COMPUTER FORENSICS LAB

London digital forensics & e-discovery · Established 2007

Android Forensics: The Fragmented Landscape

Why No Two Android Phones Are Alike, and What That Means for the Evidence

Where the iPhone is one platform tightly controlled by one company, Android is a landscape: an open operating system shipped by dozens of manufacturers on thousands of models, each with its own hardware, its own modifications, its own security chip, its own version of the software, and its own update history. Samsung, Google, Xiaomi, OnePlus, Huawei and the rest do things differently, and a phone's age, model and patch level can change what is forensically possible far more than they do on an iPhone. This fragmentation is the defining fact of Android forensics. It means there is no single answer to what can be recovered from "an Android phone"; there is only an answer for this phone, this model, this version, this security state. This guide opens the Android block by setting out for UK lawyers why the landscape is fragmented, how it shapes acquisition and analysis, what is consistent across it, and how to think about what an Android device will yield.

⏱ 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 forensics is a major part of its mobile work, spanning the whole fragmented landscape: assessing each device on its own model, version and security state, acquiring it by the method that device permits, and analysing its data in the knowledge that manufacturers store and protect it differently. The analysis is honest about what a particular device will and will not yield, corroborates across the estate, and is reported under CPR Part 35 and CrimPR Part 19.

ESTABLISHED 2007 · LONDON | ISO 17025-ALIGNED PROCEDURES | ANDROID FORENSICS ACROSS MANUFACTURERS | PER-DEVICE FEASIBILITY ASSESSMENT | CPR PART 35 EXPERT REPORTS | FULL CHAIN-OF-CUSTODY DOCUMENTATION

§ CONTENTS

In this guide

§ 0 1 · ORIENTATION

Executive summary

The headline point: Android is not one platform but a fragmented landscape of manufacturers, models, versions and security states, so there is no single answer to what "an Android phone" will yield, only an answer for this device, this model, this version and this security state, which must be assessed before expectations are set.

Fragmentation (§3): dozens of manufacturers, thousands of models, differing hardware and security chips, manufacturer modifications, and a spread of software versions and patch levels. **Acquisition (§4):** what method is possible, and how deep it reaches, depends on the specific device, its version and its security state, far more than on iOS. **Consistency (§5):** beneath the variation, Android shares a common core, app storage in SQLite, a Google account estate, and common artefact families, which the analysis relies on. **Assessing this device (§6):** feasibility is assessed per device and expectations set honestly. **Deployment (§7):** getting the most from the device in hand, by the method it permits, corroborated across the Google and manufacturer estates.

- **A landscape, not a platform:** manufacturers, models and versions all differ.
- **Feasibility is per device:** model, version and security state decide what is possible.
- **Acquisition varies most:** the method and depth depend on the specific phone.
- **A common core remains:** SQLite, the Google account estate, and shared artefacts.
- **Set expectations honestly:** assess this phone before promising what it will yield.

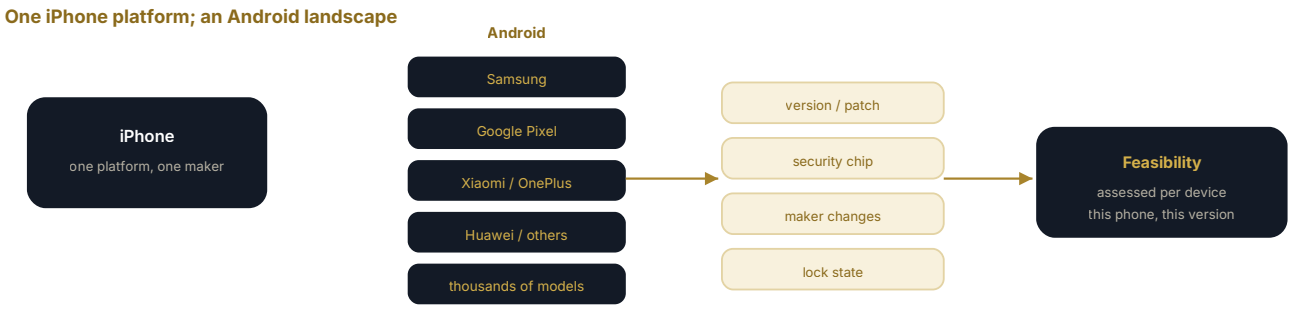

Figure 1 - where the iPhone is one platform, Android is a landscape of makers and models whose version, security chip, manufacturer changes and lock state together decide feasibility, assessed per device.

§ 0 2 · FIRST PRINCIPLES

The problem in plain English: a landscape, not a platform

The preceding block examined the iPhone, and one fact simplified everything about it: an iPhone is one platform, made by one company, running one operating system that the same company controls and updates. Whatever the model, an examiner knows broadly what to expect, and the differences are in degree. Android is the opposite. It is an open operating system that Google develops but that dozens of manufacturers take, modify and ship on thousands of different models. Samsung, Google's own Pixel line, Xiaomi, OnePlus, Huawei and many others each use different hardware, add their own layers and features, choose their own security chips, and ship and update the software on their own schedules. Two Android phones bought in the same shop on the same day can differ, forensically, more than an old and a new iPhone.

This fragmentation is the single most important fact about Android forensics, and it has a direct consequence for the lawyer. There is no such thing as "an Android phone" for evidential purposes. A question like "what can be recovered from an Android device?" has no general answer; it has an answer only for this device, this manufacturer, this model, running this version with this patch level, in this security state, and that answer can range from almost everything to almost nothing. What is routine on one model may be impossible on another, and a phone that yields fully today may become far harder after an update. Acquisition, the method by which the phone's data is obtained and how deep it reaches, is where this variation bites hardest, because the available methods depend intimately on the device's hardware and software. And yet, beneath the variation, there is a common core that the analysis relies on: Android apps store their data in SQLite databases much as iOS apps do, a Google account sits behind most devices as a cloud estate much as iCloud does behind an iPhone, and a family of common artefacts recurs across models. The task, then, is twofold: to understand that the landscape is fragmented and to assess each device individually before setting any expectation, and to recognise the consistent core that makes analysis possible once the data is obtained. This guide, opening the Android block, sets out both.

§ 0 3 · THE SOURCES OF FRAGMENTATION

The sources of fragmentation

- **Manufacturers and models:** dozens of manufacturers and thousands of models, each with different hardware, so the device population is vastly more varied than the iPhone's (§2): the maker-and-model spread.
- **Manufacturer modifications:** each maker layering its own interface, features and system changes over Android, so the same data may be stored or protected differently by maker (guide 151's per-app theme, now per-maker): the customised software.
- **Versions and patch levels:** a wide spread of Android versions and security patch levels in use at once, since makers update on their own schedules, so the same model can differ by its update state (§4): the version spread.
- **Security hardware:** differing security chips and secure elements across makers and generations, governing encryption and lock protections, so what can be defeated or must be authenticated varies (guide 133's Secure Enclave theme): the hardware security variation.
- **Lock and encryption state:** whether the device is locked, its encryption type, and whether the user has enabled certain protections, so the individual phone's state matters as much as its model (guide 145): the device-state factor.

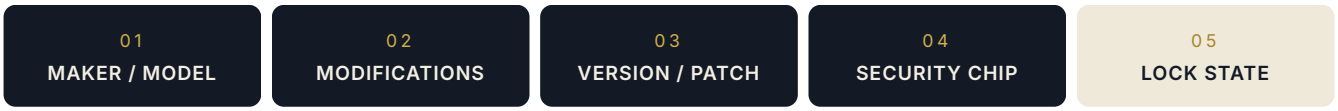

Figure 2 - the sources of Android fragmentation: maker and model, manufacturer modifications, version and patch level, security hardware, and the individual device's lock state.

§ 0 4 · ACQUISITION

How fragmentation shapes acquisition

- **Method depends on the device:** the available acquisition methods, from a simple logical extraction to deeper physical or full-file-system methods, depending on the specific model, version and security state (guide 146): the device-dependent method.
- **Depth varies widely:** some devices yielding a full file system and even deeper, others only a limited logical extraction, so depth, and therefore completeness, varies far more than on iOS (guide 146): the variable depth.
- **Lock state is decisive:** a locked Android device ranging from readily accessible to effectively sealed depending on model, version and protections, so the lock state and any available credential govern what is possible (guide 145): the lock question.
- **Updates change feasibility:** a device's update state altering what methods work, sometimes closing routes that worked before, so feasibility is assessed as of the device's current state (§3): the moving target.
- **Preserve as on iOS:** the same seizure disciplines applying, keeping the device from locking, from the network and from remote wipe, and minimising use (guides 145, 155), since preservation is decisive here too: the constant preservation duty.

§ 0 5 · WHAT IS CONSISTENT

What is consistent across Android

- **SQLite everywhere:** Android apps storing their data in SQLite databases, so the deleted-data recovery disciplines apply across the landscape (guide 155): the common data technology.
- **The Google account estate:** a Google account behind most devices, syncing messages, photos, backups, location history and app data to the cloud, a counterpart to iCloud (guide 158): the common cloud estate.
- **Common artefact families:** recurring families of artefacts, call logs, messages, contacts, usage and activity records, location, across makers, differing in detail but not in kind (guides 153, 154): the shared artefacts.
- **Common app ecosystem:** the same third-party apps, WhatsApp, Signal, Telegram and the rest, running across Android with broadly similar storage, so app disciplines carry over (guides 151, 156): the shared apps.
- **The same disciplines:** preservation, integrity, corroboration, attribution and honest confidence applying unchanged, so the method is constant even where the device is not (guides 2, 100, 130): the constant method.

§ 0 6 · ASSESSING THIS DEVICE

Assessing this device: feasibility and honest expectations

- **Identify precisely:** the device's manufacturer, model, Android version, patch level and security state identified precisely at the outset, since these decide everything (§3): the precise identification.
- **Assess feasibility per device:** what acquisition is possible, and how deep, assessed for this specific device rather than assumed from "Android" in general (guide 145): the per-device assessment.
- **Set expectations honestly:** the client told what this device will and will not yield before work begins, so expectations rest on the device's reality (guide 100): the honest expectation.
- **Consider the estate early:** where the device is constrained, the Google account, backups and estate identified as alternative routes from the start (§8): the early fallback.
- **Reassess as needed:** feasibility revisited if the device's state changes or new methods emerge, so the assessment stays current (§4): the living assessment.

§ 0 7 · DEPLOYMENT

Deployment: getting the most from the device in hand

- **Choose the best available method:** the deepest lawful acquisition this device permits chosen, so the most complete data is obtained (§4, guide 146): the maximised acquisition.
- **Apply the common disciplines:** the shared analysis disciplines applied to the recovered data, SQLite recovery, app analysis, location, activity, so the evidence is read correctly (§5): the disciplined analysis.
- **Fill gaps from the estate:** where the device yields less, the Google account, backups and manufacturer cloud filling the gaps (§8): the estate deployed.
- **State the device's limits:** what this device could not yield stated plainly, so silence is not read as absence (guide 146): the honest limit.
- **Corroborated, honest presentation:** the findings corroborated across the estate and presented at honest confidence with the device's feasibility explained, under CPR Part 35 or CrimPR Part 19 (guides 100, 130): the Android evidence deployed defensibly.

§ 0 8 · THE WIDER MAP

Source architecture: where else the evidence lives

Per this series' source-architecture discipline: an Android device, like an iPhone, is one window onto a wider estate, and because Android feasibility varies so much by device, the estate matters even more, as the route around a constrained phone. The device holds its local data at whatever depth was acquired. But the Google account behind it syncs messages, photos, backups, location history and app data to the cloud, a counterpart to iCloud (guide 158) reachable lawfully; the manufacturer's own cloud (Samsung Cloud and equivalents) holds backups and synced data; local and cloud backups hold earlier states; a paired wearable or tablet holds shared data (guide 157); the counterparts hold their side of communications; and the app providers and carriers hold records by disclosure. The discipline is to map the device and its estate together at the outset, so that where this particular device permits only a shallow acquisition, or is locked beyond reach, the Google account, manufacturer cloud, backups and counterparts supply what the device cannot. When the device yields little, the account often yields much; when the device is sealed, the cloud and counterparts are the route; and the estate makes the fragmented landscape navigable.

EVIDENCE	DEVICE (DEPTH VARIES)	GOOGLE ACCOUNT	MANUFACTURER CLOUD	BACKUPS	PAIRED / COUNTERPART	PROVIDER / CARRIER
Messages	Y: if reached	Y: if synced	Y: if backed up	Y	Y: counterpart	Provider metadata
Photos	Y	Y: Google Photos	Y: gallery sync	Y	Shared copies	n/a
Location	Y: multi-source	Y: location history	Some	Some	Wearable	Carrier cell (area)
App data	Depth-dependent	Y: app backups	Some	Y	Counterpart	App provider
Deleted data	Remnants (if deep)	Earlier states	Earlier backups	Y: pre-deletion	Their copy	Provider retention

Fallback strategy in one line: map the device and its estate together at the outset; where this device yields little or is sealed, the Google account, manufacturer cloud, backups and counterparts supply what the fragmented device cannot.

§ 0 9 · IN THE WILD

Worked examples

EXAMPLE 1 · THE TWO ANDROIDS THAT BEHAVED NOTHING ALIKE

A matter involved two Android phones, and a party assumed they would yield the same. The §3 fragmentation said otherwise: one, an older model on an outdated version, §4 permitted a deep acquisition that recovered a full file system including deleted data (guide 155); the other, a recent flagship fully patched with a modern security chip, §4 permitted only a limited logical extraction. The §6 per-device assessment identified each precisely and set honest expectations for both, and for the constrained phone the §8 Google account and manufacturer cloud filled much of the gap. Treating them as "two Android phones" would have misled about both. The lesson is §2's: there is no general answer for "an Android phone", only an answer for this device, this version and this security state, so each is identified and assessed individually, and expectations are set from the device's reality rather than from the platform's name.

EXAMPLE 2 · THE UPDATE THAT CLOSED THE DOOR

A device had been assessed as amenable to a deep acquisition, but before it was examined it received a system update, and the §4 method that had worked no longer did; the update had closed the route. The §6 living assessment caught this, reassessing feasibility as of the device's new state and switching to the best method now available, while the §8 estate, the Google account and a pre-update backup, supplied data the shallower method could not reach (guide 147). The §7 device's new limits were stated honestly. The moving target had moved, and the approach moved with it. The lesson is §4's: on Android, feasibility is a moving target, an update can close a route that worked the day before, so feasibility is assessed as of the device's current state, preservation prevents avoidable change, and the estate is identified early as the route around a device whose door has closed.

EXAMPLE 3 · THE COMMON CORE THAT MADE THE ANALYSIS POSSIBLE

A device from an unfamiliar manufacturer, with its own heavily modified software, appeared daunting. The §5 common core made it tractable: once the data was acquired by the method this device permitted, its apps stored their data in §5 SQLite, so the deleted-record recovery of guide 155 applied; the §5 Google account behind it held synced messages, photos and location history; and the §5 familiar artefact families, calls, messages, usage and location, were present, differing in detail from other makers but not in kind. The unfamiliar surface concealed a familiar core. The lesson is §5's: beneath Android's fragmentation lies a consistent core, SQLite storage, the Google account estate, common artefact families and the same third-party apps, so that once the data is obtained, the established disciplines apply across the landscape, and an unfamiliar manufacturer's device is analysed on familiar foundations.

§ 10 · WHERE IT GOES WRONG

Common mistakes and technical limitations

Common mistakes

- **Treating "Android" as one thing:** the cardinal error, assuming what one device yielded another will (Example 1, §2).
- **Promising before assessing:** setting expectations before the device's model, version and state are identified (§6).
- **Letting the device update or change:** allowing an update to close an acquisition route before examination (Example 2, §4).
- **Reading shallow silence as absence:** treating a limited logical extraction's gaps as proof of nothing (§7, guide 146).
- **Ignoring the Google and manufacturer estate:** examining a constrained device alone when the cloud holds more (§8).
- **Being daunted by an unfamiliar maker:** not recognising the common core beneath the modified surface (Example 3, §5).
- **Neglecting preservation:** letting the device lock, wipe or be used, as damaging on Android as on iOS (§4).
- **Not reassessing:** relying on an old feasibility view when the device's state has changed (§6).

Technical limitations

- **Feasibility varies enormously:** from almost everything to almost nothing by device: the core limit (§2).
- **Modern devices are often constrained:** recent, patched flagships may permit only shallow acquisition (§4).
- **Updates change the picture:** feasibility is a moving target (§4).
- **Manufacturer variation complicates analysis:** the same data may be stored differently by maker (§3).
- **The landscape evolves constantly:** new models, versions and protections appear continually (§3).

§ 11 · INTERROGATORIES & DRAFTING AIDS

Questions to ask · Suggested wording

Ask your client

- Exactly which Android device is it, manufacturer, model, version and patch level, and is it locked, so feasibility can be assessed for this phone?
- Has it been preserved from locking, the network, remote wipe and updates, and is its use minimised?
- Is the Google account, manufacturer cloud and any backup available, as the route around a constrained device?

Ask your opponent

- Please preserve the device at Schedule 1 without use, updates or changes to its state, and disclose its manufacturer, model, version, patch level and lock state.
- Please preserve and disclose the associated Google account, manufacturer cloud data and any backups.
- Please disclose the acquisition method and depth of any Android evidence you rely on.

Ask your eDiscovery / forensic provider

- Having identified this device precisely, what acquisition is feasible and how deep will it reach?
- What will this device not yield, and what does the Google and manufacturer estate supply instead?
- Has feasibility been reassessed as of the device's current state?

SUGGESTED WORDING · INSTRUCTION FOR AN ANDROID FEASIBILITY ASSESSMENT AND ACQUISITION

> "We instruct you to assess and acquire the Android device at Schedule 1 relevant to Schedule 2. Please: (1) identify the device precisely, manufacturer, model, Android version, security patch level, security hardware and lock and encryption state, and preserve it from locking, the network, remote wipe, updates and use; (2) assess, for this specific device rather than for Android in general, what acquisition methods are feasible and how deep each reaches, and advise honestly what this device will and will not yield before work proceeds; (3) acquire by the deepest lawful method this device permits, with integrity preserved by hashing and documentation; (4) analyse the recovered data by the established disciplines, recognising the common Android core (SQLite storage, the Google account estate, common artefact families and shared apps) beneath any manufacturer modifications; (5) identify at the outset, and use, the Google account, manufacturer cloud, backups and counterparts to supply what the device cannot; and (6) state the device's limits plainly, never reading a shallow extraction's silence as absence, and present findings at honest confidence with the device's feasibility explained."

§ 12 · QUICK CONTROL

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

The Android landscape checklist

- ☐ Device identified precisely: maker, model, version, patch, security state
- ☐ Device preserved from locking, network, wipe, updates and use
- ☐ Feasibility assessed for this specific device
- ☐ Expectations set honestly before work begins
- ☐ Deepest lawful acquisition this device permits chosen
- ☐ Integrity preserved; chain documented
- ☐ Common Android core recognised beneath maker modifications
- ☐ Google account, manufacturer cloud and backups identified early
- ☐ Device's limits stated; silence not read as absence
- ☐ Feasibility reassessed if the device's state changes

Red flags

- "Android" treated as one thing: Example 1.
- Expectations promised before the device is identified.
- An update allowed to close an acquisition route: Example 2.
- A shallow extraction's silence read as absence.
- The Google and manufacturer estate ignored.
- An unfamiliar maker treated as unanalysable: Example 3.
- Preservation neglected.

When to involve a digital forensic expert

Before any expectation is set and before the device's state can change, because on Android feasibility is decided by the specific device and can close overnight: the expert identifies the device precisely, manufacturer, model, version, patch level, security hardware and lock state, preserves it from locking, the network, remote wipe, updates and use, and assesses feasibility for this device rather than for the platform, setting honest expectations of what it will and will not yield (Example 1, §6). The deepest lawful acquisition the device permits is chosen (§4), the recovered data is analysed on the common Android core of SQLite storage, the Google estate and shared artefacts beneath any manufacturer modifications (Example 3, §5), and the Google account, manufacturer cloud, backups and counterparts are identified early as the route around a constrained or sealed device (Example 2, §8). The device's limits are stated plainly and findings reported at honest confidence under CPR Part 35 or CrimPR Part 19. This guide opens the Android block. Through **cflab.uk** and **e-discovery.uk**, Android work navigates the fragmented landscape by assessing each device on its own reality, obtaining the most it permits, and reaching around it through the estate when it permits little.

§ 13 · COMMON QUESTIONS

Frequently asked questions

Why is Android so different from iPhone forensically?

Because it is a landscape, not a platform (§2, §3). An iPhone is one platform from one company; Android is an open system shipped by dozens of manufacturers on thousands of models, each with different hardware, security chips, modifications, versions and update histories. Two Android phones can differ forensically more than an old and a new iPhone. So while the iPhone block could speak of "the iPhone" broadly, Android forensics must speak of this device, this model, this version and this security state, which is the defining fact of the whole block.

What can be recovered from an Android phone?

There is no general answer, only an answer for the specific device (§6, Example 1). Depending on manufacturer, model, version, patch level and lock state, the answer ranges from almost everything, a full file system with deleted data, to almost nothing beyond a shallow logical extraction. So the device is identified precisely and its feasibility assessed individually before any expectation is set, and the client is told honestly what this phone will and will not yield. Where the device yields little, the Google account and estate often supply much of what it cannot.

Does the acquisition method matter more on Android?

Yes, considerably (§4). Because the available methods depend intimately on the device's hardware, software and security state, and their depth varies from a full file system to a limited logical extraction, acquisition is where fragmentation bites hardest. A deep acquisition on one device reaches deleted data a shallow one on another never sees, so the method's depth governs completeness far more than on iOS, and a shallow extraction's silence must never be read as absence. The deepest lawful method the specific device permits is chosen, and its limits stated.

Can an update really change what's possible?

Yes, and it happens (§4, Example 2). Android feasibility is a moving target: a system or security update can close an acquisition route that worked the day before, so a device assessed as amenable may become constrained after updating. This is why the device is preserved from updates as well as from locking and wiping, why feasibility is assessed as of the device's current state and reassessed if that state changes, and why the Google account, manufacturer cloud and backups are identified early as the route around a device whose door has closed.

Is anything the same across all Android phones?

Yes, a common core beneath the variation (§5, Example 3). Android apps store data in SQLite, so deleted-data recovery disciplines apply across makers; a Google account sits behind most devices as a cloud estate like iCloud; common artefact families, calls, messages, contacts, usage, location, recur across models, differing in detail not kind; and the same third-party apps run everywhere. And the method, preservation, integrity, corroboration, attribution, honest confidence, is constant. So once data is obtained, the established disciplines apply, and an unfamiliar maker's device is analysed on familiar foundations.

What if the specific Android device yields very little?

Then the estate is the route (§8). Because an Android device is one window onto a wider estate, a constrained or sealed phone does not end the enquiry: the Google account syncs messages, photos, backups, location history and app data; the manufacturer's cloud holds backups and synced data; local and cloud backups hold earlier states; a paired wearable holds shared data; and counterparts and providers hold their records. Mapped at the outset, these frequently supply much of what a constrained device cannot, which is why the estate is identified early rather than as an afterthought.

§ 14 · REFERENCE

Glossary

Android
Google's open mobile operating system, shipped by many makers.

Fragmentation
The variation across makers, models and versions.

Manufacturer modification
A maker's own layer over Android.

Patch level
The security update state of a device.

Security chip / element
Hardware governing encryption and lock protection.

Feasibility
What acquisition is possible for a specific device.

Logical extraction
A shallower acquisition of accessible data.

Full file system
A deeper acquisition reaching more, including deleted data.

Google account estate
The cloud data behind an Android device.

Manufacturer cloud
A maker's own cloud service (e.g. Samsung Cloud).

Sources and authoritative references

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

- cflab.uk, digital forensics services (Android forensics across manufacturers, per-device feasibility assessment, CPR Part 35 and CrimPR Part 19 reporting): cflab.uk/digital-forensics-services · guides library: cflab.uk/guides
- e-discovery.uk, EDRM Phase 02 · Identification and Phase 04 · Collection (device identification and acquisition as practised): e-discovery.uk/edrm/identification · e-discovery.uk/edrm/collection
- e-discovery.uk, practice method and Knowledge Centre: e-discovery.uk · e-discovery.uk/knowledge
- Android Open Source Project, security documentation (encryption, verified boot, security updates): source.android.com, security · Android security bulletins (patch levels): source.android.com, bulletins
- Forensic Science Regulator, Code of Practice v2 (validated methods and honest reporting) and ISO/IEC 27037: gov.uk, FSR Code · iso.org, 27037 · NPCC digital evidence guidance: npcc.police.uk
- 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 Android forensics as at September 2026. Android is a fragmented and rapidly evolving landscape: what can be recovered depends on the specific manufacturer, model, version, patch level, security hardware and lock state, and can change with a single update, so no general statement about "Android" should be relied on for a particular device, and feasibility must be assessed for each. 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

Android work at the laboratory navigates the fragmented landscape by assessing each device on its own reality. Every Android device is identified precisely at the outset, manufacturer, model, version, patch level, security hardware and lock state, and preserved from locking, the network, remote wipe, updates and use, because on Android feasibility is decided by the specific device and can close overnight (Example 2). Feasibility is assessed for this device rather than for the platform, and the client is told honestly what it will and will not yield before work proceeds (Example 1). The deepest lawful acquisition the device permits is chosen, with integrity preserved by hashing and documentation, and the recovered data is analysed on the common Android core, SQLite storage, the Google account estate, common artefact families and shared third-party apps, that lies beneath any manufacturer's modifications, so that an unfamiliar maker's device is analysed on familiar foundations (Example 3, guides 155, 151, 156). The Google account, manufacturer cloud, backups, paired devices and counterparts are mapped early as the route around a constrained or sealed device, the device's limits are stated plainly so that a shallow extraction's silence is never read as absence (guide 146), and findings are reported at honest confidence under CPR Part 35 or CrimPR Part 19. This guide opens the Android block. Conflicts checks, confidential scoping discussions and fixed-fee estimates are routine; and the guiding principle of the block is set here: there is no "Android phone", only this device, assessed honestly, acquired as deeply as it permits, and reached around through its estate when it permits little.

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