COMPUTER FORENSICS LAB

London digital forensics & e-discovery · Established 2007

Android Messaging: SMS, RCS and Messaging Apps

Where Conversations Live on Android, Why the Messaging App Matters, and What Deletion Leaves Behind

On an iPhone, messaging runs through one Messages app that handles both iMessage and SMS. On Android there is no single equivalent, and that difference shapes the evidence. The carrier channels, SMS and its richer successor RCS, are handled by whichever messaging app the user or manufacturer chose, and different apps store them differently; Google Messages, which increasingly dominates, adds end-to-end encryption to RCS and its own cloud backup. Alongside sit the third-party apps, WhatsApp, Signal, Telegram, Messenger, each its own world. For the lawyer, three things follow. The messaging app in use must be identified, because it decides where the messages are and how they are stored. Deleted messages very often survive, in the app databases, in backups, in the Google account and on the counterpart's device. And the same disciplines that governed iPhone messaging, edits, deletion, encryption at the endpoint, attribution, carry over in full. This guide sets out for UK lawyers where Android messaging lives, how SMS, RCS and app messaging differ, what survives deletion, and how conversations are recovered and read honestly.

⏱ 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. Android messaging recovery is central to its mobile work: identifying the messaging app in use, recovering SMS, RCS and third-party app conversations from their databases, recovering deleted messages from remnants, backups, the Google account and counterpart devices, and interpreting encryption, edits and deletion honestly. The analysis attributes conversations with care and is reported under CPR Part 35 and CrimPR Part 19, with the messaging app's particular behaviour understood.

ESTABLISHED 2007 · LONDON | ISO 17025-ALIGNED PROCEDURES | SMS, RCS & APP MESSAGING | DELETED-MESSAGE RECOVERY | CPR PART 35 EXPERT REPORTS | FULL CHAIN-OF-CUSTODY DOCUMENTATION

§ CONTENTS

In this guide

§ 0 1 · ORIENTATION

Executive summary

The headline point: Android has no single Messages app, so SMS and RCS are handled by whichever messaging app is in use, each storing them differently, alongside the third-party apps, which means the messaging app must be identified because it decides where the messages are, while deleted messages very often survive in databases, backups, the Google account and counterpart devices.

SMS, RCS and the app (§3): the carrier channels, SMS and its richer successor RCS, handled by the messaging app the user or maker chose, with Google Messages increasingly dominant and adding end-to-end encryption to RCS. **Where it lives (§4):** in the messaging app's own database, in each third-party app's store, and in backups and the Google account. **What survives (§5):** deleted messages persist in SQLite remnants, recently-deleted areas, backups, the Google account and on the counterpart's device. **Encryption, edits and attribution (§6):** end-to-end RCS is recovered at the endpoint, edits and deletion read honestly, and conversations attributed to people with care. **Deployment (§7):** recovering and proving conversations across every channel.

- **No single Messages app:** the messaging app in use decides where SMS and RCS live.
- **RCS is the new SMS:** richer, often end-to-end encrypted, backed up to Google.
- **Third-party apps alongside:** WhatsApp, Signal, Telegram, each its own world.
- **Deleted often survives:** in databases, backups, the Google account and counterparts.
- **Endpoint recovery holds:** encryption protects the wire, not the phone that displays it.

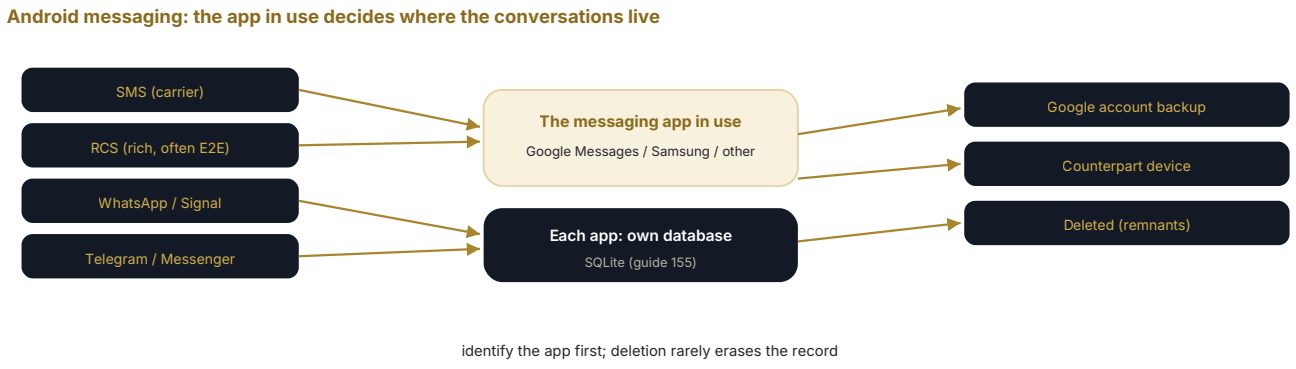

Figure 1 - SMS, RCS and third-party apps each reach the phone through a messaging app with its own database, replicated to the Google account, counterparts and remnants, so the app is identified first and deletion rarely erases the record.

§ 0 2 · FIRST PRINCIPLES

The problem in plain English: no single Messages app

The iPhone messaging guide (guide 148) could start from a fixed point: one Messages app handles both iMessage and ordinary texts, and its database is where the core conversations live. Android offers no such fixed point, and the difference matters for the evidence. The carrier channels, SMS and its richer modern successor RCS, are not handled by a single built-in app but by whichever messaging app is set as the default: Google Messages on many devices, Samsung's own app on many Samsung phones, or another the user chose. Each of these stores messages in its own way, in its own database, with its own features and quirks. And RCS itself has changed the picture: it carries rich media and read receipts, Google Messages adds end-to-end encryption to it, and Google backs up messages to the user's Google account (guide 164), so an RCS conversation on a modern Android device behaves less like an old text message and more like an iMessage.

Alongside the carrier channels sit the third-party messaging apps, WhatsApp, Signal, Telegram, Messenger and the rest, each its own world with its own storage, exactly as on iOS (guides 151, 156). So for the lawyer three things follow. First, the messaging app in use must be identified at the outset, because it decides where the SMS and RCS messages are, how they are stored, whether they are encrypted, and where they are backed up; examining the wrong app, or assuming a fixed location, misses the conversations. Second, deletion is rarely final. Android messaging apps store their conversations in SQLite databases, so deleted messages frequently survive in the databases' free space and journals (guide 155), in the Google account's backups, in local and manufacturer backups, and on the other party's device. Third, every discipline from the iPhone messaging guides carries over unchanged: end-to-end encryption protects the message in transit but not on the endpoint that displays it (guide 156), edits and unsending mean the current view may not be the original, and a message comes from an account or number and is attributed to a person only with care. The task, then, is to identify the app, recover the conversations across every channel, recover what was deleted, and read the result honestly. This guide sets out how.

§ 0 3 · SMS, RCS AND THE APP

SMS, RCS and the messaging app in use

- **SMS: the carrier text:** ordinary text messages carried by the mobile network, stored by the messaging app, and also reflected in carrier records, the traditional channel (guide 153): the plain text.
- **RCS: the richer successor:** RCS carrying rich media, read receipts and typing indicators over data, increasingly the default for Android-to-Android messaging, so modern "texts" are often RCS (§2): the modern text.
- **End-to-end encrypted RCS:** Google Messages encrypting RCS end-to-end between Google Messages users, so these messages resist interception but remain readable, and recoverable, on the endpoints (guide 156): the encrypted text.
- **The default app decides:** the messaging app set as default (Google Messages, Samsung Messages, another) handling SMS and RCS and storing them in its own database, so identifying it is the first step (§4): the app that governs.
- **Backed up to Google:** Google Messages backing up messages to the Google account, and manufacturer apps to their clouds, so the account holds a copy reachable even when the device is sealed (guide 164): the cloud copy.

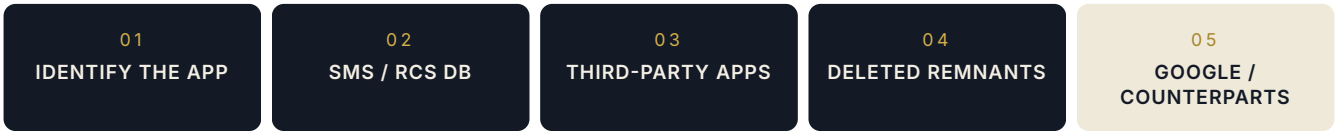

Figure 2 - identify the messaging app, recover its SMS and RCS database and each third-party app's store, recover deleted remnants, and corroborate from the Google account and counterparts.

§ 0 4 · WHERE IT LIVES

Where Android messaging lives

- **The messaging app's database:** SMS and RCS stored in the default messaging app's own database, so the core carrier conversations sit in that app's store, reached at sufficient depth (guide 162): the carrier-message store.
- **The system message store:** Android also maintaining a system-level store of SMS and MMS that apps read and write, so messages may be present there as well as in the app's own database (§3): the shared system store.
- **Third-party app stores:** WhatsApp, Signal, Telegram and others each keeping their own databases and media, examined app by app on their own terms (guides 151, 156): the per-app record.
- **Attachments and media:** the images, videos and files sent in conversations stored alongside or within the app data, so the full content is recovered (guide 166): the media of the conversation.
- **Depth matters:** app databases and deleted data reached only by a file-system or deeper extraction, so the acquisition depth governs what is seen (guide 162): the depth-dependent record.

§ 0 5 · WHAT SURVIVES DELETION

What survives deletion

- **Deleted messages in databases:** deleted messages persisting in the free space and write-ahead logs of the messaging and app databases, so a deleted conversation is often reconstructed from the database itself (guide 155): the message that outlived deletion.
- **The Google account backup:** messages backed up to the Google account holding copies and earlier states, so a message deleted since the backup survives in the cloud (guide 164): the account copy.
- **Local and manufacturer backups:** local backups and manufacturer clouds (Samsung Cloud and equivalents) holding earlier states, so a pre-deletion backup restores the record (guide 161): the preserved prior state.
- **The counterpart device:** the other party holding their copy, so a message deleted on one phone survives on the other, reachable by disclosure (guide 148): the other side of the exchange.
- **Carrier records for SMS:** the carrier retaining SMS metadata (and, for a period, content in some cases) independently of the phone, so SMS contact survives device deletion (guide 153): the independent carrier record.

§ 0 6 · ENCRYPTION, EDITS AND ATTRIBUTION

Encryption, edits, attribution and honest limits

- **Endpoint recovery holds:** end-to-end encrypted RCS and app messages readable on the phone existing there decrypted, so they are recoverable from the endpoint whatever the encryption protected in transit (guide 156): the wire-not-endpoint principle.
- **Edits, unsending and disappearing:** messages edited, unsent or set to disappear, so the current view may differ from the original and some content may be genuinely gone, read honestly (guides 148, 156): the altered and vanishing record.
- **Account, not person:** a message coming from a number or account, attributed to a person with corroboration, since devices and accounts may be shared (guide 105): the who-sent-it question kept honest.
- **App behaviour understood:** each messaging and third-party app understood on its own terms, since storage, deletion and backup behaviour differ by app and version (guide 151): the per-app understanding.
- **State confidence plainly:** content, timing, attribution and completeness expressed at the confidence the recovered data supports, with the limits of ephemeral and deleted content stated (guide 100): the honest messaging account.

§ 0 7 · DEPLOYMENT

Deployment: recovering and proving conversations

- **Proving what was said:** recovered SMS, RCS and app conversations establishing the content of communications, including messages a party deleted (guide 148): the communication proved.
- **Proving timing and sequence:** message metadata dating and ordering the exchange on a timeline, corroborated with carrier records (guides 96, 153): the chronology established.
- **Exposing deletion and alteration:** deleted or edited messages recovered against a party's account, so a pruned or altered chain is corrected (guide 155): the concealment undone.
- **Reaching encrypted messaging at the endpoint:** end-to-end RCS and app conversations recovered from the device where interception was impossible (guide 156): the encrypted channel opened at the device.
- **Corroborated, honest presentation:** conversations corroborated across the phone, the Google account, backups, carrier records and counterparts, and presented at honest confidence with attribution stated, under CPR Part 35 or CrimPR Part 19 (guides 100, 130): the messaging evidence deployed defensibly.

§ 0 8 · THE WIDER MAP

Source architecture: where else the evidence lives

Per this series' source-architecture discipline: an Android conversation exists in several places, and identifying the messaging app is what tells the enquiry where to look. The phone holds the messaging app's database, the system message store and each third-party app's store, with deleted remnants. But the Google account (guide 164) holds backed-up messages and earlier states reachable regardless of the phone's lock; the manufacturer's cloud holds its own message backups; local backups hold prior states; the counterparts hold their side of every exchange; the carrier holds SMS records and call-detail data independently of the device; and the app providers hold metadata by disclosure. The discipline is to identify the app in use, recover its conversations and those of every other app on the device, and then reconstruct each conversation across the estate, so that a message deleted, unsent or edited on the phone is caught by its presence, or earlier form, in the account, a backup, the carrier's records or the counterpart's device. When the phone is sealed, the Google account and counterparts hold the messages; when a message is deleted, a pre-deletion backup or the counterpart retains it; and the sources together reconstruct the genuine exchange.

EVIDENCE	THE PHONE	GOOGLE ACCOUNT	MANUFACTURER / LOCAL BACKUP	COUNTERPART DEVICE	CARRIER RECORDS	DELETED / RECOVERABLE
SMS	Y: app + system store	Y: if backed up	Y: if backed up	Y: their copy	Y: metadata, some content	Often (guide 155)
RCS	Y: app db	Y: Google Messages backup	Some	Y: their copy	Limited (data channel)	Often (remnants)
App messages	Y: app stores	Varies (app backups)	Varies	Y: their copy	n/a	Often (remnants)
Attachments	Y: stored	Y: if backed up	Y	Y: their copy	n/a	Varies
Edited / unsent	Current + remnants	Earlier backup state	Earlier state	Their received version	n/a	Original may survive

Fallback strategy in one line: identify the messaging app, recover every channel on the phone, and reconstruct each conversation across the Google account, backups, carrier records and counterparts; when the phone is sealed or a message deleted, those sources usually hold it or its earlier form.

§ 0 9 · IN THE WILD

Worked examples

EXAMPLE 1 · THE MESSAGES THAT WERE IN A DIFFERENT APP

An examiner looked for a party's texts in the expected place and found little. The §3 identify-the-app discipline found the reason: the device's default messaging app was not the one assumed, and its §4 SMS and RCS conversations sat in that app's own database, not where the examiner had looked; the §4 system message store held a partial view only. Once the correct app was identified and its database recovered at §4 sufficient depth (guide 162), the conversations, including deleted ones from §5 remnants (guide 155), emerged in full. The messages had been there all along, in the app in use. The lesson is §2's: Android has no single Messages app, so the messaging app in use is identified first, because it decides where SMS and RCS live and how they are stored, and examining the wrong location, or assuming a fixed one, misses the conversations entirely.

EXAMPLE 2 · THE DELETED RCS THREAD AND THE GOOGLE BACKUP

A party deleted an RCS conversation and, the phone being modern and sealed (guide 163), the device gave up nothing. The §5 estate recovered it: the §3 Google Messages backup in the party's Google account, reached §8 lawfully (guide 164), held the conversation from before the deletion, and the §8 counterpart's device held the other side. The §6 end-to-end encryption of the RCS had protected it in transit but not in the backup or on the endpoints. The deleted thread was reconstructed without the phone. The lesson is §5's: Android messaging is backed up to the Google account and held by counterparts, so a message deleted on a sealed phone frequently survives in the account backup or on the other party's device, reachable regardless of the phone's lock, and end-to-end encryption does not remove it from the endpoints or the backup.

EXAMPLE 3 · THE THREE APPS AND THE ONE THAT WAS TRULY GONE

A phone held a messaging app, WhatsApp and a secure messenger, and a party relied on conversations across all three. The §6 per-app discipline distinguished them: the messaging app's RCS and WhatsApp's conversations were fully recovered from their databases and backups; but the secure messenger had used disappearing messages that had expired, and, per §6, that content was genuinely gone from the device, an honest limit (guide 156), though the §8 counterpart's copy retained some of it. Each app was understood on its own terms and reported truthfully. Two channels yielded fully; the third yielded honestly. The lesson is §6's: the messaging apps on a device differ in storage, backup and deletion behaviour, so each is understood on its own terms, deleted and ephemeral content is pursued across the estate, and where content is genuinely gone the limit is stated plainly rather than overpromised.

§ 10 · WHERE IT GOES WRONG

Common mistakes and technical limitations

Common mistakes

- **Assuming a fixed messaging location:** the cardinal error, looking in the wrong app or store for SMS and RCS (Example 1, §2).
- **Treating RCS like old SMS:** not recognising its encryption, backup and app-dependence (§3).
- **Assuming deletion is final:** ignoring remnants, the Google account, backups and counterparts (Example 2, §5).
- **Thinking encryption blocks recovery:** assuming end-to-end RCS or app messages cannot be produced from the endpoint (§6, guide 156).
- **Shallow extraction:** relying on a logical extraction that misses app databases and deleted data (§4, guide 162).
- **Examining only one channel:** the messaging app but not the third-party apps, or vice versa (§4).
- **Overpromising past real limits:** claiming recovery where disappearing messages have expired (Example 3, §6).
- **Attributing carelessly:** a message from a number treated as certainly from a person (§6).

Technical limitations

- **App variation:** storage, backup and deletion differ by app and version, so each is understood specifically (§3, §6).
- **Depth-dependent:** app databases and remnants need a file-system or deeper extraction (§4).
- **Ephemeral content may be gone:** expired disappearing messages may leave nothing (§6).
- **Backup depends on settings:** whether messages reached the Google account depends on the user's choices (§5).
- **Account is not identity:** attribution to a person needs care (§6).

§ 11 · INTERROGATORIES & DRAFTING AIDS

Questions to ask · Suggested wording

Ask your client

- Which messaging app is the default on the device, and which third-party apps carry the relevant conversations?
- Might messages have been deleted, edited or unsent, and was messaging backed up to the Google account?
- Is the counterpart's device available, and has the phone been preserved promptly?

Ask your opponent

- Please identify the default messaging app on the device at Schedule 1 and preserve and disclose its SMS and RCS data, the system message store and all third-party messaging apps, including deleted messages, and any Google account and manufacturer backups.
- Please confirm whether relevant messages have been deleted, edited or unsent, and when.
- Please preserve the relevant carrier records.

Ask your eDiscovery / forensic provider

- Which messaging app is in use, and where do its SMS and RCS conversations live?
- What deleted messages can be recovered from the databases, the Google account, backups and counterparts?
- How is encrypted, edited or ephemeral content handled, and how are conversations attributed?

SUGGESTED WORDING · INSTRUCTION FOR AN ANDROID MESSAGING ANALYSIS

> "We instruct you to recover and analyse the messaging on the Android device at Schedule 1 relevant to Schedule 2, across all channels. Please, on the acquired device at sufficient depth and preserving integrity: (1) identify the default messaging app in use and recover its SMS and RCS conversations from its database and the system message store, and recover the conversations of every third-party messaging app, with attachments, understanding each app's storage, backup and deletion behaviour on its own terms; (2) recover deleted, edited and unsent messages from database remnants and journals, and from the Google account backup, manufacturer and local backups, and the counterpart's device; (3) treat end-to-end encrypted RCS and app messages as recoverable from the endpoint, and state honestly where disappearing messages have genuinely removed content; (4) reconstruct each conversation with participants, timing and reply structure, corroborated with carrier records, and attribute conversations to accounts and, with corroboration, to people; and (5) state content, timing, attribution and completeness at honest confidence."

§ 12 · QUICK CONTROL

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

The Android messaging checklist

- ☐ Default messaging app identified
- ☐ SMS and RCS recovered from the app database and system store
- ☐ Every third-party messaging app recovered, with attachments
- ☐ Extraction depth sufficient for app databases and deleted data
- ☐ Deleted, edited and unsent messages recovered from remnants
- ☐ Google account, manufacturer and local backups used
- ☐ Counterpart device and carrier records used to corroborate
- ☐ Encrypted messaging recovered at the endpoint
- ☐ Ephemeral limits stated honestly
- ☐ Conversations attributed with care; confidence stated honestly

Red flags

- SMS and RCS sought in the wrong app or store: Example 1.
- RCS treated as plain SMS.
- A deleted conversation assumed gone: Example 2.
- Encrypted messaging treated as unrecoverable from the endpoint.
- A shallow extraction relied on for app databases.
- Only one channel examined.
- Recovery overpromised past expired disappearing messages: Example 3.

When to involve a digital forensic expert

Whenever messaging matters on an Android device, and promptly, because remnants are overwritten and the messaging app must be identified before anything else is sought: the expert identifies the default messaging app and recovers its SMS and RCS conversations from its database and the system store, together with every third-party app's conversations, at the depth needed to reach app databases and deleted data (Example 1, §3, §4, guide 162). Deleted, edited and unsent messages are recovered from database remnants and from the Google account, backups and the counterpart's device, so a sealed phone or a deletion rarely defeats the record (Example 2, §5); end-to-end encrypted RCS and app messages are recovered at the endpoint where they sit decrypted (§6, guide 156); each app is understood on its own terms, with expired disappearing content stated honestly as gone (Example 3); and conversations are reconstructed, corroborated with carrier records and counterparts, attributed with care, and reported at honest confidence under CPR Part 35 or CrimPR Part 19. This guide continues the Android block. Through **cflab.uk** and **e-discovery.uk**, Android messaging work finds the

conversations wherever the app in use has put them, recovers what deletion tried to erase, and reads the result with the honesty the evidence demands.

§ 13 · COMMON QUESTIONS

Frequently asked questions

Where are the text messages on an Android phone?

In whichever messaging app is the default, which must be identified first (§2, §3, Example 1). Unlike the iPhone's single Messages app, Android hands SMS and RCS to the app set as default, Google Messages, Samsung Messages or another, and each stores them in its own database in its own way; Android also keeps a system-level message store that gives a partial view. So the app in use is identified at the outset, its database recovered at sufficient depth, and the third-party apps examined alongside, rather than assuming the messages sit in one fixed place.

What is RCS, and does it change anything?

RCS is the richer modern successor to SMS, and it changes a good deal (§3). It carries rich media, read receipts and typing indicators over data rather than the traditional SMS channel, is increasingly the default for Android-to-Android messaging, and in Google Messages is end-to-end encrypted and backed up to the Google account. So a modern Android "text" behaves more like an iMessage than an old SMS: it resists interception, is recoverable from the endpoints, is often backed up to the cloud, and leaves fewer carrier-side traces than SMS did.

Someone deleted their messages. Can they be recovered?

Frequently yes, from several directions (§5, Example 2). Android messaging apps store conversations in SQLite databases, so deleted messages often survive in their free space and journals; messages backed up to the Google account or a manufacturer cloud survive there from before the deletion; local backups hold earlier states; the counterpart's device holds their copy; and for SMS the carrier retains records independently. Even a sealed phone rarely defeats recovery, because the Google account and counterparts are reachable regardless of its lock, provided everything is preserved promptly.

The messages were end-to-end encrypted. Doesn't that stop recovery?

No, it stops interception, not endpoint recovery (§6, guide 156). Encrypted RCS and app messages are protected in transit, but they sit decrypted on the phones that display them, and, for Google Messages, in the account backup, so they are recovered from the endpoint or backup like any other messaging. What genuinely limits recovery is different: disappearing messages that have expired, aggressive deletion by some apps, or a strongly encrypted local store without its passphrase, and where those apply the limit is stated honestly rather than overpromised.

Do we need to look at the third-party apps too?

Yes, always (§4). People spread their conversations across the messaging app and WhatsApp, Signal, Telegram, Messenger and others, each its own world with its own storage, backup and deletion behaviour, so examining only the carrier channels can miss most of the exchange. Every installed messaging app is recovered and understood on its own terms (guides 151, 156), with its attachments, because the relevant conversation may be in any of them, and the apps differ enough that one cannot stand in for another.

Does a message prove who sent it?

It shows the sending number or account; attributing it to a person takes care (§6, guide 105). Phones and accounts may be shared, so a message is attributed to the number or account first and to a person only with corroboration from other evidence and the estate. Edits and unsending also mean an apparent history may be partial or altered. So content and attribution are stated at honest confidence, distinguishing what is certain from what is inferred, rather than assuming the account holder personally sent every message.

§ 14 · REFERENCE

Glossary

SMS
Ordinary text messaging over the carrier network.

RCS
The richer successor to SMS, carried over data.

Default messaging app
The app handling SMS and RCS on the device.

Google Messages
Google's messaging app, with E2E RCS and backup.

System message store
Android's shared SMS/MMS store.

Third-party app
WhatsApp, Signal, Telegram, Messenger and others.

End-to-end encryption
Protection in transit, not at the endpoint.

Remnant
Deleted data recoverable from a database.

Ephemeral message
A disappearing message that limits survival.

Counterpart
The other party to a conversation.

Sources and authoritative references

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

- cflab.uk, digital forensics services (SMS, RCS and app messaging recovery, deleted-message analysis, CPR Part 35 and CrimPR Part 19 reporting): cflab.uk/digital-forensics-services · guides library: cflab.uk/guides
- e-discovery.uk, EDRM Phase 04 · Collection and Phase 06 · Analysis (message recovery and reconstruction 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
- Google, Messages end-to-end encryption and backup documentation, and GSMA RCS specifications: support.google.com, Messages encryption · gsma.com, RCS
- Forensic Science Regulator, Code of Practice v2 (recovery, interpretation and honest reporting) and ISO/IEC 27037: 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

§ HOW A SPECIALIST LABORATORY CAN ASSIST

Working with Computer Forensics Lab

Android messaging work at the laboratory finds the conversations wherever the app in use has put them and recovers what deletion tried to erase. Because Android has no single Messages app, the default messaging app is identified first, and its SMS and RCS conversations are recovered from its own database and the system message store, alongside the conversations of every third-party app, WhatsApp, Signal, Telegram, Messenger and the rest, each understood on its own terms, with attachments, at the extraction depth needed to reach app databases and deleted data (Example 1, guides 162, 151). Deleted, edited and unsent messages are recovered from database remnants and journals (guide 155) and from the Google account backup, manufacturer and local backups and the counterpart's device, so that a sealed phone or a deletion rarely defeats the record (Example 2, guide 164). End-to-end encrypted RCS and app messages are recovered at the endpoint where they sit decrypted, because encryption protects the wire and not the phone (guide 156), and where disappearing messages have genuinely expired the limit is stated honestly (Example 3). Each conversation is reconstructed with its participants, timing and reply structure, corroborated with carrier records and counterparts, attributed to accounts and, with corroboration, to people, handled proportionately under the UK GDPR, and reported at honest confidence under CPR Part 35 or CrimPR Part 19. This guide continues the Android block. Conflicts checks, confidential scoping discussions and fixed-fee estimates are routine; and the guiding rule is the block's own: identify the app, because it decides where the messages are, and then reach across the estate, because deletion rarely erases the record.

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