MAVLink Message Signing

MAVLink 2 message signing allows PX4 to cryptographically verify that incoming MAVLink messages originate from a trusted source (authentication).

> **INFO**
>
> This mechanism does not *encrypt* the message payload.

Overview

When signing is enabled, PX4 appends a 13-byte signature to every outgoing MAVLink 2 message.

Incoming messages are checked against the shared secret key, and unsigned or incorrectly signed messages are rejected (with exceptions for safety-critical messages).

The signing implementation is built into the MAVLink module and is always available — no special build flags are required. It is enabled and disabled at runtime through the MAV_SIGN_CFG parameter.

Enable/Disable Signing

The MAV_SIGN_CFG parameter controls whether signing is active:

Value	Mode	Description
0	Disabled (default)	No signing. All messages are accepted regardless of signature.
1	Non-USB	Signing is enabled on all links **except** USB serial connections. USB links accept unsigned messages.
2	Always	Signing is enforced on all links, including USB.

WARNING

Setting `MAV_SIGN_CFG` alone does not enable signing — a secret key must also be present (see Key Provisioning below). If no key has been set (or the key is all zeros with a zero timestamp), all messages are accepted regardless of this parameter.

To **disable** signing, set `MAV_SIGN_CFG` to zero.

Key Provisioning

The signing key is set by sending the MAVLink SETUP_SIGNING message (ID 256) to PX4. This message contains:

- A 32-byte secret key
- A 64-bit initial timestamp

WARNING

For security, PX4 only accepts `SETUP_SIGNING` messages received on a **USB** connection. The message is silently ignored on all other link types (telemetry radios, network, and so on). This ensures that an attacker cannot remotely change the signing key.

Key Storage

The secret key and timestamp are stored on the SD card at:

```txt
/mavlink/mavlink-signing-key.bin
```

The file is a 40-byte binary file:

Offset	Size	Content
0	32 bytes	Secret key
32	8 bytes	Timestamp (`uint64_t` , little-endian)

The file is created with mode `0600` (owner read/write only), and the containing `/mavlink/` directory is created with mode `0700` (owner only).

On startup, PX4 reads the key from this file. If the file exists and contains a non-zero key or timestamp, signing is initialized automatically.

INFO

The timestamp in the file is set when `SETUP_SIGNING` is received. A graceful shutdown also writes the current timestamp back, but in practice most vehicles are powered off by pulling the battery, so the on-disk timestamp will typically remain at the value from the last key provisioning.

INFO

Storage of the key on the SD card means that signing can be disabled by removing the card. Note that this requires physical access to the vehicle, and therefore provides the same level of security as allowing signing to be modified via the USB channel.

How It Works

Initialization

1. The MAVLink module calls `MavlinkSignControl::start()` during startup.
2. The `/mavlink/` directory is created if it doesn't exist.

3. The `mavlink-signing-key.bin` file is opened (or created empty).
4. If a valid key is found (non-zero key or timestamp), signing is marked as initialized.
5. The `accept_unsigned` callback is registered with the MAVLink library.

Outgoing Messages

When signing is initialized, the `MAVLINK_SIGNING_FLAG_SIGN_OUTGOING` flag is set, which causes the MAVLink library to automatically append a SHA-256 based signature to every outgoing MAVLink 2 message.

Incoming Messages

For each incoming message, the MAVLink library checks whether a valid signature is present. If the message is unsigned or has an invalid signature, the library calls the `accept_unsigned` callback, which decides whether to accept or reject the message based on:

1. **Signing not initialized** — If no key has been loaded, all messages are accepted.
2. **Allowlisted message** — Certain safety-critical messages are always accepted.
3. **Sign mode** — The `MAV_SIGN_CFG` parameter determines behavior:
   - Mode 0 (disabled): All unsigned messages are accepted.
   - Mode 1 (non-USB): Unsigned messages are accepted only on USB links.
   - Mode 2 (always): Unsigned messages are rejected on all links.

Unsigned Message Allowlist

The following messages are **always** accepted unsigned, regardless of the signing mode. These are safety-critical messages that may originate from systems that don't support signing:

Message	ID	Reason
RADIO_STATUS	109	Radio link status from SiK radios and other radio modems

Message	ID	Reason
ADSB_VEHICLE	246	ADS-B traffic information for collision avoidance
COLLISION	247	Collision threat warnings

Security Considerations

- **Physical access required for key setup**: The `SETUP_SIGNING` message is only accepted over USB, so an attacker must have physical access to the vehicle to provision or change the key.
- **Key not exposed via parameters**: The secret key is stored in a separate file on the SD card, not as a MAVLink parameter, so it cannot be read back through the parameter protocol.
- **SD card access**: Anyone with physical access to the SD card can read or modify the `mavlink-signing-key.bin` file, or just remove the card. Ensure physical security of the vehicle if signing is used as a security control.
- **Replay protection**: The MAVLink signing protocol includes a timestamp that prevents replay attacks. The on-disk timestamp is updated when a new key is provisioned via `SETUP_SIGNING`. A graceful shutdown also persists the current timestamp, but since most vehicles are powered off by pulling the battery, the timestamp will typically reset to the value from the last key provisioning on reboot.
- **No encryption**: Message signing provides authentication and integrity, but messages are still sent in plaintext. An eavesdropper can read message contents but cannot forge or modify them without the key.

Edit on GitHub

Previous page
Custom MAVLink Messages

Next page
Security Hardening