MeshCore Packet Format and Encryption

This page covers MeshCore's packet encryption as verified from `docs/packet_format.md`, `docs/payloads.md`, and `src/Utils.cpp` in the official MeshCore repository.

Encryption at the Packet Level

Encrypted payload types (text, group, request/response) use AES-128 in ECB mode with a 2-byte truncated HMAC-SHA256 MAC, in an encrypt-then-MAC construction. Crucially, the 2-byte MAC is **prepended** to the ciphertext (it precedes the encrypted data), not appended. Note that advertisements (`PAYLOAD_TYPE_ADVERT`) and control packets are sent unencrypted, so not all traffic is confidential. The ECB mode and 16-bit MAC limit the strength of this protection - see the security/encryption overview for caveats.

```
Direct message payload: [dest hash (1 byte)] [src hash (1 byte)] [2-byte cipher MAC] [AES-128-ECB ciphertext]
Group message payload:  [channel hash (1 byte)] [2-byte cipher MAC] [AES-128-ECB ciphertext]
```

In both layouts the 2-byte cipher MAC **precedes** the AES-128 ciphertext, and a hash prefix (destination/source hash for direct messages, channel hash for group messages) comes before the MAC. This matches `Utils::encryptThenMAC` (which writes the MAC to the first bytes and the ciphertext after it) and the field order in `docs/payloads.md`.

Route Types

Packets carry one of four route types (from packet_format.md):

- `ROUTE_TYPE_FLOOD` - broadcast to all repeaters; used for initial contact and group messages
- `ROUTE_TYPE_DIRECT` - embeds a specific repeater path; only listed repeaters forward the packet
- `ROUTE_TYPE_TRANSPORT_FLOOD` - flood with transport/region code prefix
- `ROUTE_TYPE_TRANSPORT_DIRECT` - direct-routed with transport/region code

Path Learning (How Direct Routing Works)

MeshCore uses a flood-then-direct-route mechanism (not AODV path discovery/acknowledgment):

1. First message to a new destination is flood-routed
2. The destination node returns a `PAYLOAD_TYPE_PATH` packet containing the full repeater path it received the message through
3. The sender stores this path and uses `ROUTE_TYPE_DIRECT` for subsequent messages, embedding the learned path
4. Only the specific repeaters in the path forward the packet - all others ignore it

This mechanism reduces channel load significantly compared to pure flooding once paths are established. This benefit assumes a stable topology with repeated traffic between the same pairs. In mobile or rapidly-changing deployments (common in emergencies), learned paths break frequently, forcing re-floods and reducing or eliminating the savings - in the worst case the network can degrade toward continuous flooding plus failed direct sends.

Source: docs/packet_format.md, docs/payloads.md, and src/Utils.cpp in the official MeshCore repository.

Revision #6
Created 2026-05-03 04:19:16 UTC by Mesh America Admin
Updated 2026-06-09 14:39:39 UTC by Mesh America Admin