



C O M P U T E R F O R E N S I C S L A B

London digital forensics & e-discovery · Established 2007

iOS Encrypted Apps and Secure Messengers

Signal, Telegram and Wickr: What Encryption Protects, What It Does Not, and What the Device Still Gives Up

Secure messengers such as Signal, Telegram and Wickr are built to keep conversations private, and they do so well. But their reputation for being "unrecoverable" is widely misunderstood, and the misunderstanding cuts both ways. What these apps protect is the message in transit and, in the strongest cases, at rest against anyone but the endpoints. What they cannot protect is the message from the very device it is displayed on: if a conversation is readable on the phone, it is, in principle, recoverable from the phone, subject to the app's own on-device protections. Some apps store far more than users assume; others store almost nothing, or delete aggressively; and features like disappearing messages genuinely limit what survives. The forensic task is to understand each app precisely, what it encrypts, what it stores locally, what it deletes, and how, and to recover honestly from the endpoint what the encryption never hid there. This guide sets out for UK lawyers how secure messengers work, what they do and do not protect, what the device still yields, and how their evidence is recovered and its limits stated.

© Estimated reading time: 25 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. Secure-messenger analysis is a specialist strand of its mobile work: understanding precisely what Signal, Telegram, Wickr and similar apps encrypt, store and delete, recovering from the endpoint what encryption never hid there, and stating honestly what each app does and does not yield, including where disappearing messages and aggressive deletion genuinely limit recovery. The analysis is reported under CPR Part 35 and CrimPR Part 19, with the reality of each app, not its reputation, driving the conclusions.

ESTABLISHED 2007 · LONDON

ISO 17025-ALIGNED PROCEDURES

SECURE-MESSENGER FORENSICS

ENDPOINT RECOVERY & HONEST LIMITS

CPR PART 35 EXPERT REPORTS

FULL CHAIN-OF-CUSTODY DOCUMENTATION

§ CONTENTS

In this guide

- 01 Executive summary
- 02 The problem in plain English: encryption protects the wire, not the endpoint
- 03 What secure messengers protect, and how
- 04 What the device still holds
- 05 Where recovery genuinely stops
- 06 Understanding each app precisely
- 07 Deployment: recovering from the endpoint, honestly
- 08 Source architecture: where else the evidence lives
- 09 Worked examples
- 10 Common mistakes and technical limitations
- 11 Questions to ask · Suggested wording
- 12 Checklist and red flags · When to involve a digital forensic expert
- 13 Frequently asked questions
- 14 Glossary · References · Disclaimer · How a specialist laboratory can assist

§ 0 1 · ORIENTATION

Executive summary

The headline point: secure messengers like Signal, Telegram and Wickr protect messages in transit and against interception, not from the very device that displays them, so if a conversation is readable on the phone it is in principle recoverable from the phone, subject to each app's own on-device protections and deletion, which vary greatly.

What they protect (§3): the message on the wire and against interception, through end-to-end encryption, a genuine and strong protection. **What the device holds (§4):** if a message is readable on the phone, it exists decrypted there and is, in principle, recoverable from the endpoint, subject to the app's local storage and protections. **Where recovery stops (§5):** disappearing messages, aggressive local deletion and strong on-device encryption genuinely limit what survives, honest limits that vary by app. **Understanding each app (§6):** Signal, Telegram and Wickr differ greatly in what they store and delete, so each is understood precisely rather than by reputation. **Deployment (§7):** recovering from the endpoint what encryption never hid there, and stating honestly what each app does not yield.

- **Encryption protects the wire:** the message in transit and against interception, not the endpoint.
- **Readable means recoverable:** if it is on the phone's screen, it exists on the phone.
- **Apps vary enormously:** some store much, others little, with different deletion.
- **Some limits are real:** disappearing messages and aggressive deletion genuinely limit recovery.
- **Reality, not reputation:** each app is understood precisely, not by its "unrecoverable" name.

Encryption protects the wire; the endpoint still displays the message

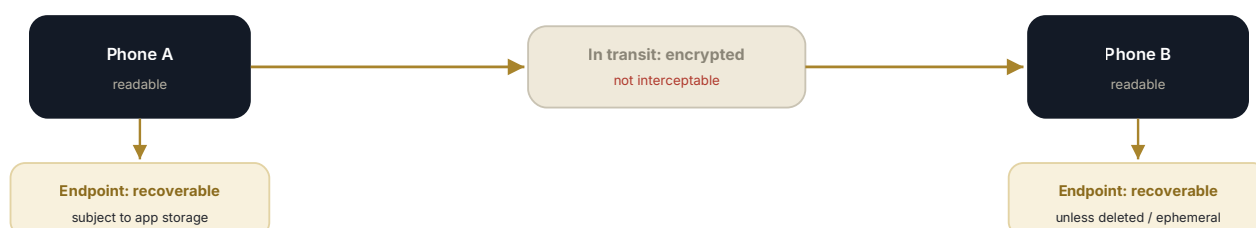


Figure 1 - end-to-end encryption protects the message in transit; but if it is readable on either endpoint, it exists there decrypted and is, in principle, recoverable, subject to each app's storage and deletion.

The problem in plain English: encryption protects the wire, not the endpoint

Secure messengers, Signal, Telegram, Wickr and their kin, have a formidable reputation for privacy, and much of it is deserved. They are also widely misunderstood, and the misunderstanding matters because it leads both to false despair (assuming nothing can ever be recovered) and to false confidence (assuming these apps hide everything). The key is to be precise about what their encryption actually protects. End-to-end encryption protects the message while it travels between the sender and recipient, and against anyone trying to intercept it in transit or obtain it from the service in the middle. That is a real and strong protection: these messages generally cannot be read off the wire or handed over by the provider in readable form.

But there is a simple, decisive point that the reputation obscures: a message that a person can read on their phone must exist, in readable form, on that phone. Encryption protects the message from the outside world; it does not, and cannot, protect the message from the very device that decrypts and displays it to its user. So if a conversation is visible on the screen of an iPhone, it exists somewhere on that iPhone in a form the phone can render, and is therefore, in principle, recoverable from the phone, subject only to how the particular app stores and protects its own data locally. This endpoint reality is what makes secure-messenger forensics possible at all. It is, however, only half the picture, because the apps differ enormously in what they actually keep. Some store a great deal locally, in databases much like any other app; others store very little, encrypt their local data strongly, or delete aggressively. And features designed precisely to defeat recovery, disappearing messages that vanish after a time, aggressive deletion, ephemeral modes, genuinely limit what survives. So the honest position is neither "unrecoverable" nor "always recoverable", but "it depends on the app, precisely understood". The forensic task is to know each app, what it encrypts, what it stores, what it deletes and how, and to recover from the endpoint what the encryption never hid there, while stating honestly where recovery genuinely stops. This guide sets out how.

§ 0 3 · WHAT THEY PROTECT

What secure messengers protect, and how

- **The message in transit:** end-to-end encryption protecting the message as it travels, so it cannot readily be intercepted on the wire, a genuine and strong protection (guide 148): the in-transit protection.
- **Against the provider:** the strongest apps designed so the provider itself cannot read the messages, so a demand on the provider yields little readable content, unlike ordinary services (guides 111 to 114): the provider protection.
- **Local data, variably:** some apps also encrypting their data at rest on the device, more or less strongly, so on-device protection varies greatly between apps (§5): the varying local protection.
- **Metadata, variably:** apps differing in how much metadata they retain and protect, so even where content is protected, some contact metadata may survive (guide 153): the metadata question.
- **Not the endpoint display:** the crucial limit, that no app can protect a message from the device that decrypts and displays it, so the endpoint is where recovery is possible (§2): the endpoint exception.

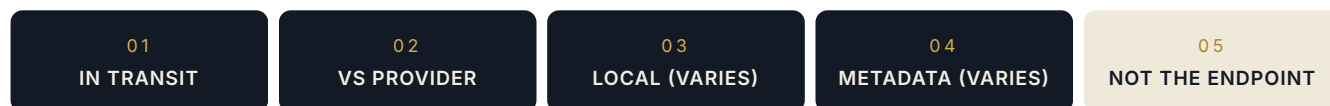


Figure 2 - secure messengers protect the message in transit and against the provider, protect local data and metadata variably, but cannot protect the message from the endpoint that displays it.

What the device still holds

- **Readable means present:** a message readable on the phone existing in readable form on the phone, so it is, in principle, recoverable from the endpoint whatever the encryption protected in transit (§2): the endpoint principle.
- **Local databases and stores:** many apps keeping their conversations in local databases and stores on the device, recovered like any app data (guides 151, 155), so the content is reached at the endpoint: the local store.
- **Attachments, caches and previews:** media, caches, notification previews and thumbnails that apps leave on the device, so content survives even where a main store is protected (guide 138): the peripheral traces.
- **Deleted app records:** deleted messages within these apps surviving in database remnants and journals (guide 155), and in backups, so a deleted secure-messenger conversation is often recoverable: the deleted-record recovery.
- **Reached lawfully and at depth:** the app data recovered from the acquired, decrypted, unlocked device at sufficient depth (guides 145, 146, 133), lawfully, so the endpoint is examined properly: the sound endpoint recovery.

§ 0 5 · WHERE RECOVERY STOPS

Where recovery genuinely stops

- **Disappearing messages:** messages set to vanish after a time genuinely removing content from the device, so where they have expired there may be nothing to recover, an honest limit (guide 148): the ephemeral limit.
- **Aggressive local deletion:** some apps deleting their data aggressively and securely, so little survives locally, another real constraint that varies by app (§6): the deletion limit.
- **Strong local encryption:** apps that encrypt their local store strongly and tie it to credentials, so without the passphrase or key the local data may be inaccessible (guide 133): the local-encryption limit.
- **Provider yields little:** for the strongest apps, the provider holds little readable content, so a demand on the provider is not a route to the messages (§3): the provider limit.
- **State the limit honestly:** where recovery genuinely stops, the limit stated plainly rather than overpromised, so expectations rest on the reality of the specific app (guide 100): the honest limit.

Understanding each app precisely

- **Reality, not reputation:** each app assessed on what it actually encrypts, stores and deletes, not on its general reputation for security, since the reality varies enormously (guide 151): the precise assessment.
- **Signal, Telegram, Wickr differ:** these and similar apps differing greatly, in local storage, deletion, ephemerality and cloud use, so each is understood on its own terms rather than lumped together (§4, §5): the per-app understanding.
- **Cloud versus local:** some apps (for example certain Telegram modes) keeping data in their own cloud, a further source reachable by lawful process, while others keep everything local, so the storage model is established (guides 111 to 114): the storage model.
- **Version and configuration:** what an app yields depending on its version and the user's settings (disappearing timers, passphrases), so the specific configuration is understood (§5): the configuration check.
- **Validate the interpretation:** the understanding of each app validated by analysis and testing, not assumed, so conclusions rest on how the app truly behaves (guide 151): the validated understanding.

Deployment: recovering from the endpoint, honestly

- **Recovering readable conversations:** conversations readable on the phone recovered from the endpoint, so content the parties believed protected by encryption is obtained where the app stored it locally (§4): the endpoint content recovered.
- **Recovering deleted secure messages:** deleted messages within these apps recovered from remnants, journals and backups, so a deleted secure-messenger conversation is restored where it survives (guide 155): the deleted secure message restored.
- **Corroborating from the counterpart:** the counterpart's device holding the same conversation, so recovery from one endpoint is corroborated, or achieved, from the other (guide 148): the other endpoint.
- **Stating honest limits:** where disappearing messages, aggressive deletion or strong local encryption genuinely prevent recovery, the limit stated plainly, so the account is truthful (§5): the honest boundary.
- **Corroborated, honest presentation:** the recovered content corroborated across endpoints and the estate, attributed with care, and presented at honest confidence with the app's limits stated, under CPR Part 35 or CrimPR Part 19 (guides 100, 130): the secure-messenger evidence deployed defensibly.

Source architecture: where else the evidence lives

Per this series' source-architecture discipline: a secure-messenger conversation, though protected in transit, exists at both endpoints and sometimes elsewhere, so recovery draws on the whole estate rather than despairing at the encryption. The phone holds the app's local store, caches, previews and deleted remnants, where the app keeps them. But the counterpart's device holds the same conversation, reachable by disclosure; backups and iCloud (guide 147) hold copies and prior states where the app is included, and an encrypted backup may hold more; APFS snapshots (guide 132) hold earlier states; the paired Mac and iPad (guide 135) hold the app's data where the app runs there; and, for apps that use their own cloud (some Telegram modes), the provider holds data by lawful process, though for the strongest apps the provider holds little readable content. The discipline is to recover from the endpoint where the app stores content, corroborate from the counterpart and the estate, and reach any app cloud lawfully, while stating honestly where ephemerality or deletion has genuinely removed the content. When one endpoint has deleted or expired the message, the counterpart may retain it; when the local store is protected, a backup or the counterpart may hold it; and the honest limit is reached only after the whole estate has been considered.

EVIDENCE	THE PHONE (ENDPOINT)	COUNTERPART DEVICE	BACKUPS / ICLOUD	APFS SNAPSHOTS	APP CLOUD (SOME APPS)	PROVIDER
Readable message	Y: if stored locally	Y: their copy	Y: if app included	Y: at snapshot	Some (e.g. cloud chats)	Little (strong apps)
Deleted message	Remnants / journals	Their copy	Pre-deletion backup	At snapshot	Varies	Little
Attachments / previews	Y: caches	Their copy	In backup	At snapshot	Cloud media	n/a
Ephemeral (expired)	Often gone	Maybe (their copy)	Maybe (if backed up)	Maybe	Varies	Little
Metadata	Some	Some	In backup	At snapshot	Some	Some

Fallback strategy in one line: recover from the endpoint where the app stores content, corroborate from the counterpart, backups and the estate, reach any app cloud lawfully, and state the honest limit only where ephemerality or deletion has genuinely removed the content.

Worked examples

EXAMPLE 1 · THE "UNRECOVERABLE" APP RECOVERED FROM THE ENDPOINT

A party argued that conversations in a secure messenger could never be produced because the app was end-to-end encrypted. The §2 endpoint principle answered this: the messages were readable on the party's phone, so they existed in readable form there, and a §4 recovery of the app's local store on the acquired, decrypted device produced the conversations in full (guide 155). The §3 encryption had protected the messages in transit and from the provider, but not from the endpoint that displayed them. The §8 counterpart's device corroborated them. The claim of unrecoverability misunderstood what the encryption protected. The lesson is §2's: end-to-end encryption protects the message in transit and against interception, not from the very device that decrypts and displays it, so a readable secure-messenger conversation is, in principle, recoverable from the endpoint, and a claim that such an app is simply unrecoverable is usually wrong.

EXAMPLE 2 · THE DISAPPEARING MESSAGES THAT WERE GENUINELY GONE

In another matter, key exchanges had used disappearing messages set to a short timer. The §5 honest-limit discipline governed: where the messages had expired, they had genuinely been removed from the device, and there was often nothing to recover, a real limit, not a failure of technique (guide 100). Rather than overpromise, this was stated plainly, and the §8 estate pursued for what remained, a few messages preserved in an earlier backup before expiry, and the §8 counterpart's device where their copy survived. The honest position, that some content was genuinely gone, was given, alongside what could be recovered. The lesson is §5's: features like disappearing messages genuinely limit recovery, so where content has expired the limit is stated honestly rather than overpromised, and the estate and counterpart are pursued for what survives, giving a truthful account of both what is gone and what remains.

EXAMPLE 3 · THE THREE APPS THAT BEHAVED QUITE DIFFERENTLY

A phone held three different secure messengers, and the §6 precise-understanding discipline showed how differently they behaved. One stored its conversations in a straightforward local database, fully recovered from the endpoint. Another kept much of its data in its own cloud, reached by §8 lawful process (guides 111 to 114). The third deleted aggressively and encrypted its local store strongly, so little survived without the passphrase, an §5 honest limit. Treating all three as "the same secure app" would have produced wrong conclusions in every direction. Each was §6 understood and validated on its own terms. The lesson is §6's: secure messengers differ enormously in what they store, delete and encrypt, so each is assessed on its actual behaviour, reality not reputation, rather than lumped together, since only a precise, per-app understanding yields the true position on what each does and does not yield.

Common mistakes and technical limitations

Common mistakes

- **Assuming "unrecoverable":** the false-despair error, treating an end-to-end app as impossible to recover when the endpoint holds the readable messages (Example 1, §2).
- **Assuming everything hides:** the false-confidence error, assuming these apps conceal all content when many store much locally (§4).
- **Lumping apps together:** treating all secure messengers as the same, when they behave quite differently (Example 3, §6).
- **Overpromising past real limits:** claiming recovery where disappearing messages or aggressive deletion have genuinely removed content (Example 2, §5).
- **Ignoring the counterpart:** not pursuing the other endpoint, which holds the same conversation (§8).
- **Missing deleted secure records:** not recovering deleted app messages from remnants and backups (§4, guide 155).
- **Not understanding the app's version and settings:** ignoring the configuration that governs what survives (§6).
- **Attributing carelessly:** a recovered conversation attributed to a person without care (§7, guide 105).

Technical limitations

- **Ephemeral content is genuinely gone:** expired disappearing messages may leave nothing to recover: a real limit (§5).
- **Aggressive deletion limits survival:** some apps keep little locally (§5).
- **Strong local encryption may block access:** without the passphrase or key, the local store may be inaccessible (§5).
- **Provider yields little for strong apps:** a demand on the provider is not a route to content (§3).
- **Apps evolve constantly:** what an app stores and deletes changes with versions, so analysis is current-aware (§6).

§ 11 · INTERROGATORIES & DRAFTING AIDS

Questions to ask · Suggested wording

Ask your client

- Which secure messengers hold the relevant conversations, and are they readable on the device, so endpoint recovery is possible?
- Were disappearing messages or aggressive deletion in use, and is the counterpart's device available?
- Has the phone been preserved promptly, so app remnants and any ephemeral content in backups survive?

Ask your opponent

- Please preserve and disclose the data of the secure-messenger apps at Schedule 1 on the device, including local stores, caches, deleted records and any app-cloud data, and any backups and paired devices.
- Please confirm whether disappearing messages or deletion were used, and when.
- Please provide any passphrase needed for a strongly-encrypted local store.

Ask your eDiscovery / forensic provider

- For each app, what does it actually store, delete and encrypt, and what can be recovered from the endpoint?
- Where does recovery genuinely stop (ephemeral, deletion, encryption), and where does the counterpart or estate help?
- How is the recovered content attributed and its completeness and limits stated?

SUGGESTED WORDING · INSTRUCTION FOR A SECURE-MESSENGER ANALYSIS

"We instruct you to examine the secure-messenger applications at Schedule 1 on the device relevant to Schedule 2. Please, on the acquired, decrypted, unlocked device at sufficient depth and preserving integrity: (1) for each app, establish precisely what it encrypts, stores locally, keeps in its own cloud, and deletes, assessing it on its actual behaviour and configuration rather than its reputation, and validating that understanding; (2) recover from the endpoint the readable conversations, attachments, caches and previews the app stores locally, and deleted messages from remnants, journals and backups; (3) reach any app-cloud data lawfully, and state where the provider holds little for the strongest apps; (4) state honestly where recovery genuinely stops, disappearing messages that have expired, aggressive local deletion, or a strongly-encrypted local store inaccessible without a passphrase, rather than overpromising; and (5) corroborate the recovered content from the counterpart's device and the estate, attribute it with care, and present it at honest confidence with the app's limits clearly stated, recognising that encryption protects the wire, not the endpoint."

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

The secure-messenger checklist

- ☐ Relevant secure-messenger apps identified
- ☐ Phone preserved promptly; ephemeral content in backups considered
- ☐ Each app assessed on its actual behaviour, not reputation
- ☐ Readable conversations recovered from the endpoint
- ☐ Attachments, caches, previews and deleted records recovered
- ☐ App-cloud data reached lawfully where used
- ☐ Genuine limits (ephemeral, deletion, encryption) stated honestly
- ☐ Counterpart device and estate used to corroborate or recover
- ☐ Content attributed with care
- ☐ Completeness, limits and confidence stated honestly

Red flags

- An end-to-end app assumed simply unrecoverable: Example 1.
- All secure messengers treated as the same: Example 3.
- Recovery overpromised past disappearing messages: Example 2.
- The endpoint not examined despite readable messages.
- The counterpart's device ignored.
- Deleted secure records not recovered.
- An app's version and settings not understood.

When to involve a digital forensic expert

Whenever secure messengers are in play, because their reputation misleads in both directions and only precise, app-by-app understanding gives the true position: the expert establishes, for each app, what it actually encrypts, stores and deletes, assessing reality not reputation (Example 3, §6), and recovers from the endpoint the readable conversations, attachments and deleted records the app keeps locally, because encryption protects the wire, not the device that displays the message (Example 1, §2, §4). Where recovery genuinely stops, expired disappearing messages, aggressive deletion, a strongly-encrypted local store, the limit is stated honestly rather than overpromised, and the counterpart's device and the estate are pursued for what survives (Example 2, §5, §8). Any app-cloud data is reached lawfully, the content is attributed with care and corroborated, and findings, with the app's limits stated, are reported under CPR Part 35 or CrimPR Part 19. This guide continues the iPhone and iPad block. Through **cflab.uk** and **e-discovery.uk**, secure-messenger work replaces the myth of the "unrecoverable" app with the reality: encryption protects the message in transit, but the endpoint that displays it usually gives it up, and where a genuine limit is reached, that limit is stated with equal honesty.

§ 13 · COMMON QUESTIONS

Frequently asked questions

Aren't apps like Signal simply unrecoverable?

No, that misunderstands what the encryption protects (§2, Example 1). End-to-end encryption protects the message in transit and against interception, and, for the strongest apps, from the provider, but it cannot protect the message from the very device that decrypts and displays it. If a conversation is readable on the phone, it exists in readable form on the phone, and is, in principle, recoverable from the endpoint, subject to how the app stores its data locally. So a claim that such an app is simply unrecoverable is usually wrong.

So can you always recover secure-messenger conversations?

No, and claiming so would be the opposite error (§5, Example 2). While the endpoint usually holds readable messages, some content is genuinely beyond recovery: disappearing messages that have expired may leave nothing behind, some apps delete aggressively and keep little locally, and some encrypt their local store strongly so that without a passphrase it is inaccessible. The honest position is neither "always" nor "never" but "it depends on the app and its use", precisely understood, with the genuine limits stated plainly alongside what can be recovered.

Do these apps all behave the same way?

No, they differ enormously (§6, Example 3). Signal, Telegram, Wickr and others vary greatly in what they store locally, what they keep in their own cloud, how aggressively they delete, and whether and how they use disappearing messages. Treating them as interchangeable produces wrong conclusions. Each is assessed on its actual behaviour and the user's configuration, validated by analysis, rather than on its general reputation, so the true position on what a particular app on a particular phone does and does not yield is established precisely.

Can deleted secure-messenger messages be recovered?

Often yes, where they survive (§4, guide 155). Because these apps store their data in databases like other apps, deleted messages frequently persist in the databases' free space and journals and in backups, and can be recovered and reassembled, unless the app deletes securely or the content was ephemeral and has expired. So deletion within a secure messenger is not necessarily final, and the same SQLite recovery that reaches other deleted app data reaches deleted secure-messenger conversations, subject to the app's particular deletion behaviour.

If we can't get it from the phone, is there another way?

Often the counterpart's device (§8). A conversation exists at both endpoints, so where one party's phone has deleted or cannot yield the content, the other party's device usually holds their copy, reachable by disclosure. Backups, snapshots and, for apps that use their own cloud, the provider (by lawful process) may also hold data. For the strongest apps the provider holds little readable content, so the counterpart and the estate are the realistic routes. The whole estate is considered before any conclusion that content is genuinely gone.

Does using a secure messenger look suspicious?

Not in itself, and it is treated neutrally (§6). Many people use secure messengers for entirely ordinary privacy reasons, so their use is not, by itself, evidence of wrongdoing, and the analysis draws no such inference. What matters is the content and pattern of the communications, recovered and interpreted on the evidence, and the honest position on what each app yields. The app choice is context, not proof, and is assessed as neutrally as any other feature of how a person used their phone.

§ 1 4 · REFERENCE

Glossary

Secure messenger

An app built for private, encrypted messaging.

End-to-end encryption

Encryption protecting messages in transit.

Endpoint

A sender's or recipient's device, where messages are readable.

Local store

An app's data kept on the device.

Disappearing message

A message set to expire and be removed.

Ephemeral

Content designed to leave little or nothing behind.

App cloud

An app's own cloud storage (used by some apps).

Provider

The service behind the app; may hold little for strong apps.

Aggressive deletion

An app removing its data quickly and securely.

Reality not reputation

Assessing an app on how it actually behaves.

Sources and authoritative references

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

- cflab.uk, digital forensics services (secure-messenger forensics, endpoint recovery, honest-limit reporting, CPR Part 35 and CrimPR Part 19 reporting): cflab.uk/digital-forensics-services · guides library: cflab.uk/guides
- e-discovery.uk, EDM Phase 04 · Collection and Phase 06 · Analysis (encrypted-app recovery 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
- Apple Platform Security guide (app data protection and sandboxing on iOS): support.apple.com, Platform Security
- Forensic Science Regulator, Code of Practice v2 (interpretation and honest reporting of limits) and ISO/IEC 27037: gov.uk, [FSR Code](https://gov.uk/fsr-code) · iso.org, 27037
- PD 57AD and CPR Part 31 · CPR Part 35 · UK GDPR: justice.gov.uk, PD 57AD · justice.gov.uk, Part 35 · ico.org.uk

DISCLAIMER

This publication provides general information about iOS encrypted apps and secure messengers as at August 2026. These apps and their storage, encryption and deletion behaviours change constantly and differ greatly between apps, so each must be assessed on its actual behaviour and configuration: end-to-end encryption protects the message in transit and against interception, not from the endpoint that displays it, but disappearing messages, aggressive deletion and strong local encryption genuinely limit recovery. 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

Secure-messenger work at the laboratory replaces the myth of the "unrecoverable" app with the reality, in both directions. Because end-to-end encryption protects the message in transit and against interception but cannot protect it from the device that decrypts and displays it, a conversation readable on the phone is, in principle, recoverable from the endpoint, and, on the acquired, decrypted, unlocked device at sufficient depth (guides 145, 146, 133), the readable conversations, attachments, caches and previews an app stores locally are recovered, along with deleted messages from remnants, journals and backups (Example 1, guide 155). Crucially, each app, Signal, Telegram, Wickr and others, is assessed on its actual behaviour and configuration rather than its reputation, because they differ enormously in what they store, keep in their own cloud, and delete (Example 3), and any app-cloud data is reached lawfully. Where recovery genuinely stops, expired disappearing messages, aggressive deletion, a strongly-encrypted local store, the limit is stated honestly rather than overpromised, and the counterpart's device and the estate are pursued for what survives (Example 2). The content is attributed with care, corroborated across endpoints and the estate, and reported at honest confidence with the app's limits clearly stated, under CPR Part 35 or CrimPR Part 19. This guide continues the iPhone and iPad block. Conflicts checks, confidential scoping discussions and fixed-fee estimates are routine; and the guiding truth is simple: encryption protects the wire, the endpoint usually gives up the message, and where a real limit is reached it is stated with equal honesty.



I N S T R U C T T H E L A B

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