MeshCore Security and Encryption

MeshCore uses a layered cryptographic system verified from the project's source code. All claims on this page are sourced from `src/Utils.cpp`, `src/MeshCore.h`, and `src/Identity.h` in the official MeshCore repository.

Symmetric Encryption

- **Algorithm: AES-128** (ECB mode with zero-padding for the final block)
- Key size: 16 bytes (`CIPHER_KEY_SIZE = 16`)
- The shared AES key is derived via ECDH (see below)

Message Authentication

- **MAC: HMAC-SHA256** truncated to 2 bytes (`CIPHER_MAC_SIZE = 2`)
- Scheme: **Encrypt-then-MAC** - the ciphertext is MACed, not the plaintext
- Functions: `encryptThenMAC` / `MACThenDecrypt`

Key Exchange

- **ECDH via X25519** - Ed25519 identity keys are transposed to X25519 for Diffie-Hellman key exchange (`calcSharedSecret` in Identity.h)
- The resulting shared secret is used as the AES-128 key for the session

Identity and Signing

- **Identity keys: Ed25519**
- Public key size: 32 bytes (`PUB_KEY_SIZE = 32`)
- Private key size: 64 bytes (`PRV_KEY_SIZE = 64`)
- Signature size: 64 bytes (`SIGNATURE_SIZE = 64`)
- **Advertisements are signed** with Ed25519 to prevent node identity spoofing

What This Means in Practice

- Messages between two MeshCore nodes use a unique AES-128 key derived from their ECDH exchange - no shared secret needs to be pre-distributed
- The 2-byte HMAC provides integrity checking (detects tampering) with low overhead
- Node identities are cryptographically verified - a node cannot impersonate another node's public key
- Channel/group messages use a shared symmetric key derived from the channel configuration

Source: Official MeshCore repository, src/Utils.cpp, src/MeshCore.h, src/Identity.h. Verified 2026-05-03.

Revision #3
Created 2026-05-03 03:55:57 UTC by Mesh America Admin
Updated 2026-05-03 12:58:37 UTC by Mesh America Admin