Security & Privacy

- Meshtastic Channel Encryption
- Privacy Best Practices

Meshtastic Channel Encryption

How Meshtastic Encryption Works

Each Meshtastic channel uses **AES-256-CTR** (counter mode) encryption with a 256-bit pre-shared key (PSK). Any node that has both the correct channel name and the correct key can decrypt messages on that channel. Encryption is end-to-end across the mesh: relay nodes forward ciphertext without decrypting it. 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 is the byte `0x01` followed by zeros. 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 2.3+)

Starting with Meshtastic 2.3, the platform introduced PKI-based direct messaging. 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.

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, 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.

Privacy Best Practices

Position Privacy

By default, Meshtastic nodes broadcast GPS coordinates to the entire channel at regular intervals. Your approximate location is visible to all channel participants. If your node is on the Default public channel, your position also appears on aggregation sites such as meshmap.net.

Mitigations

- **Use a private channel:** Position broadcasts on a private channel are only visible to nodes with the key.
- **Disable the position module:** Navigate to *Config* → *Position* → *GPS Mode* → *DISABLED*. The node will not broadcast any location data.
- **Set a fixed imprecise location:** Instead of live GPS, configure a fixed position - use the center of your neighborhood or a nearby landmark rather than your exact address. This allows you to appear on mesh maps without revealing your precise location.
- **Reduce position precision:** Meshtastic supports configurable position precision levels. Setting a lower precision (e.g., nearest kilometer) reduces location exposure while still being useful for routing and map display.

Node Naming

Your node's long name is broadcast to everyone on the channel and appears on public mesh maps. Use a callsign, handle, or alias rather than your full real name if privacy is a concern. Your node's short name (4 characters) is used in the mesh and is also public.

Telemetry Data

If you enable device telemetry or environmental sensors, data such as battery voltage, temperature, humidity, and pressure are broadcast on the channel. This data can reveal:

- Whether your device is plugged in or on battery (usage patterns)
- Environmental conditions at your location (indirectly revealing location details)
- When the device is active or idle

Disable telemetry modules you do not need, especially on nodes deployed in sensitive locations.

Public Mesh Visibility

If your node is on the Default channel, sites like meshmap.net automatically aggregate and display:

- Node long name and short name
- GPS position
- Last-heard timestamp
- Hardware model
- Firmware version

This information is public and indexed. Consider the public mesh as a zero-privacy environment.

Sensitive Use Cases

For deployments where participant safety depends on privacy (e.g., domestic violence shelters, witness protection situations, activist networks operating in hostile environments), apply all of the following:

- Use a private channel with a strong random PSK generated by the app.
- Disable GPS entirely (Config → Position → GPS Mode → DISABLED).
- Use an alias or callsign as the node name - never a real name.
- Disable all telemetry modules.
- Consider using **MeshCore** instead of Meshtastic (see below), which has a fundamentally different routing architecture that does not broadcast NodeInfo to the whole network by default.

Meshtastic vs. MeshCore Privacy Profile

The two platforms have meaningfully different privacy characteristics as a consequence of their routing architectures:

Meshtastic (Flood Routing)

Meshtastic uses a managed-flood routing model. NodeInfo packets - advertising each node's ID, name, and position - are broadcast to the entire reachable mesh. By design, all nodes on the

channel learn about all other nodes. This is high-transparency by design and enables features like mesh maps, but it means even passive observers on the channel receive location and identity data for all nodes.

MeshCore (Path-Based Routing)

MeshCore uses a path-based (source-routed) architecture. Nodes primarily exchange routing information with nodes they have direct routes to; advertisements are directed rather than broadcast to the whole mesh. In dense networks, this means a node's existence and location may not be known to distant nodes that have no route to it. MeshCore can be more privacy-preserving in scenarios where blanket network-wide NodeInfo broadcasting is undesirable.

Neither platform should be considered a complete anonymity solution - LoRa transmissions are detectable by anyone with appropriate radio hardware, regardless of the software layer's privacy features.