About the Nano's security

Altimeter Cloud

A flight log is only worth something if you can trust it hasn't been tampered with, which matters all the more for records and competitions. The Nano is built so the data it produces can be proven genuine, and so the way it works can't be quietly altered.

Two pieces of well established cryptography do the heavy lifting: AES-256 encryption protects the firmware, and an Ed25519 digital signature protects every flight log.

GOOD TO KNOW

You don't need to do anything to enable any of this. Every log is signed automatically as it is written, and checked for you when you upload it to the Altimeter Cloud. An edited log will always fail to verify, which is what keeps the results board honest.

Signed flight logs (Ed25519)

Every flight log the Nano saves is sealed with an Ed25519 digital signature, a modern scheme used to secure web traffic, secure shell (SSH) access and messaging apps such as Signal.

ON THE NANO

1. Fingerprint

SHA-256 reduces the whole log to one 256-bit value

ON THE NANO

2. Sign

The fingerprint is signed with the private key, sealed in the encrypted firmware

ON UPLOAD

3. Check

The Altimeter Cloud recomputes the fingerprint and checks it with the public key

RESULT

4. Verified

A match means genuine and unedited. Any change at all and the log is rejected

A fingerprint. SHA-256 reads the entire log and reduces it to a single 256-bit value. Change even one character anywhere in the file and the fingerprint comes out completely different.

A signature. The Nano signs that fingerprint with a private key held inside its encrypted firmware. A valid signature can only be made with the private key, but anyone holding the matching public key can check it. The two keys are mathematically linked, yet the private key can't be worked out from the public one.

In practice that means:

- If a log is edited in any way after the Nano writes it, the fingerprint changes, the signature no longer matches, and the log is rejected.
- Only a genuine Nano can make a valid signature, so an altered or invented log can't be signed to pass as real.

- Our Altimeter Cloud servers hold only the public key, never the private one. Even if our servers were ever compromised, there would be nothing there that could be used to forge a log.
- Because checking a log only needs the public key, it can be verified independently: a competition official can confirm it is genuine without relying on us.

What the signature covers

DEVICE AND COMPETITOR TAGS ARE SIGNED TOO

Your device tag and all three competitor tags are written into the log header and are part of the signature. Changing any of them in a saved log, even by one character, makes it fail verification. To fly under a different name or number, change the tags in the settings before the flight.

The signature covers the whole file, including the header with your device serial number, the settings flown, your device tag and all three competitor tags. None of those can be swapped or changed after the flight without the log failing verification.

- **ACLZ logs** carry two signatures: one over the complete log before compression, and one over the CSV it expands to. So a CSV the Altimeter Cloud rebuilds from an ACLZ log can still be verified.
- **CSV logs** are signed over the whole file.
- **Flight info pages** are signed the same way as the log beside them.
- **Recovered logs**, rebuilt after a power loss, are signed like any other, so they can still be proved genuine. See log recovery.

Opening a CSV in a spreadsheet and saving it is enough to break the signature, so always upload the original file from the Nano. See flight log verification for how to check a log.

Encrypted firmware (AES-256)

The Nano's firmware is stored on the chip encrypted with AES-256, the same standard governments and banks use to protect sensitive information. The Nano will only run properly encrypted firmware, so a modified or unofficial program can't be loaded onto it, and the firmware can't be read back off the chip in any usable form.

That protects our work from being copied, but more importantly for you it protects the private signing key. The key is sealed inside the encrypted firmware, never appears on the USB drive or in a log, and can't be lifted off the device to forge logs. The Nano has no wireless connection either: WiFi is permanently disabled, and firmware can only be installed over USB with our update tool.

AES-256 can't be brute forced. A 256-bit key has roughly 1.2 followed by 77 zeros possible values. Even if every computer on the planet worked together testing billions of keys a second, it would take vastly longer than the age of the universe to get through a meaningful fraction of them, and there is no known shortcut.

Why the two together make it trustworthy

The strength of the log signature rests on the private key staying on the device, and that is exactly what the firmware encryption guarantees. Because the Nano refuses to run unencrypted firmware, there is no way to load a modified program that might leak the key or sign false data. The encryption keeps the private key out of reach, and the signature uses it to vouch for every log, while the public half, which is all anyone needs to verify a log, can be shared freely.

Both AES-256 and Ed25519 are open, published standards that have been studied intensely and are trusted to secure online banking, web traffic and secure shell access. Paired with a private key that can't be reached, that makes a Nano flight log something you can stand behind, whether you are chasing a personal best or submitting an official competition result.

Logs from firmware before 1.52

Earlier firmware signed logs with HMAC SHA-256 rather than Ed25519, mixing a secret key shared between the Nano and our servers into the same whole-file fingerprint. It is tamper evident in exactly the same way, and the Altimeter Cloud still verifies these logs automatically, so nothing you have already recorded is affected.

The move to Ed25519 in 1.52 kept that and added two things: the signing key on the device became private, with our servers holding only the public half, and a log can be verified independently by anyone with the public key.