

Channel Security and Private Networks

MeshCore's channel system organizes mesh traffic into communities of interest. Understanding what public and private mean in the MeshCore context is essential for anyone deploying MeshCore in environments where confidentiality matters.

How Channel Keys Work

Each MeshCore channel is identified by a name and protected by a 16-byte (128-bit) secret key. The channel name and channel secret are separate values - the name is a human-readable label, while the secret is the actual cryptographic key used to encrypt and authenticate channel traffic. Channel traffic is encrypted with AES-128 in ECB mode and carries a 2-byte (16-bit) truncated HMAC-SHA256 MAC.

Channel configuration (name + secret) can be shared between devices via QR code, using the format:

```
meshcore://channel/add?name=ChannelName&secret=<32-hex-chars>
```

where the secret is a 16-byte value expressed as 32 hexadecimal characters (verify the exact scheme against the MeshCore QR code formats page before hand-constructing a link). Note that the default **public** channel uses a well-known, publicly documented key (`8b3387e9c5cdea6ac9e5edbaa115cd72`), so do not confuse the public default key with a private secret you generate.

Public Channels

A public channel uses a well-known channel name and a shared secret that is distributed openly within a community (the default public channel's key is publicly documented). Any node configured with the same channel name and secret can join and read all traffic. Because the key is publicly known, public-channel traffic is readable by anyone in radio range — it is not private or secure against observers.

Public channels are appropriate for:

- General-purpose mesh traffic
- Broad-access community traffic where confidentiality is not required
- Testing environments
- Wide-area mesh backbones carrying routing advertisements

Do not use a public channel for sensitive emergency coordination. Public channels offer no confidentiality (the key is publicly known) and no sender authentication (any participant can forge messages from any name). For emergency nets, use a private channel with a secret key, and treat even that as group-shared, not per-sender authenticated.

Nodes on a public channel broadcast their advertisements including display names and public keys to all other nodes. Traffic analysis (who communicates with whom, at what signal strength) is visible to all participants and to any passive radio receiver tuned to the correct LoRa parameters.

Private Channels

A private channel uses a channel secret that is kept confidential within the intended community. Only nodes configured with the correct channel secret can decode traffic on that channel.

Private channels are appropriate for:

- Closed community networks such as neighbourhood groups, amateur radio clubs, and emergency response teams
- Commercial or industrial deployments requiring channel isolation
- Multi-tenant mesh scenarios where multiple independent groups share physical infrastructure

Configuring a Private MeshCore Channel

1. **Generate a channel secret:** Use a cryptographically random 16-byte value. A password manager or command like `openssl rand -hex 16` works well. Avoid predictable values.
2. **Configure the channel in the MeshCore app:** Go to channel settings and enter the channel name and secret.
3. **Share the channel configuration:** Use the QR code export feature in the MeshCore app to share the channel with trusted members. Anyone who receives the QR code will have access to the channel.

What Channel Encryption Provides and Does Not Provide

Channel encryption provides:

- Confidentiality of message content from nodes not on the channel (only for private channels with a secret key; the public channel's key is publicly known, so it provides no confidentiality)
- Group-level authentication only — any holder of the channel key can generate valid messages under any display name. There is no per-sender authentication on channel messages, and the 2-byte (16-bit) MAC is forgeable by an active attacker (on the order of ~32k attempts); it is an integrity check, not strong authentication.

Channel encryption does NOT provide:

- Perfect forward secrecy - the same key is used indefinitely; compromise of the key reveals all past and future traffic
- Individual message authentication - unlike direct messages (which use per-pair ECDH keys), channel messages are authenticated only by possession of the shared key, so any key-holder (or anyone who obtained the key from a lost device or leaked QR code) can forge a message attributed to any other member
- Protection against traffic analysis - signal strength, transmission timing, and node identifiers are visible to any receiver tuned to the correct LoRa parameters

For communications requiring stronger guarantees, use direct (unicast) messages, which use per-pair ECDH key agreement providing individual authentication and stronger confidentiality. Note the caveats: direct messages still use AES-128-ECB, a 16-bit (2-byte) MAC, and static keys (no forward secrecy), and their authentication depends on having verified the peer's public key out-of-band.

Source: MeshCore QR code format documentation (github.com/meshcore-dev/MeshCore)

Revision #4

Created 2026-05-03 06:20:55 UTC by Mesh America Admin

Updated 2026-06-09 14:39:46 UTC by Mesh America Admin