PKC Direct Messaging (v2.5+)

Meshtastic v2.5 introduced **Public Key Cryptography (PKC) encrypted direct messages** - a significant security upgrade that encrypts DM content per-recipient rather than relying only on the shared channel key. Note that PKC protects the message *body*: the packet header (source/destination node IDs, hop count, timing) remains plaintext and can be uplinked to MQTT, so PKC protects what you say, not the fact that two nodes are communicating.

Note on terminology: This feature is officially called "PKC Direct Messages" or "encrypted direct messages" in Meshtastic documentation. PKC encrypted direct messages were introduced in **firmware v2.5.0** (not v2.3 or v2.4 as some sources incorrectly state).

Before v2.5: How DMs Worked

Prior to v2.5, "direct messages" in Meshtastic were standard channel messages with a `to` field set to the recipient's node ID. Anyone on the same channel with the channel key could decrypt and read all DMs. There was no per-recipient encryption.

v2.5+: PKC Encrypted Direct Messages

From v2.5 onward, direct messages use per-node asymmetric encryption:

- **Key exchange: X25519 ECDH** - each node has an X25519 public/private key pair
- **Encryption: AES-CCM** - using the derived shared secret as the key (per the Meshtastic encryption documentation)
- Only the intended recipient can decrypt the message - the channel key is not used
- Node public keys are distributed automatically via NodeInfo packets

Limitations: PKC has **no forward secrecy** — keys are long-lived, so the "harvest now, decrypt later" risk applies if a node's private key is ever compromised — and there is **no key revocation** mechanism. The plaintext packet header still leaks metadata even on PKC DMs.

Backward Compatibility

If you send a PKC-encrypted DM to a node running firmware 2.4.3 or older, Meshtastic automatically falls back to the legacy channel-based method (which is unprotected against anyone holding the channel key). Some app versions display a key/lock icon indicating that PKC was used for a given message.

Requirements

- Both sender and recipient must be running Meshtastic firmware **v2.5 or later**
- Both nodes must have exchanged NodeInfo packets (public keys are included automatically)
- Compatible with Android, iOS, and Python CLI clients that support v2.5+

Source: meshtastic.org/docs/overview/encryption/ and meshtastic.org/blog/introducing-new-public-key-cryptography-in-v2_5/. Verified 2026-05-03.

Revision #5
Created 2026-05-03 05:43:24 UTC by Mesh America Admin
Updated 2026-06-09 12:13:13 UTC by Mesh America Admin