Compression & Encryption

Every production Jupiter ships with its firmware AES-256 encrypted in flash and locked in production mode, not development mode. The firmware cannot be read out of the device, and the device cannot be re-flashed or reconfigured into a different setup with the exception of firmware updates encrypted by us. All key material lives inside that encrypted flash, so nothing secret is recoverable from the hardware. AES-256 is the standard trusted for the most sensitive government data worldwide; breaking it by brute force is considered impossible with any current or foreseeable technology.

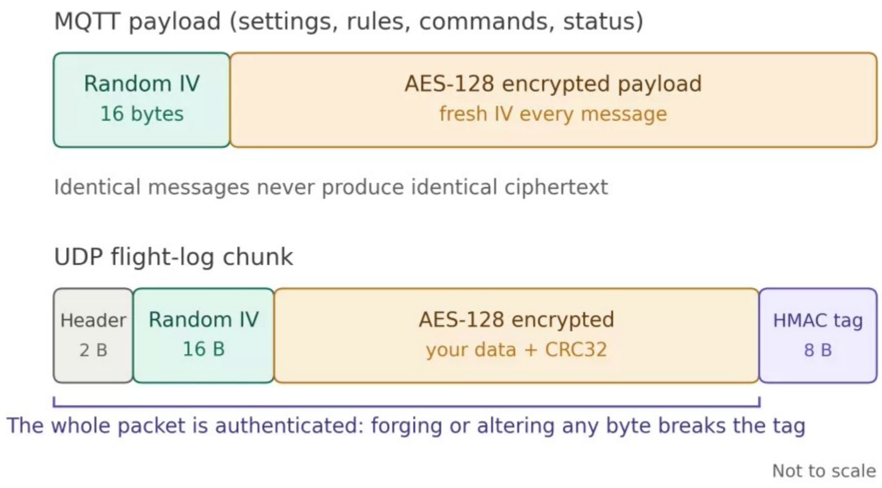
The flight log

Flights are recorded across multiple data tiers at different rates, stored as differences between consecutive samples, compressed with LZMA when recording ends, and cryptographically signed with Ed25519, a signature that is considered impossible to forge with current technology, so any alteration to a saved flight is detectable by the server. The result is the .aclz flight file. The full format, tier structure and signature scheme have their own page: ACL flight log format.

Compression on the device

Jupiter uses three compression techniques, each where it earns its keep. Delta encoding stores each sample as the change from the previous one, stripping the natural redundancy out of sensor streams before any compressor sees them. Key frames keep the data in check at periodic intervals. LZMA, the algorithm family behind 7-Zip, compresses the flight log itself; the log_zlevel setting picks the effort, from level 4, finishing a typical flight in under a second, through the default 5, to level 9, which squeezes out the last fraction at several times the cost. Since compression runs after landing and before upload, the trade is simply upload size against how quickly the board is back on the air. Deflate (zlib) handles console and diagnostic logs, and anything under a couple of kilobytes is sent uncompressed, because below that the overhead exceeds the saving.

Communication security



MQTT transport

Settings, rules, commands and status travel over MQTT, and every payload in both directions is AES-128 encrypted at the application layer with a fresh random initialisation vector per message, so identical messages never produce identical ciphertext. AES-128 has no known practical attack and is considered unbreakable by brute force with current technology. The device rejects any inbound command that doesn't decrypt cleanly.

UDP transport

Live tracking and flight-log upload use compact UDP datagrams built for weak signal. The tracking beacon is a 50-byte packet sealed with an HMAC-SHA256 authentication tag from the per-device secret, so positions can't be forged or altered in transit.

An HMAC-SHA256 tag cannot be produced without your device's secret, and no practical attack against the scheme exists. Flight-log chunks carry full protection: each datagram is AES-128 encrypted with a fresh IV, authenticated with the same per-device HMAC scheme, and carries a CRC32 of the original data, so every chunk the server accepts is confidential, provably from your device, and verified bit-for-bit. Lost chunks are re-requested; corrupted or forged ones can never enter your flight.

LoRa

When using our LoRa expansion module and because LoRa frames carry position data over an open radio band, they'll receive the same unbreakable-grade protection before release: encrypted and authenticated per packet, so a Jupiter's position is only ever readable and trustable by its owner.