About the Nano's security

A flight log is only worth something if you can trust it has not 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 cannot 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.

Encrypted firmware (AES-256)

The program that runs the Nano, its 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 it is not possible to load a modified or unofficial program onto it, nor to read the firmware back off the chip in any usable form.

This matters for two reasons. It protects our work from being copied, but more importantly for you it protects the private signing key that vouches for your flight logs, which is described below. That key lives inside the encrypted firmware, so it cannot be lifted off the device and used to forge logs.

AES-256 is not something that can be brute forced. A 256 bit key has roughly 1.2 followed by 77 zeros possible values, and trying them all is so far beyond any current or foreseeable computing power that it is treated as impossible. Even if every computer on the planet worked together testing billions of keys every second, it would still take vastly longer than the age of the universe to get through a meaningful fraction of them, and there is no known shortcut that changes this.

Tamper evident, independently verifiable logs (Ed25519)

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

First, 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 that fingerprint comes out completely different. Anyone can compute this fingerprint; on its own it simply summarises the data.

Second, a signature. The Nano signs that fingerprint with a private key held inside its encrypted firmware. A valid signature can only be created with the private key, but it can be checked by anyone holding the matching public key. The two keys are mathematically linked, yet the private key cannot be worked out from the public one. This is what turns the fingerprint into a true signature rather than a simple checksum.

In practice that means:

- If a log is edited in any way after the Nano writes it, its fingerprint changes, the signature no longer matches, and the log is rejected.
- Only a genuine Nano can produce a valid signature, because only it holds the private key, so an altered or invented log cannot 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 verifying a log needs only the public key, a log can be checked independently — a competition official can confirm it is genuine without relying on us.

The signed data is the whole file, including your device serial number and your competitor and device tags, so none of those can be swapped or altered after the flight either. When you upload a log to the Altimeter Cloud, our server recomputes the fingerprint and checks the signature against the public key. A valid signature confirms the log is genuine and unedited; anything else is rejected.

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. The key is sealed inside the encrypted firmware. It never appears on the USB drive or in the log, and because the Nano refuses to run unencrypted firmware there is no way to load a modified program that might leak it or sign false data. The two protections reinforce one another: the encryption keeps the private key out of reach, and the signature uses that key to vouch for every log — while the public half, which is all anyone needs in order to verify, can be shared freely.

Both AES-256 and Ed25519 are open, published standards that have been studied intensely and are trusted to secure things like online banking, secure web traffic and secure shell access. There is no known practical way to break either of them. Paired with a private key that cannot be reached, that makes a Nano flight log something you can genuinely stand behind, whether you are chasing a personal best or submitting an official competition result.

> **Logs from firmware before version 1.52**
>
> Earlier firmware signs logs with HMAC SHA-256 rather than Ed25519. This mixes 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 — any edit breaks it — and the Altimeter Cloud still verifies these logs automatically, so nothing you have already recorded is affected. The move to Ed25519 in version 1.52 keeps that tamper evidence and adds two things: the signing key on the device is now private, with our servers holding only the public half so a server compromise cannot forge logs, and a log can be verified independently by anyone holding the public key.

> *i* **GOOD TO KNOW**
>
> You do not need to do anything to enable any of this. Every log is signed automatically as it is written, and verification happens for you when you upload to the Altimeter Cloud. It simply means an edited log will always fail to verify, which is what keeps the results board honest.