing the unverifiable from the false. Findings run under CPR Part 35 and CrimPR Part 19, and opposing chronologies are audited by the same questions. Conflicts checks, confidential scoping discussions and fixed-fee estimates are routine; and if a date is in dispute now, the first instruction is the simplest: stop opening the files.

INSTRUCT THE LAB

Speak to a forensic examiner, not a salesperson.

Describe your matter in confidence and an examiner will respond the same working day. Conflicts checks and scoping calls with US counsel are routine.

NEW ENQUIRIES
+44 (0)20 7164 6971

EMAIL
info@cflab.uk

E-DISCOVERY
e-discovery.uk

Computer Forensics Lab Ltd · Euro House, 133 Ballards Lane, Finchley, London N3 1LJ, United Kingdom
cflab.uk · Customer service +44 (0)20 7164 6915 · ICO ZB395023 · NPCC · ISO 17025 aligned practice · UK Gov Digital Marketplace