Considus

TECHNICAL WHITE PAPER

Catchlight for iPhone

How to tell whether a notes app is private

Seven tests you can run on any notes app in an afternoon, what each one really probes, and Catchlight scored on the same scale.

Applies to Catchlight 1.0

Introduction

Every few weeks somebody asks a version of the same question in one of the privacy forums. Is this notes app actually private? And the thread does the same thing every time. One person has read the privacy policy, another has read the marketing page, a third has strong opinions about the founder, and forty comments later nobody has opened the app and tried anything.

Which is odd, because you wouldn't buy a car that way.

You'd not take the advert's word for it. You'd walk round it, open the boot, look at the tyres, and then you'd insist on driving it, because everybody understands that a car you haven't driven is a car you know nothing about. Somehow we install an app, hand it years of private writing, and the closest we get to a test drive is looking at the screenshots.

Here's the walk-round, then. Seven things you can do to any notes app, on the phone in your hand, none of which need you to understand a word of cryptography. Each one gives you a straight answer, and each one is hard for a company to fake, because they're all tests of what the software does rather than what the company says.

They don't tell you everything either. What none of this tells you is a section of its own further down, and I'd rather you read that than come away thinking a clean sweep means safe.

Why test at all, when there's a policy to read

Three places claim to tell you how an app treats your data. The marketing page, the privacy policy, and the software.

The marketing page picks its facts, which is what marketing is for and I'm not going to be sniffy about it. The privacy policy is a legal document written by people whose job is to protect the company, and it describes the widest thing the company is allowing itself to do rather than the thing it's actually doing this morning. Both of those can be rewritten next Tuesday and you'd never know.

The software is the only one of the three that can't be edited after you've looked at it.

Why these seven, and not some others

I didn't pick them for variety. Each one had to clear four bars, and a lot of perfectly sensible checks failed one of them.

It has to be runnable by anyone. No tools, no jailbreak, no reading of source code, nothing you'd need a background for. If a test needs a proxy and a rooted device, it belongs in a professional audit and not on this page.

It has to test behaviour rather than a document. Anything a company can rewrite on a Tuesday afternoon isn't evidence, which is what rules out the privacy policy and most of the marketing.

It has to be hard to fake. This is the one that does most of the work, and it's why App Store privacy labels aren't on the list. They're useful, they're worth reading, and they're self-declared by the developer and not audited by anyone. A label is a claim wearing the uniform of a check. The seven below are all properties of how the software was built, which is a much harder thing to dress up.

Two good checks fell at those hurdles and are worth naming, because you should still do them. Whether the app is open source matters a great deal, and most people can't read the code, so it doesn't work as a test you personally run. And where the company is registered changes what pressure it can be put under, which is real and important and simply isn't something you can determine by tapping about in an app.

Where to start

Run the first one tonight, on whatever you're using now. Two minutes, and it needs nothing but the app you already have.

The other six are the sort of thing you do while using an app normally rather than sitting down to a test session. There's an order worth taking them in, and it's at the end, where the numbers will mean something to you.

Test 1, try to reset your password

Open the app, start the forgotten-password process, and stop before you finish it. Just look at what you're offered.

If it can email you a link, let you set a new password, and your old notes are all sitting there afterwards exactly as they were, then somebody other than you can read those notes. A company can only hand back what it's already able to open. If instead it warns you that recovering the account means losing the writing, or it asks for a phrase you were told to write down on day one, then the key is yours.

I've written about why that works in more detail in who holds the key, so I won't do the whole thing again here. What matters for a test list is that it takes two minutes, it needs no expertise, and the answer isn't something a company can dress up.

Do this one first. It settles more than the other six put together.

Test 2, count the screens before you can write

Delete the app, install it again, and count how many screens stand between you and typing one sentence.

An app that wants an email address before you've written anything has made a record connecting you to whatever comes next. Everything after that point can be encrypted beautifully and the connection still exists, sitting in a database, waiting for the day somebody has a reason to ask for it.

An app that opens on a blank page knows nothing about you, and it can't be made to know it later.

One honest caveat, because it applies to me too. A paid app on the App Store always tells Apple you bought it. No iOS app gets round that, and any iOS app claiming otherwise is either confused or lying.

Test 3, turn off the network and carry on

Airplane mode, then use the thing properly for a day. Write, search, edit, delete.

An app that stops working without a signal was sending your writing off somewhere to do perfectly ordinary jobs. An app that carries on was doing those jobs on your phone all along.

Watch the search particularly, because that's where this usually breaks and it's the failure people don't notice. If everything else works and search goes dead, the search index lives on a server, and a search index is made of your words.

Test 4, go and look at the files

Find out where the app actually keeps your data, open that place in the Files app or on a computer, and open one of the files.

You'll get one of three answers. Readable text, which tells you the encryption isn't doing the thing you hoped it was. Unreadable rubbish, which is the answer you want. Or you can't find your data anywhere at all, which means it lives on their servers and the other six tests matter a great deal more than they did a minute ago.

This is the engine bay, and it's worth saying that a clean one isn't automatically good news. The cleanest engine bay I ever looked at had been jet-washed the morning I arrived, which is a fine way to make a leak invisible for about a fortnight.

Test 5, make two devices disagree

This one needs two devices. Turn the network off on both, edit the same note differently on each, then reconnect them.

Three things can happen and they're not equally good. The app can show you both versions and ask which you want, which is correct and is also the rarest. It can keep both as separate copies, which is messy and safe. Or it can quietly keep one and bin the other, and you'll find out months later when you go looking for a paragraph that isn't there.

You might reasonably ask what any of that has to do with privacy. It's this. A company that thought hard about your data doesn't silently delete half of it, and the care you can measure is a decent predictor of the care you can't. Nobody builds a careful conflict resolver and a careless key store.

Test 6, get everything out

Export the lot, open the export in something else entirely, and then find out what happens to it if you stop paying.

An export only the original app can read isn't an export, it's a souvenir. And an export that stops working the day your subscription lapses isn't portability, it's a hostage arrangement with a nicer name.

Two things to look for. It has to come out in something other software reads, which in practice means plain text or Markdown. And it has to keep working when you're no longer a customer, because that's the exact moment you'll want it.

Test 7, imagine the company is gone

No button to press for this one. Just ask whether your data still works if the company stops trading tomorrow.

You already know the answer, because it falls out of the first six. Data in a folder you control, encrypted under a key you hold, in a format documented well enough that somebody could write a reader for it. All three and you're fine. Miss any one of them and you aren't, and you'll discover which one at the worst possible moment, which is generally how this goes.

What none of this tells you

A clean sweep of all seven doesn't mean an app is safe. It means an ordinary person couldn't find anything wrong with it in a day, which is genuinely worth knowing and is not the same claim.

These tests can't tell you whether the cryptography is implemented properly. A good design with one bad line in it looks identical from outside, and only somebody reading the code finds that.

They can't find a deliberate backdoor. Nothing you can do from the outside will, which is why source availability and a real independent audit matter, and why both are a lot rarer than the marketing round here implies.

They don't measure metadata. File sizes, file counts, timestamps, all of that usually stays visible even when your content doesn't, and for some people that's the part that matters.

They tell you nothing about where the company is or who paid for it. Jurisdiction and ownership decide what pressure that company can be put under, and you can't test either from your sofa.

And they only describe the software you have today. An update changes the software. An acquisition changes the incentives, and the acquisition is the one nobody sees coming.

What the tests are really probing

Each of these has a formal name, and knowing it is useful if you ever want to take an argument further than an app store review.

Test 1 is key custody, which is the property underneath every "end-to-end encrypted" and "zero-knowledge" claim ever made. It is the single question that separates apps whose marketing sounds identical.

Test 2 is data minimisation. UK and EU data protection law requires personal data to be adequate, relevant and limited to what is necessary, at Article 5(1)(c) of the GDPR. An app that demands an email address to let you write a note is worth asking about. In privacy-engineering terms it is also about identifiability and linkability, which is to say whether a record about you can be tied to you at all.

Test 3 is about where processing happens, and a search index sitting on a server is processing your content whatever the encryption story says elsewhere.

Test 4 is encryption at rest. OWASP's Mobile Application Security Verification Standard covers this under MASVS-STORAGE, which also asks whether sensitive data leaks into logs, caches and device backups, and those are worth a thought even when the main store is sealed.

Test 5 is data integrity, and strictly it isn't a privacy property at all. It's in the list as a proxy for care, on the argument I made when describing it.

Test 6 is portability, and it has actual legal weight. Article 20 of the UK and EU GDPR gives you a right to receive your personal data in a structured, commonly used, machine-readable format. An app that can only export into itself is not honouring the spirit of that, whatever its lawyers say about the letter.

Test 7 has no formal name that I know of. It's continuity, and it's the one that turns out to matter most when it goes wrong.

What the professionals do instead. A real assessment uses the MASVS categories above with the Mobile Application Security Testing Guide, which supplies actual test procedures and needs tooling and access this protocol deliberately avoids. LINDDUN is the equivalent for privacy threats specifically, seven categories worked over a data-flow diagram. These seven tests are a lay approximation of a small corner of that work. They are not a substitute for it and I'd rather say so than let the list look more authoritative than it is.

Catchlight, on the same seven

I'd have no business publishing this if I ducked it. Catchlight hasn't launched, so this scores the app as built and merged rather than as sold.

TEST	RESULT	WHY
1, reset	Pass	No account and no password, so there's nothing to reset and nobody to email. Recovery is a 12-word phrase you hold
2, sign-up	Pass	No email address, no account, straight to a blank page. Apple knows you bought it, as with every paid iOS app
3, airplane mode	Pass	All of it works with no network, search included. Sync is a folder operation, not a dependency
4, files	Pass	Your own iCloud Drive or Dropbox folder, files are ciphertext, and the index is encrypted too
5, conflict	You can't run it yet	v1.0 syncs one device. The resolver shows both versions and never merges silently, and you can't exercise any of that until pairing ships in v1.1
6, exit	Pass	Markdown export of everything, losslessly, and it keeps working if your subscription lapses
7, disappearance	Pass	Your folder, your key, a format documented in full in the architecture paper, and the source is open

Now the parts that don't look as good on a table.

v1.0 is one device. Test 5 can't be run on it, and the sync story tells a bigger tale than the launch build delivers. That gap is real and it's the thing I'd pick at first if I were you.

It isn't audited. A cryptography specialist went through the design with me in June 2026, verbally, before the implementation was finished. That's a design review. It is not an audit and I'm not going to let the word drift.

iOS only. If your writing needs to be on Android or a laptop, this whole protocol has just scored an app you can't use.

Import costs money, export doesn't. Export is free and stays free because getting your writing out should never have a price on it. Import sits behind the subscription because import creates data. I think that's the right way round and it's still true that one direction costs and the other doesn't.

The folder shows its shape. The index is encrypted, so nobody can count your notes or see what you deleted. What they can still see is how many files there are, how big they are, and when the provider last touched them. Nobody reads a word. Somebody can tell you were writing on Tuesday night.

The order I'd do them in

Test 1 tonight, as I said. If the app survives that, give it tests 2, 3 and 4 over a normal day of use, because those three are just using the app while paying attention.

Do 5 and 6 before you move years of writing in, rather than after, which is when most people think of it. And test 7 needs nothing from you at all, which is convenient, because it's the one you'll care about most on the day it stops being hypothetical.

Nobody let you drive the app before you filled it with your life. That's what the seven are for.

References

Standards and guidance

OWASP Mobile Application Security Verification Standard v2.1.0

OWASP Mobile Application Security Testing Guide

LINDDUN privacy threat modelling, KU Leuven

UK General Data Protection Regulation, Article 5 on data minimisation and Article 20 on portability

Further reading

EFF Surveillance Self-Defense

Apple Platform Security

Companion documents

The Catchlight encryption architecture, for what happens when I put my own app through all of this at length

Who might read your notes, for the adversary-by-adversary version of the same questions

What AI features cost you in a notes app, for test 3 applied to the AI case, where airplane mode is the whole answer

Version history

VERSION	DATE	CHANGE
1.2	1 September 2026	The Companion documents block named two papers in plain text. It now lists three and each is a link

VERSION	DATE	CHANGE
1.1	1 September 2026	Test 7's middle condition asked for "a format other software reads". Nothing reads a sealed blob today. It now asks for a format documented well enough that a reader could be written, which is the property that actually survives a company, and Catchlight's own row is corrected the same way
1.0	31 August 2026	First release