Channels & Encryption

- How Meshtastic Channels Work
- Creating Private Channels

How Meshtastic Channels Work

Meshtastic uses a channel system for message segmentation and encryption. Each node can have up to 8 channels simultaneously, each with its own name and (optionally) its own encryption key. Channel encryption uses AES-256-CTR keyed by the channel PSK; a channel can also have no key, in which case its traffic is unencrypted. See the PSK reference for key details.

Channel Structure

- **Up to 8 channels** per node, indexed 0 through 7
- **Channel 0** is special - it is the primary channel (only one channel can be primary; channels 1-7 are secondary). Position updates and telemetry are broadcast on channel 0 by default.
- Each channel has:
  - A **name** (displayed in the app)
  - A **pre-shared key (PSK)** - the encryption key for that channel (a channel may also have no key, making it unencrypted)
  - Optional **uplink/downlink MQTT settings** for internet bridging

The Default Public Channel

Out of the box, Meshtastic nodes are configured with:

- **Channel name:** LongFast (note: "LongFast" is really the name of the default modem *preset*; the firmware uses it as the default channel name too, but the preset and the channel name are distinct concepts)
- **PSK:** `AQ==` - this is the single byte `0x01` (base64 `AQ==`), which is firmware shorthand meaning "use the built-in default key." It is not itself the full key; it is an index that selects the publicly-known default key. Because that key is public, the default channel offers no privacy.

Any node using the default LongFast channel can communicate with any other node using the same channel - the encryption provides no privacy since the key is public. This is intentional: it

allows strangers to discover and communicate across the mesh.

Channel URL Scheme

Channels are shareable via URL or QR code. Example channel URL:

```
https://meshtastic.org/e/#CgUYAyIBAQ==
```

The hash after `#` is a base64-encoded channel configuration. To encode or decode channel configurations, use the tool at https://meshtastic.org/e/.

Sharing a channel URL (or its QR code) is the standard way to invite someone to a private channel - they scan or paste the URL and their node is automatically configured with the correct name and PSK. **Important:** the channel URL/QR contains the PSK in cleartext (base64) - the URL effectively *is* the key. Anyone who sees it can join (and decrypt) the channel. Share a private channel's URL only over a secure out-of-band path (in person, or an encrypted messenger such as Signal); never post a private channel URL in chat, email, a forum, or anywhere public.

MQTT Uplink and Downlink

Each channel can be individually configured to bridge traffic to/from an MQTT broker:

```
meshtastic --ch-index 0 --ch-set uplink_enabled true
meshtastic --ch-index 0 --ch-set downlink_enabled true
```

When uplink is enabled on a channel, packets on that channel are published to the MQTT broker. Downlink delivers MQTT messages back to the mesh. This is the basis of Meshtastic internet bridging and long-distance message delivery via the MQTT network. (Caution: enabling downlink on the default public channel lets anyone on the public broker inject messages into your local RF mesh - only enable downlink on channels you control and intend to bridge.)

Admin Channel

A designated private admin channel allows remote configuration of any node that shares the admin channel's PSK. **Note:** this shared-PSK admin channel is a *legacy* method; firmware 2.5+ uses public-key (PKC) remote administration - admin keys based on each node's X25519 public key - instead of a shared admin-channel PSK, and that is the preferred approach on modern firmware.

- Create a channel with a unique name (e.g., `admin`) and a strong, randomly generated PSK
- Configure it as the admin channel on all nodes you want to remotely manage
- From any node with the admin channel, you can send configuration commands to remote nodes over the mesh - no physical access required

This is essential for maintaining remote or hard-to-reach infrastructure nodes.

Creating Private Channels

To communicate privately with a group, create a channel with a unique PSK known only to group members. Anyone without the PSK cannot decrypt messages on that channel - **unless** a gateway on that channel uplinks to MQTT without `encryption_enabled`, which republishes the traffic in cleartext to the broker. Channels are encrypted with AES-256-CTR when you use a full 32-byte (256-bit) PSK (a 16-byte key gives AES-128).

Via the App

1. Open the Meshtastic app and go to **Radio Config → Channels**
2. Select an unused channel slot (index 1 - 7; leave index 0 as the public primary unless you have a specific reason to change it)
3. Set a **channel name** (e.g., `TeamAlpha`)
4. Tap **Generate** to create a random PSK, or enter a known PSK manually
5. Save the channel
6. Share the channel URL or QR code with group members out-of-band (signal, in person, etc.)

Via the CLI

Add a new channel (this creates an empty channel at the next free index; do not pass the name to `--ch-add`):

```
meshtastic --ch-add
```

Name the channel and set its PSK on that index. Use `random` to have the firmware generate a strong key, or supply your own base64 key string directly (there is no `base64:` prefix):

```
meshtastic --ch-index 1 --ch-set name TeamAlpha
meshtastic --ch-index 1 --ch-set psk random
```

Export the channel URL for sharing:

```
meshtastic --export-config
```

The config export includes channel URLs that can be shared with other users.

Security Considerations

- **PSK distribution security:** The security of a private channel is entirely dependent on how the PSK is distributed. Share it via an end-to-end encrypted channel (Signal, in person) - not via SMS or unencrypted email.
- **The default LongFast channel is not private.** All Meshtastic users can read it. Never send sensitive information on LongFast.
- **MQTT uplink can leak even a private channel.** If any gateway node on your private channel uplinks to an MQTT broker without `mqtt.encryption_enabled` set, your channel's traffic is republished to the broker in cleartext - so "no PSK = can't read it" only holds for the RF mesh, not for an MQTT-connected mesh.
- **Channel names are not secret.** Only the PSK encrypts message content. The channel name may be visible to other nodes in some circumstances.
- **Changing the PSK:** If a group member's device is lost or compromised, generate a new PSK and redistribute it to all remaining members. The compromised device will no longer be able to decrypt messages after the PSK change. Note there is no per-user revocation and no forward secrecy - rotating the key protects future traffic, but anyone who captured past ciphertext can still decrypt it with the old key.

Position and Telemetry Privacy

By default, position and telemetry are broadcast on channel 0 (the public primary channel). If you want location data to remain within your private group:

1. The simplest, lowest-risk option is to disable position broadcasting entirely: **Radio Config → Position → Position Broadcast Interval → 0**. This keeps you connected to the public mesh while withholding your location.
2. Alternatively, you can make your private channel the primary (index 0). **Be aware of the tradeoff:** putting a private channel at index 0 replaces the default public LongFast primary, which cuts your node off from the public mesh - you will no longer see or be reachable on the public network. Only do this if isolation from the public mesh is intended.