Meshtastic Channel Encryption

How Meshtastic Encryption Works

Each Meshtastic channel is encrypted with **AES-CTR** (counter mode), keyed by the channel pre-shared key (PSK). The key length depends on the PSK: the default public key (`AQ==`) yields AES-128, while a generated 32-byte PSK yields AES-256. Any node that has both the correct channel name and the correct key can decrypt messages on that channel. This is **shared-key** encryption, not end-to-end: every member holding the PSK can read all traffic on the channel, while relay nodes that do not have the key forward ciphertext without decrypting it. There is no per-recipient secrecy and no forward secrecy, and an MQTT gateway can decrypt the traffic before uplink. They do not need the channel key to relay packets.

The Default Public Key

Meshtastic's built-in "Default" channel uses a well-known, publicly documented key. The PSK is represented in base64 as `AQ==`, which decodes to a single byte `0x01` - a shorthand "simple" key that selects the firmware's built-in default key (effectively AES-128). This is **intentionally not secret** - the Default channel is the public mesh. Anyone running the Meshtastic app can read messages on the Default channel. Do not send sensitive information on the Default channel.

Creating a Private Channel

To communicate privately:

1. Set a custom channel name (different from "LongFast" or "Default").
2. Use the app's **Generate** button to create a cryptographically random 256-bit PSK. Do not manually type a key - human-chosen keys have low entropy.
3. Share the channel configuration with intended participants using the channel QR code. Share this QR code only through a secure side channel (e.g., in person or via an encrypted messenger).

Messages on your private channel are unreadable to any node that does not possess the key, even if those nodes relay the encrypted packets.

PKI and Direct Messages (Meshtastic 2.5+)

Starting with Meshtastic 2.5, the platform introduced PKI-based direct messaging (X25519 ECDH key exchange with AES-CCM). Each node generates a public/private key pair. When you send a direct message (DM) to a specific node, it is encrypted to that node's public key - only the intended recipient's private key can decrypt it. This is separate from channel encryption and provides stronger guarantees for one-to-one communication. Note that DMs to nodes running firmware 2.4.3 or older are unprotected, there is no forward secrecy ("harvest now, decrypt later" applies), and there is no key revocation.

What Encryption Does NOT Protect

Meshtastic encryption protects message content, but several pieces of information remain visible to anyone monitoring the RF spectrum:

- **Packet metadata:** Source node ID, destination node ID, hop count, packet ID, SNR, and timing are in the unencrypted packet header and visible to any LoRa receiver tuned to the frequency.
- **Node existence:** Even on a private channel, NodeInfo packets (which advertise node IDs and positions) are broadcast on the public mesh. A passive observer can know your node exists and track its position even if they cannot read your messages.
- **Traffic analysis:** An attacker can observe transmission patterns - when and how often you transmit - without ever decrypting content. This can reveal usage patterns and correlate activity.

Admin Channel Security

Meshtastic's admin channel allows remote configuration of nodes. The admin channel PSK is a high-value secret: anyone who possesses it can reconfigure your nodes remotely, change channels, adjust power settings, or disable the device. Treat it accordingly:

- Use a unique PSK for the admin channel, separate from any user channels.
- Do not share the admin PSK with regular channel users.

- Store it securely (e.g., in a password manager).

Key Distribution and Revocation

Meshtastic has no built-in key revocation mechanism. If an admin channel or private channel PSK is compromised, you must manually change it on every node that uses it. For networks with many nodes, this can be operationally complex - plan key distribution carefully and limit who has access to keys from the outset. Because Meshtastic channel encryption has no forward secrecy, rotating a compromised PSK protects only future traffic. Any ciphertext an attacker captured over RF before the rotation can be decrypted with the old key. Assume all traffic sent under a leaked PSK is exposed.

Revision #5
Created 2026-05-03 04:15:59 UTC by Mesh America Admin
Updated 2026-06-09 12:03:43 UTC by Mesh America Admin