COMPUTER FORENSICS LAB

London digital forensics & e-discovery · Established 2007

# Digital Redaction That Cannot Be Reversed

## Why Black Rectangles Fail, How Real Redaction Works, and How to Verify It

The history of failed redaction is a history of drawing over text instead of removing it: black rectangles lifted off with a cursor, highlighted-in-black passages that copy-paste perfectly, "redacted" PDFs whose text layer still carries every word, metadata fields still naming the protected client, and images whose thumbnails preserve what the full picture hid. Each failure is the same category error: treating redaction as a visual act when it must be a data act. Courts, regulators and opponents all read the failures fluently, and the consequences run from guide 92's inadvertent-disclosure machinery to ICO enforcement. This guide explains irreversible redaction for UK lawyers: where content actually hides in files, the tooling and workflow that genuinely removes it, verification before anything leaves, spreadsheet and native-format problems, and the protocol terms that keep both sides' redactions honest.

⏱ 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. Its eDiscovery arm, **e-discovery.uk**, engineers redaction workflows on production matters: tooling selection, burn-in pipelines, verification testing and the audit records that prove removal: while the laboratory side examines failed redactions, recovers what they left behind, and reports on both under CPR Part 35.

ESTABLISHED 2007 · LONDON | ISO 17025-ALIGNED PROCEDURES | REDACTION ENGINEERING & VERIFICATION | FAILED-REDACTION EXAMINATION | CPR PART 35 EXPERT REPORTS | FULL CHAIN-OF-CUSTODY DOCUMENTATION

§ CONTENTS

# In this guide

§ 0 1 · ORIENTATION

# Executive summary

**THE HEADLINE POINT: REDACTION IS REMOVAL OF DATA, NOT CONCEALMENT OF PIXELS: BURN IN, SANITISE, VERIFY, AND KEEP THE AUDIT RECORD THAT PROVES ALL THREE**

A document is layered: visible rendering, text layer, metadata, embedded objects, revision history: and content survives in every layer the redaction did not touch. Real redaction therefore runs as a pipeline. **Mark:** the redaction regions defined against the fixed rendering (which is why guide 88 §6 images for redaction: fixity is the precondition). **Burn in:** the marked content *removed* from the output: pixels replaced, the underlying text deleted from the text layer, not merely obscured: using redaction-grade tooling whose output genuinely lacks the content, never drawing tools whose rectangles lift off. **Sanitise:** the non-visible layers cleaned: document metadata, embedded files and objects, comments, revision data, thumbnails: the fields guide 87 taught carrying exactly what redaction protects. **Verify:** the output tested as an attacker would: text extraction over redacted regions, copy-paste attempts, metadata dumps, the guide 86-style text layer read: on a sampled or per-document basis scaled to sensitivity, with results logged. The unredacted originals are preserved behind every redaction (guide 88's route back), the conversion is flagged per the protocol, and the audit record: who redacted what, with which tool, verified how: is the answer when a redaction is challenged, and the exhibit if one fails anyway and guide 92 begins.

- **Fixity first:** redact on the fixed rendering, per guide 88's imaging-for-cause: dynamic formats are sanitised differently (§5).
- **Burn-in, not overlay:** the content deleted from the output's data, provably: rectangles that lift off are the canonical failure.
- **Sanitise every layer:** metadata, embeds, comments, revisions, thumbnails: the invisible strata cleaned deliberately.
- **Verify adversarially:** extract, copy, dump: your own production attacked before the opponent does it for you.
- **Record everything:** tool, operator, verification results, preserved originals: the audit file the challenge meets.

§ 0 2 · FIRST PRINCIPLES

# The problem in plain English: covering is not removing

On paper, a black marker destroys what it covers: the ink and the information occupy the same physical layer. Digital documents broke that intuition: what you see is a rendering of underlying data, and drawing a black rectangle changes the rendering while leaving the data intact: the text still in the text layer for extraction, still in the file for anyone who selects, copies, or opens the document in a different tool. Decades of public failures: court filings, government releases, corporate disclosures: are all this one error, made by people who trusted their eyes.

The stakes are the full set: privileged content reaching the opponent (guide 92's machinery, self-triggered); personal data breaching UK GDPR with ICO consequences; commercially sensitive terms escaping confidentiality rings; and the credibility cost of a production whose redactions peel. The cure is entirely procedural: use tools that remove, sanitise the layers the eye never sees, test the output as an adversary, and write down what was done. None of it is difficult; all of it is skippable under deadline pressure, which is why the checklist exists and why the verification step is a gate, not a virtue.

§ 0 3 · THE HIDING PLACES

# Where content hides: text layers, metadata, embeds and revisions

- **The text layer:** the searchable text beneath the rendering (guide 86's subject): the classic failure's home: rectangles drawn over pages whose text layer still carries every redacted word, yielded by extraction or copy-paste in seconds.
- **Document metadata:** authors, titles, custom fields, comments and tracked changes (guide 87's families): the "redacted" settlement agreement whose title field still names the confidential counterparty: sanitisation's core targets.
- **Embedded objects:** files inside files: the spreadsheet embedded in the Word report carrying its full data behind a pasted chart; attachments inside PDFs; OLE objects: each a complete document the page-level redaction never saw.
- **Revision and version data:** deleted text recoverable from tracked changes and prior versions retained in the file: the paragraph "removed" before production, still present in the revision stream.
- **Images' own layers:** EXIF and embedded thumbnails preserving the pre-crop, pre-redaction picture; and pixelation or blurring of text, which modern reconstruction increasingly defeats: image redaction means solid replacement, never obfuscation.

§ 0 4 · THE PIPELINE

# How real redaction works: burn-in, sanitisation and the pipeline

- **Fix the rendering:** the document rendered to its production form (guide 88 §6's imaging-for-cause), so what is redacted is what is produced: the region marked against a stable page.
- **Apply with redaction-grade tooling:** purpose-built functions that delete content within regions: pixels replaced, text-layer entries removed, the output file demonstrably lacking the data: platform redaction modules and professional PDF tools qualify; drawing markup, highlighting and image annotation never do.
- **Sanitise the file:** the §3 strata cleaned: metadata stripped or reviewed field-by-field, embedded objects removed or separately handled, revision history flattened, thumbnails regenerated: the tool's sanitisation pass run and its report kept.
- **Re-OCR the output where needed:** the redacted document's new text layer generated clean (guide 86's disciplines), so the production is searchable minus the removed content: searchability preserved, secrets not.
- **Preserve and flag:** unredacted natives sealed behind the production (guide 88's route back; guide 91 §6's ring machinery where applicable), the conversion and redaction flagged per the protocol, and the audit entry written: document, regions, basis, tool, operator, verification.

§ 0 5 · THE HARD FORMATS

# The hard formats: spreadsheets, natives, images and media

- **Spreadsheets:** guide 88 §4's forcing class, forced again: cell values feed formulas, hidden sheets and pivot caches retain source data, and "deleting" a cell can leave its value computable from dependents: native spreadsheet redaction means removing values *and* auditing dependencies and caches, with the operation logged per guide 88 §6's partial-native method: or, where agreed, imaging the relevant sheets for cause.
- **Word-processing natives:** where native production is owed (guide 88's default) but content must go: text removed, revision history flattened, metadata sanitised, and the file re-saved clean: with the change flagged, because an edited native is a conversion of a different kind and the receiving party is told.
- **Images and scans:** solid-fill replacement of regions, EXIF and thumbnail sanitisation, output re-encoded: no blurring, no pixelation, per §3's reconstruction warning.
- **Audio and video:** bleeping and blanking performed on re-encoded output (not container-level edits that preserve original streams), with the redacted spans logged: guide 77's media disciplines meeting this one.
- **Chat and structured exports:** redaction at the export-generation stage where possible (fields and messages excluded by query) rather than post-hoc editing of rendered transcripts: guide 71's reproducibility making the redaction a documented query, the cleanest form the discipline takes.

§ 0 6 · THE GATE

# Verification: testing your own redactions before they leave

- **Attack your own output:** text extraction over every redacted region (the guide 86 tooling pointed at your own production); select-and-copy attempts; metadata dumps read field by field; embedded-object enumeration: the adversary's first hour, run by you first.
- **Scale to sensitivity:** per-document verification for privilege and high-stakes redactions; sampled verification at stated rates for volume personal-data redactions: proportionality applied, and recorded, per the series' standing pattern.
- **Independent-tool checks:** the output opened in tools other than the one that made it: failures hide inside the producing tool's own rendering assumptions.
- **The verification log:** per document or batch: tests run, results, tester, date: the §1 audit record's second half, and the exhibit that separates a process failure from a process that never existed if guide 92 arrives anyway.
- **The pre-production join:** verification integrated with guide 91 §6's privilege sweep: one outbound gate, two disciplines, every production.

§ 0 7 · THE RECEIVING CHAIR

# Receiving redacted productions: audit, challenge and recovery ethics

- **Intake checks, extended:** guide 88 §7's audit gains a redaction pass: redactions flagged as agreed, bases stated, and the redacted documents' own layers checked: not to harvest, but to know what you hold.
- **The failed-redaction dilemma:** where the opponent's redaction visibly fails (text extracts, rectangles lift): the safe course mirrors guide 92 §5: stop, quarantine, notify: deliberately mining a failed redaction for privileged content is the Example 2 of guide 92 with the roles chosen rather than stumbled into, and professional-conduct duties bite.
- **Challenging over-redaction:** redactions must have stated bases (privilege, agreed irrelevant-confidential categories): patterns of unexplained redaction, or redactions swallowing plainly relevant content, met with §11's requests and, where needed, inspection applications: the court can look behind the ink.
- **Personal-data redactions:** UK GDPR-driven redactions checked for consistency both ways: the same data class redacted here and produced there is a question in either direction.
- **The neutral option:** contested redactions examined by an independent specialist against the unredacted originals under ring terms: guide 71 §7's agreed-specialist pattern, again.

§ 0 8 · THE WIDER MAP

# Source architecture: where else the evidence lives

Per this series' source-architecture discipline, read both ways here. **For the redacting party**, the map is a threat model: the content being removed from this copy usually exists elsewhere: duplicates across custodians (guide 82's tables), thread instances quoting the passage (guide 83), the DMS versions and backups (guide 75, the backup guides), counterpart copies already in other hands: so redaction scope is planned across the estate (every instance the production touches gets the same treatment, per the propagation discipline), and confidentiality claims are made knowing what the other side may already hold. **For the receiving party**, the same map answers legitimate needs without touching the ink: where a redacted passage's subject matter is genuinely disclosable, the route is the unredacted original under ring terms, the same content in an unredacted duplicate or thread instance properly produced, the version ladder, or the counterpart copy: the content reached through disclosure machinery, never through the redaction's failure. The map that makes redaction hard to do consistently is the same map that makes over-redaction hard to sustain: both sides read it, and the estate keeps everyone honest.

| WHERE THE CONTENT ALSO LIVES | DUPLICATES (GUIDE 82) | THREAD INSTANCES (GUIDE 83) | DMS / VERSIONS (GUIDE 75) | BACKUPS / ARCHIVES | COUNTERPART COPIES | UNREDACTED ORIGINALS (RING) |
|---|---|---|---|---|---|---|
| **Redactor's task** | Propagate treatment | Redact every quoting hop | Scope the ladder | Know what persists | Claim realistically | Preserve + seal |
| **Recipient's legitimate route** | The produced duplicate | The produced thread | Version production requests | n/a | Your own client's copy | Y: inspection machinery |
| **Consistency check** | Same passage, same ink | Y: the classic gap | Redacted here, open there | n/a | Reveals over-claiming | The court's yardstick |
| **Failure consequence** | One missed copy leaks all | The unredacted quote | The open earlier version | Restoration surprises | Already out | n/a |

Fallback strategy in one line: plan redactions across the estate, not the page: one passage, every instance, same treatment: and reach contested content through machinery, never through the ink's failure.

§ 0 9 · IN THE WILD

# Worked examples

### EXAMPLE 1 · THE RECTANGLES THAT LIFTED OFF

A defendant produced a settlement-adjacent agreement with pricing terms "redacted": black rectangles drawn in a general-purpose PDF editor's markup layer. The claimant's intake audit (§7's pass) found the failure in minutes: select-all-copy yielded the full text, and the markup objects could be deleted outright, restoring the page. The claimant's team did the disciplined thing: stopped, quarantined, notified: and the defendant's scramble ran guide 92's §4 playbook against its own production. Re-production with burn-in tooling followed, with the original "redaction" tool's identity in the incident correspondence. Costs were agreed against the producing party, whose provider's workflow: markup, not removal: had never included a verification step. The §6 gate exists because this example is the most common redaction failure in circulation, and the cheapest to prevent.

### EXAMPLE 2 · THE THREAD REDACTED IN ONE PLACE AND QUOTED IN ANOTHER

A producing party redacted its GC's advice from an email under a proper burn-in workflow: and produced, forty documents later in the same tranche, a later thread member quoting the identical advice in full, coded by a different reviewer who never connected the two. The §8 propagation discipline had run at document level, not estate level: the redaction was flawless; the scope was not. The receiving party notified rather than exploited (the guide 92 calculus applying to content however it arrives), the quoting instance was re-produced redacted, and the producing party's workflow gained the thread-propagation step this guide's §8 table describes: one passage, every instance, same ink. The example is why redaction planning starts from guide 82's tables and guide 83's trees, not from a list of documents.

### EXAMPLE 3 · THE SPREADSHEET THAT COMPUTED ITS OWN SECRETS

A native spreadsheet production redacted individual salary figures from an HR model under an agreed personal-data protocol: the cells cleared, the file re-saved. The verification pass (§6, run properly this time) caught what the clearing had missed: a summary sheet's subtotals and a surviving pivot cache allowed most cleared values to be recomputed or read directly, and a hidden reconciliation sheet carried the original figures untouched. The redaction was re-engineered per §5: dependent formulas converted to values, the pivot cache purged, hidden sheets reviewed and cleaned, the dependency audit logged: and the incident, caught inside the gate, never left the building. The lesson is the forcing-class lesson of guide 88 §4 completed: spreadsheets are computation, not pages, and their redaction is an engineering task with a dependency graph, not an ink task with a rectangle.

§ 10 · WHERE IT GOES WRONG

# Common mistakes and technical limitations

## Common mistakes

- **Markup as redaction:** Example 1's rectangles: drawing tools doing a removal tool's job.
- **Text layer forgotten:** the image redacted, the searchable text untouched: extraction defeating the ink.
- **Metadata unsanitised:** the protected name in the title field, the advice in a comment: guide 87's families skipped.
- **Estate-blind scope:** Example 2's quoting thread: propagation run at page level, not corpus level.
- **Spreadsheets treated as pages:** Example 3's caches and dependents: the computation left computing.
- **Pixelation and blurring:** obfuscation offered where replacement was owed: reconstruction's easy targets.
- **No verification gate:** productions leaving untested under deadline: the §6 hour, skipped exactly when it mattered.
- **Originals not preserved:** the unredacted native discarded: guide 88's route back demolished, and the redaction now unauditable.

## Technical limitations

- **Tools vary in what they clean:** sanitisation coverage differs by product and version: the tool's actual behaviour verified per matter, not assumed from its marketing.
- **Complex formats resist certainty:** exotic embeds and container formats can carry strata standard passes miss: high-sensitivity documents get specialist review.
- **Redaction shape can leak:** region sizes and positions hint at content length and structure: rarely material, occasionally (short names, numbers) worth padding conventions.
- **Media redaction is span-based:** re-encoding removes the spans; contextual inference from what remains is not preventable by tooling: scope decisions, not tool decisions.
- **Verification proves the tested paths:** the §6 battery covers known attack classes: stated as such, per the standing honesty rule: which is why tooling choice and sanitisation discipline carry the load verification confirms.

§ 11 · INTERROGATORIES & DRAFTING AIDS

# Questions to ask · Suggested wording

## Ask your client

- What are we redacting and why, per category: privilege, personal data, agreed confidential-irrelevant: so bases are stated and consistent?
- Where else does each redacted passage live: duplicates, threads, versions, counterpart hands: for the §8 scope plan?
- Are the unredacted originals preserved and sealed, with the ring machinery agreed where needed?

## Ask your opponent

- Please confirm the tooling and methodology by which redactions in your production were applied, including that redacted content has been removed from the text layer and that document metadata, embedded objects and revision data have been sanitised, and describe your verification process.
- Please state the basis of each redaction (or category of redactions) in [documents], and confirm that unredacted originals are preserved and available under the agreed inspection machinery where a redaction is challenged.
- Please confirm that redactions have been applied consistently across duplicates and thread instances of the same content within your production.

## Ask your eDiscovery / forensic provider

- Does the pipeline burn in and sanitise: and what does the tool's sanitisation pass actually cover, verified how?
- Has the estate-level propagation run: every instance of each redacted passage, same treatment?
- What did the verification battery return on this production: and is the log ready to exhibit?

### SUGGESTED WORDING · REDACTION PARAGRAPH FOR THE DISCLOSURE PROTOCOL

> "Redactions shall be applied using tooling that removes the redacted content from the produced document, including from any text layer, and each producing party shall sanitise document metadata, embedded objects, revision history and thumbnails of redacted documents prior to production. Redactions shall be indicated as such, with the basis stated by category, and shall be applied consistently across duplicates and thread instances of the same content. Blurring, pixelation or markup overlays shall not constitute redaction. Each party shall verify its redactions prior to production by methods including text extraction over redacted regions and metadata review, and shall maintain a record of redactions applied and verification performed. Unredacted originals shall be preserved for the duration of the proceedings and made available under [the confidentiality ring / paragraph X] where a redaction is challenged. An inadvertently ineffective redaction shall be treated as an inadvertent disclosure under paragraph [Y]."

§ 12 · QUICK CONTROL

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

## The redaction checklist

- ☐ Redaction-grade tooling selected; markup tools banned from the workflow
- ☐ Rendering fixed before regions are marked (guide 88 §6)
- ☐ Burn-in confirmed: content absent from output data, not hidden
- ☐ Sanitisation pass run: metadata, embeds, revisions, thumbnails
- ☐ Estate-level propagation: every duplicate and thread instance treated
- ☐ Spreadsheets engineered: dependents, caches, hidden sheets audited
- ☐ Output re-OCRed clean where searchability is owed
- ☐ Verification battery run and logged; independent-tool check included
- ☐ Unredacted originals preserved and sealed; ring machinery agreed
- ☐ Bases stated; conversions and redactions flagged per protocol

## Red flags

- Redactions that select, copy or extract: Example 1's lift-off.
- The same passage open elsewhere in the production: Example 2's thread.
- Spreadsheet redactions with live formulas nearby: Example 3's caches.
- Blurred or pixelated text offered as redaction.
- Productions with no stated redaction bases.
- "We no longer have the unredacted version."
- Opposing methodologies that cannot name their tool or verification step.

## When to involve a digital forensic expert

At workflow design, where tooling, sanitisation coverage and the verification battery are engineered per format (§4-§6); at scope, where the estate-level propagation plan is built from guides 82-83's structures; at the gate, where every production's redactions are attacked before exchange; on the receiving side, where intake checks, over-redaction challenges and the neutral-examination role of §7 run; and on failures, where guide 92's machinery takes over with this guide's records as its evidence, under CPR Part 35 where required. Through **e-discovery.uk**, redaction runs as engineering with an audit trail: ink that cannot be lifted because there is nothing under it.

§ 1 3 · COMMON QUESTIONS

# Frequently asked questions

### We drew black boxes in our PDF tool. Is that not redaction?

Almost certainly not: general-purpose markup draws over content and leaves it in the file: Example 1's lift-off is the standard demonstration. Redaction is the removal function of redaction-grade tooling, confirmed by the §6 tests: if select-and-copy over the region yields text, it is not redacted.

### Do we really need to worry about metadata on redacted documents?

Yes: titles, authors, comments, tracked changes and embedded objects routinely carry exactly what the page-level redaction removed: guide 87's families are the redactor's threat list. The sanitisation pass is one command in proper tooling; its absence is one metadata dump from being a guide 92 incident.

### Can we redact a spreadsheet by deleting the sensitive cells?

Not safely alone: dependents recompute, caches retain, hidden sheets persist: Example 3's trio: so native spreadsheet redaction is a dependency-audited engineering task (values converted, caches purged, hidden content reviewed, the operation logged), or the affected sheets are imaged for cause under guide 88 §6. The page intuition is the error; the file is a computation.

### The other side's redaction has failed and we can read the content. Can we use it?

Treat it as guide 92 §5 treats any privileged arrival: stop, quarantine, notify: CPR 31.20's logic and professional-conduct duties reach content obtained through a redaction's failure, and deliberate mining converts an opponent's error into your own conduct problem. Contested redactions are challenged through §7's machinery: the court can look behind the ink; you should not.

### Is blurring or pixelating text an acceptable redaction?

No: obfuscation preserves information that reconstruction techniques increasingly recover, and the protocol wording of §11 excludes it by name: image-region redaction means solid replacement with sanitised output. The visual softness that makes blurring feel gentler is precisely the data it leaves behind.

### How do we prove our redactions are sound if challenged?

From the records this guide has you keep: the tooling and pipeline description, the sanitisation coverage, the verification log with its per-document or sampled results, the preserved originals available under ring terms: the challenge met with files, per the series' standing principle. A sound redaction with no record argues; a recorded one exhibits.

§ 14 · REFERENCE

# Glossary

**Burn-in**
Removal of redacted content from the output's data, not just its rendering.

**Dependency audit**
Tracing spreadsheet formulas and caches that could rebuild cleared values.

**Lift-off failure**
Markup "redaction" removable to restore the content.

**Obfuscation**
Blurring or pixelation; information-preserving and non-compliant.

**Pivot cache**
A spreadsheet's retained copy of source data behind pivot tables.

**Propagation (redaction)**
The same treatment applied to every instance of a passage.

**Redaction-grade tooling**
Software whose redaction function deletes content from the file.

**Sanitisation**
Cleaning metadata, embeds, revisions and thumbnails from output.

**Text layer**
The searchable text beneath a document's rendering.

**Verification battery**
The adversarial tests run on redacted output before production.

## Sources and authoritative references

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

- e-discovery.uk, EDRM Phase 06 · Production (redaction pipelines, verification gates and audit records as practised): e-discovery.uk/edrm/production · Phase 05 · Review: e-discovery.uk/edrm/review
- cflab.uk, digital forensics services (failed-redaction examination, recovery analysis, CPR Part 35 reporting): cflab.uk/digital-forensics-services · guides library: cflab.uk/guides
- e-discovery.uk, practice method and Knowledge Centre: e-discovery.uk · e-discovery.uk/knowledge
- ICO, guidance on disclosing information safely (redaction and hidden data): ico.org.uk, disclosing information safely · UK GDPR guidance: ico.org.uk
- PD 57AD (form of production; redaction indication): justice.gov.uk, PD 57AD · CPR Part 31, r.31.20: justice.gov.uk, Part 31
- CPR Part 35: justice.gov.uk, Part 35 · ISO/IEC 27037: iso.org, 27037

DISCLAIMER

This publication provides general information about digital redaction as at August 2026. Tool behaviour and sanitisation coverage vary by product and version; verify the current position for the tooling in the matter at hand. The worked examples are fictional. This guide is not legal advice, does not create a professional relationship, and should not be relied upon in place of advice on the facts of a specific matter from a qualified lawyer or a suitably qualified forensic practitioner. Links were live at the date of publication.

§ HOW A SPECIALIST LABORATORY CAN ASSIST

# Working with Computer Forensics Lab

Redaction at **e-discovery.uk** is engineered, gated and recorded: burn-in tooling with verified sanitisation coverage, estate-level propagation planned from the duplicate tables and thread trees, the hard formats handled as the computations they are (Example 3's dependency audit as standard for native spreadsheets), and the verification battery attacking every production before exchange: with the audit log ready to exhibit if anyone asks, and guide 91's privilege sweep running through the same gate. The laboratory side examines failed redactions in either direction: demonstrating a failure's mechanics for a challenge, or evidencing a sound process against one: and serves as the neutral examining contested redactions against sealed originals under ring terms, reporting under CPR Part 35. Conflicts checks, confidential scoping discussions and fixed-fee estimates are routine; and if a redacted production is due out this month, the §6 battery is this week's gate: your own adversary first, while it is free.

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