InovaMesh

*The Agile Cyber Security Solution by*

CyberInova®

# Security Features - Cryptographic Protocols and Certificate Management in InovaMesh ZTNA

CyberInova Ltd

Dec 2024

# Contents

# The InovaMesh Security Model

This document provides a detailed overview of the security architecture implemented in the InovaMesh platform, emphasizing the Zero Trust principles enforced across both the Data Layer and the Control Layer.

The Control Layer, based on DYNOC (Patent Pending by Cyberinova Ltd), is responsible for Zero-Touch Configuration (ZTC) and centralized node management within the mesh. By applying Zero Trust security at every stage, InovaMesh guarantees robust protection, integrity, and comprehensive control across the entire ecosystem.

The Data Layer, powered by Nebula (by Defined Networking), ensures secure and authenticated data exchange between nodes. Nebula is an open project that delivers a secure, high-performance Data Layer for modern distributed networks. Its open-source architecture provides robust encryption, certificate-based authentication, and seamless connectivity between nodes, even across complex topologies.

Cyberinova Ltd selected Nebula as the Data Layer for its InovaMesh solution because it ensures strong data confidentiality and integrity, supports zero-trust principles, and enables flexible, scalable deployment across diverse environments. As an open project, Nebula also offers transparency, community support, and ongoing innovation, making it the ideal foundation for InovaMesh.

# Data Layer Communication for InovaMesh Devices

| Component | Algorithm(s) | Purpose | Security Level | Details / Notes |
|---|---|---|---|---|
| Key Exchange / Handshake | Curve25519 / X25519 (ECDH) | Node authentication, key exchange | SECRET (128-bit symmetric equiv.) | Fast, secure elliptic curve. Used during session establishment. |
| Traffic Encryption | AES-256-GCM<br>ChaCha20-Poly1305 | Data encryption and authentication | TOP SECRET (256-bit symmetric equiv.) | Both are CNSA/NIST recommended for top-tier classified data. |
| Hashing & Integrity | SHA-256 | Hashing, certificate and data integrity | TOP SECRET (widely accepted) | Used for signatures, integrity checks, certificate validation, etc. |
| Certificates | X25519 public key (signed by CA) | Node identity management | SECRET (by X25519), CA may use stronger algorithms | Certificates validate node identity and are signed by a trusted authority. |
| Lighthouse | N/A (metadata only) | Peer discovery and coordination | N/A (no user data exposure) | Does not decrypt or inspect user traffic, only aids in peer discovery. |
| Relay | AES-256-GCM / ChaCha20-Poly1305 | Relaying encrypted traffic between nodes | TOP SECRET (256-bit symmetric equiv.) | Relays forward fully encrypted packets; they cannot decrypt or view user data. |

## Data Layer Key Points

- **Key exchange (ECDH X25519) provides SECRET-level security**: Sufficient for most use cases, including many government environments, but not generally recommended for TOP SECRET-level key exchange per some governmental policies.
- **Traffic encryption with AES-256-GCM or ChaCha20-Poly1305 provides TOP SECRET-level security**: Both algorithms are recommended for the highest level of classified information by NIST and CNSA (NSA).
- **Hashing with SHA-256 is also considered secure for TOP SECRET** (though NSA CNSA now recommends SHA-384+ for new TOP SECRET deployments).
- **Certificates use X25519 public keys**, which are suitable for SECRET; CA (Certificate Authority) signatures may use RSA or ECDSA, with security dependent on key length.
- **Lighthouse nodes** only facilitate peer discovery; they do not handle or see actual mesh traffic.
- **Relay nodes** act as intermediaries to forward encrypted traffic between nodes that cannot connect directly (e.g., due to NAT/firewall).
- **Security:** Relays never decrypt the traffic—they simply forward already-encrypted packets. The end-to-end encryption (AES-256-GCM or ChaCha20-Poly1305) remains intact, preserving the same TOP SECRET security level for the actual data

## Summary Table of Security Levels and Post-Quantum Readiness

| **Algorithm** | **Security Level** | **Where Used** | **Post-Quantum Readiness** |
|---|---|---|---|
| Curve25519 / X25519 (ECDH) | SECRET (128-bit equivalent) | Handshake, key exchange, certificates | ❌ Not Ready<br>Vulnerable to quantum attacks (Shor's algorithm can break ECC efficiently). |
| AES-256-GCM | TOP SECRET (256-bit equivalent) | Traffic encryption | ✅ Quantum-Resistant<br>Resistant except to Grover's algorithm (which halves security to 128 bits; still considered strong). |
| ChaCha20-Poly1305 | TOP SECRET (256-bit equivalent) | Traffic encryption (alternative) | ✅ Quantum-Resistant<br>Same as AES-256: only Grover's applies, so 128-bit effective security. |
| SHA-256 | TOP SECRET (practically) | Hashing, integrity, certificates | ⚠️ Reduced Margin<br>Grover's algorithm reduces preimage resistance from 256 to 128 bits. For higher margin, use SHA-384+. |

---

**Legend**

- ✅ **Quantum-Resistant**: Secure against all known quantum attacks except for quadratic speedup (Grover).
- ⚠️ **Reduced Margin**: Security margin reduced, but not broken (e.g., SHA-256 to 128 bits).
- ❌ **Not Ready**: Broken or severely weakened by quantum computers (e.g., ECC, RSA).

---

**Notes**

- **AES-256-GCM and ChaCha20-Poly1305**: Both remain robust in a post-quantum world, as their only vulnerability is to Grover's algorithm, which is still impractical at key sizes used.
- **SHA-256**: Preimage resistance drops to 128 bits against quantum attackers, so for the highest post-quantum assurance, use SHA-384 or SHA-512 (standard guidance: "double your hash length").
- **Curve25519/X25519**: Not post-quantum; will be the first weak link in a post-quantum attack scenario.

# Data Layer Certificate and Key Management Workflow

The Data Layer employs a public key infrastructure (PKI) model to manage identities and secure communication within the mesh. The process unfolds as follows:

1. Certificate Authority (CA) Generation
   - The administrator uses Data Layer's certificate generation tool to create the CA certificate (ca.crt) and the corresponding CA private key.
   - The CA certificate acts as the trust anchor for the entire mesh network.
2. Device Certificate and Key Generation
   - Using the CA certificate and private key, the tool generates a unique device certificate (device.crt) and device private key (device.key) for each mesh node (device).
   - Each device certificate includes:
     - The device's public key (generated during this process)
     - The device's Data Layer IP address (or address range)
     - Group memberships and additional attributes (such as subnets or roles)
   - The CA signs each device certificate, establishing trust throughout the mesh.
3. Distribution to Devices
   - Each device is provisioned with:
     - The CA public certificate (ca.crt)
     - Its own device certificate (device.crt)
     - Its own device private key (device.key)
   - No device ever holds the CA private key; this remains securely with the administrator.
4. Mesh Authentication and Tunnel Establishment
   - When a device joins the Data Layer mesh, it presents its signed device certificate.
   - Using the CA public certificate, every device can verify the authenticity of its peers.
   - Devices leverage the Lighthouse service to discover each other's reachable addresses and network information.

    - Once discovery is complete, devices mutually authenticate using their certificates and establish an encrypted tunnel (based on Curve25519/X25519 key exchange) for secure communication.
5. Access Control and Segmentation
    - The information encoded in the device certificate (such as IP addresses and group memberships) is used to enforce network segmentation, ACLs, and routing within the Data Layer mesh.

---

**Summary**

- **CA generates and signs all device certificates.**
- **Devices receive only what they need: the CA public certificate, their own certificate, and their own private key.**
- **No device can impersonate another or act as a CA.**
- **Lighthouse nodes enable peer discovery, but every tunnel is authenticated and encrypted end-to-end using the established credentials.**

# Control Layer communication for InovaMesh devices via AWS IoT Core

InovaMesh devices leverage AWS IoT Core as a secure, scalable MQTT broker to interconnect with the BrainEngine platform. Each device establishes a mutually authenticated TLS connection to AWS IoT Core, ensuring encrypted data exchange and robust device identification using X.509 certificates. This architecture enables reliable, real-time communication between distributed InovaMesh endpoints and the BrainEngine, supporting advanced mesh networking features and centralized management, while maintaining industry-leading security and compliance standards.

**AWS IoT Core MQTT TLS Security: Detailed Summary**

**1. Connection Security Overview**

- **Transport Protocol:** TLS 1.3 (latest and most secure version)
- **Authentication:** Mutual TLS (mTLS) with X.509 certificates
- **Certificate Chain:**
    - Issued by Amazon RSA 2048 M01, rooted in Amazon Root CA 1 (widely trusted CAs)
    - Server certificate uses RSA 2048-bit keys and is signed with RSA-PSS + SHA256 (a robust modern signature algorithm)
- **Key Exchange (Handshake):**
    - Uses **ECDHE X25519** (Curve25519), providing strong Perfect Forward Secrecy (PFS)
- **Session Encryption:**
    - Negotiated cipher suite is **TLS_AES_128_GCM_SHA256**
        - Data is encrypted with AES-128-GCM (secure, authenticated encryption)
        - Integrity protected by SHA-256

### 2. Security Properties

- **Confidentiality:**
  All data in transit between device and AWS is fully encrypted.
- **Authentication:**
  Both sides (client and server) are authenticated using trusted certificates (mTLS).
- **Integrity:**
  Each message is protected against tampering with modern authenticated encryption (GCM) and SHA-256.
- **Perfect Forward Secrecy:**
  Each TLS session uses an ephemeral (temporary) key, so even if a long-term private key is compromised, past traffic remains protected.
- **Industrial Compliance:**
  Meets or exceeds requirements for ISO 27001, SOC, HIPAA, and most industrial IoT standards.
- **Post-Quantum:**
  Like all standard TLS connections today, not yet quantum-safe at handshake/certificate level.

### 3. Cryptographic Details Table

| **Step** | **Algorithm / Key** | **Security Level** | **Post-Quantum Readiness** |
|---|---|---|---|
| Server Certificate | RSA 2048 / RSA-PSS + SHA256 | Commercial / SECRET | ❌ Not Ready (broken by quantum; RSA-2048 not quantum-safe) |
| Handshake (Ephemeral) | ECDHE X25519 (Curve25519) | SECRET (128-bit equivalent) | ❌ Not Ready (ECC not quantum-safe) |
| Session Encryption | AES-128-GCM (TLS 1.3) | SECRET (128-bit equivalent) | ⚠️ Partially (quantum reduces to 64 bits; for higher security, use AES-256-GCM) |
| Integrity / Hashing | SHA-256 | SECRET / Practically TOP SECRET | ⚠️ Reduced Margin (quantum reduces strength to 128 bits) |

## Control Layer Key Points

- **This connection is extremely secure by today's standards.**
- **All data is encrypted and authenticated, with forward secrecy and mutual authentication.**
- **Weakest link, security-wise, is at handshake/certificate level due to quantum vulnerability (like all standard TLS today).**

# Control Layer Device Identification and Control

Cyberinova Ltd has implemented an advanced suite of security measures, based on the Control Layer DYNOC (Patent Pending) and the Data Layer, that provide robust protection against both advanced attacks on individual nodes and unauthorized modifications to mesh configurations.

For example, each InovaMesh device is uniquely identified by a 10-digit serial number derived from hardware-bound UIDs (such as disk, firmware, or hypervisor-generated identifiers). The InovaMesh software installed or pre-installed on the device is generic and untyped, making it suitable for any customer or mesh deployment. Once the serial number is registered in the central InovaMesh application database and the device is authorized to join a specific mesh, the authorization process is completed and all operational information is delivered to the device via the Control Layer. During the authorization phase, the MAC address of the device's WAN interface is also captured. Any subsequent authorization requests—such as those occurring after a reboot—must present the same MAC address.

This comprehensive security architecture ensures that the integrity, authenticity, and confidentiality of the entire mesh network are maintained—even in the face of sophisticated threats targeting devices or attempting to alter critical configuration data:

- **Dynamic Configuration Generation:**
  The DDYNOC service dynamically generates configuration files, alongside the version number and a SHA256 hash of the configuration. This hash is obfuscated using the device's Data Layer private key, ensuring both integrity and confidentiality.
- **Automated Integrity Verification:**
  The Cyberinova Ltd release of the Data Layer service executable automatically recalculates this hash at startup and processes the configuration only if it remains unaltered, preventing any unauthorized changes from being applied.
- **X25519 Private Key Encryption:**
  The X25519 private key is encrypted and dynamically injected by the DDYNOC service. As a result, Data Layer service executable builds lacking integrity checks cannot access or utilize the key, ensuring the highest level of key protection.
- **Key Irrecoverability Without Authorization:**
  Without the required device-specific keys, it is impossible for any Data Layer version to recover the private key body. Consequently, unauthorized software cannot insert a node into the mesh or negotiate tunnels.
- **Multi-Layer Encryption Architecture:**
  Encryption of the X25519 key leverages two independent keys:
  - **Device Key:** derived from unique hardware identifiers. This means cloned software cannot function on non-original hardware.
  - **Data Layer Key:** key generated via a proprietary algorithm. This process is non-standard and cannot be easily replicated or reverse-engineered.

## Control Layer - Security Targets Achieved

| Security Target | Description | Achieved By |
|---|---|---|
| Configuration Anti-Tampering | Detects and blocks unauthorized configuration changes via SHA256 hash validation. | SHA256 hash (obfuscated with device private key), validated at every startup. |
| Configuration Integrity | Accepts and processes only unaltered, authorized configurations. | Integrity check on the first line; configuration rejected if the hash does not match. |
| X25519 Private Key Protection | Ensures the private key is encrypted and inaccessible without the required keys. | Private key body encrypted by DDYNOC; not accessible or restorable by standard Data Layer builds. |
| Hardware Binding | Binds configuration and keys to specific hardware, preventing use on unauthorized devices. | Device key derived from unique hardware identifiers and serial number; fails if moved to other hardware. |
| Software Build Binding | Binds configuration to a specific Data Layer service executable; prevents use with unauthorized software builds. | Data Layer key linked to the executable via a proprietary algorithm. |
| Anti-Replay & Anti-Rollback | Prevents acceptance of old, replayed, or reused configurations, blocking rollback attacks. | Unique hash per configuration; only the latest valid configuration is accepted. |
| Prevention of Unauthorized Builds | Prevents custom or unofficial Data Layer service executable builds from accessing or reconstructing private keys or config files. | Key protection logic and Data Layer key algorithm are proprietary and absent in unauthorized builds. |
| Authentication and Authorization | Guarantees that only authorized nodes can join the mesh and establish tunnels. | Validation of configuration and keys required for every network join attempt. |

# Critical Analysis of Kernel Firewall Configuration

Below is the result of the critical analysis of the kernel firewall configuration on an OpenWRT device.
No issues have been identified except for port 4381, which is managed by the Data Layer service.
**The Control Layer (DDYNOC) does not present any concerns regarding open ports.**

Note on IGMP: The IGMP (Internet Group Management Protocol) rules present in the firewall configuration do **not represent a real security risk**. IGMP is used exclusively for multicast group management and does not expose any services or sensitive data on the device. However, if multicast services (such as IPTV) are not required in your environment, IGMP can be safely disabled in the firewall.

### Open Ports on WAN - Firewall Configuration Summary

| Interface | Protocol | Port | Direction | Description / Notes |
|---|---|---|---|---|
| WAN | UDP | 4381 | IN | **Open! The Data Layer service is listening and publicly reachable from the internet** |
| WAN | UDP | 67→68 | IN | DHCP renew (required for WAN IP renewal; only responds to DHCP servers, not a public/exploitable service) |
| WAN | IGMP | — | IN | IGMP multicast management (used for IPTV/multicast; not an open TCP/UDP service) |

---

### Additional Notes

- **No TCP ports are open on WAN.**
- **SSH (22), HTTPS (443), ICMP (ping), and all other standard services are closed on WAN.**
- **No port forwarding or DMZ exposes LAN or mesh services to WAN.**
- **UDP port 4381 is open and actively serviced by the Data Layer, making it accessible from the public internet.**
- **DHCP and IGMP rules are required for connectivity and do not represent a general security risk.**

### Verbal Report – Nmap Firewall Validation

A targeted UDP scan was performed using Nmap with service detection enabled, specifically on ports 67, 68, and 4381 of the device at IP address 192.168.10.153.

The results are as follows:

- **UDP port 67** (DHCP server) and **UDP port 68** (DHCP client) are reported as open|filtered, which is expected for DHCP services and does not indicate an active open service unless a DHCP request is made.
- **UDP port 4381** is explicitly reported as open and returns data, confirming the presence of a listening service. This port corresponds to the Data Layer service, as described in the firewall documentation.

### Verbal Report – UDP Port 4381 Service Detection

An Nmap UDP scan was performed on port 4381 of the target host.

The result confirms that the port is open and that the service responds to probes, returning specific data patterns when queried (for example, in response to NTP and DTLS session requests).

This behavior verifies the presence of an active service on UDP port 4381, consistent with the Data Layer service as documented in the firewall configuration analysis.

### Summary:

The scan validated that UDP port 4381 is open and actively responding, confirming the firewall's documented configuration and the correct operation of the Data Layer service on this port.

```
~$ sudo nmap -Pn -sU -p 4381 -sV 192.168.1.78
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-01 18:44 CEST
Stats: 0:01:37 elapsed; 0 hosts completed (1 up), 1 undergoing Service
Scan
Service scan Timing: About 0.00% done
Nmap scan report for InovaMesh.station (192.168.1.78)
Host is up.

PORT     STATE SERVICE VERSION
4381/udp open  unknown
1 service unrecognized despite returning data. If you know the
service/version, please submit the following fingerprint at
https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port4381-UDP:V=7.94SVN%I=7%D=7/1%Time=6864107E%P=x86_64-pc-linux-
gnu%r(
SF:NTPRequest,10,"\x12\0\0\0\0\x01\0\0\0\0\0\0\0\0\0\0")%r(DTLSSession
Req,
SF:10,"\x12\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0");
```

```
Service detection performed. Please report any incorrect results
```

**Verbal Report – Nmap Full TCP Port Scan Validation**

A comprehensive TCP port scan was conducted using Nmap with service detection enabled, targeting all 65,535 TCP ports on the device at IP address 192.168.1.78.

**Scan Results:**

- All scanned TCP ports are reported as **filtered**, with no response from the host.
- No open or unfiltered TCP ports were detected.
- Service detection did not identify any accessible TCP services on this host.

**Conclusion:**

These results confirm that the device does not expose any TCP ports to the network.

This finding is consistent with the firewall configuration analysis, which indicates that all standard TCP services (including SSH, HTTPS, Telnet, and others) are properly closed or filtered on the relevant interfaces.

The security posture of the device is therefore validated, with no unnecessary TCP exposure detected.

```
$ sudo nmap -Pn -p- -sV 192.168.1.78
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-01 14:47 CEST
Nmap scan report for InovaMesh.station (192.168.1.78)
Host is up.
All 65535 scanned ports on InovaMesh.station (192.168.1.78) are in
ignored states.
Not shown: 65535 filtered tcp ports (no-response)
Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13447.01 seconds
```

## InovaMesh Stealth Mode

InovaMesh, leveraging the Relay functionality of the Data Layer, can operate with no active UDP ports exposed on the WAN interface, achieving a Stealth-like node behavior—except for the DHCP client port (unless static addressing is preferred). Provided that one or more Relays are available to cover the relevant areas and expected traffic volume, InovaMesh can ensure the highest levels of confidentiality and non-detectability. The use of Relays does not introduce any risk to the confidentiality or integrity of the traffic, as all data remains fully protected by end-to-end encryption.

**Formal Report – UDP Stealth Mode Scan Validation**

A targeted UDP scan was performed on the device at IP address 192.168.1.78 using Nmap with service detection enabled, specifically on ports 67, 68, and 4381. The scan was conducted with the system configured in "Stealth" mode (i.e., with no UDP ports deliberately exposed on the WAN interface).

**Scan Results:**

- **UDP port 67** (DHCP server) and **UDP port 68** (DHCP client) are reported as open|filtered. This is expected for DHCP-related ports, as they typically do not respond unless an actual DHCP transaction is initiated. No open service is confirmed.
- **UDP port 4381** is also reported as open|filtered. This indicates that Nmap did not receive a response on this port. No active service was detected or confirmed as exposed to the WAN.

**Interpretation:**

- The open|filtered state for all scanned ports means that the firewall or the system is not responding to unsolicited UDP probes on these ports, and thus does not reveal the presence of any listening service.
- This result is fully consistent with "Stealth" mode operation, in which no UDP ports are actively exposed to the WAN, and the node is effectively non-discoverable via these means.
- No evidence of exposed services was observed on the relevant UDP ports.

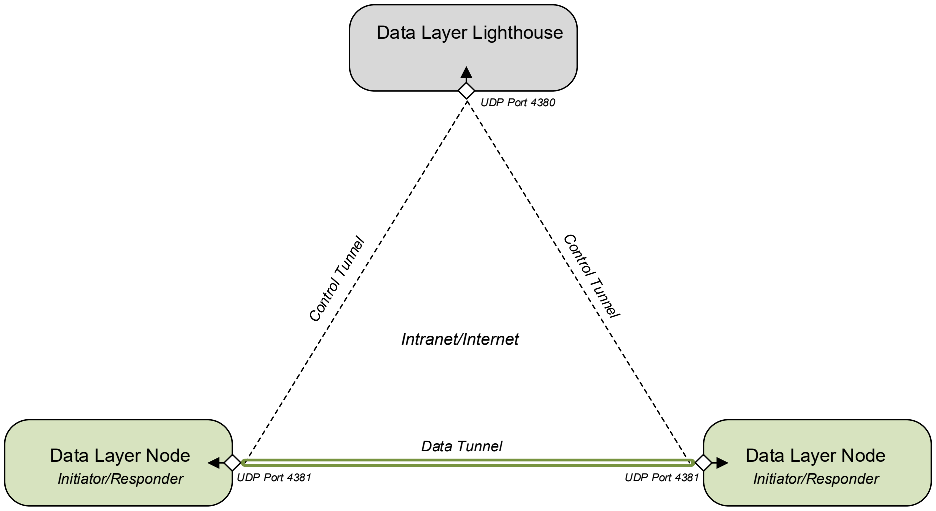
**Conclusion:**

The scan confirms that, in "Stealth" mode, the device does not present any active UDP ports on the WAN interface. This validates the effectiveness of the firewall and network configuration in ensuring a non-discoverable and confidential network posture.

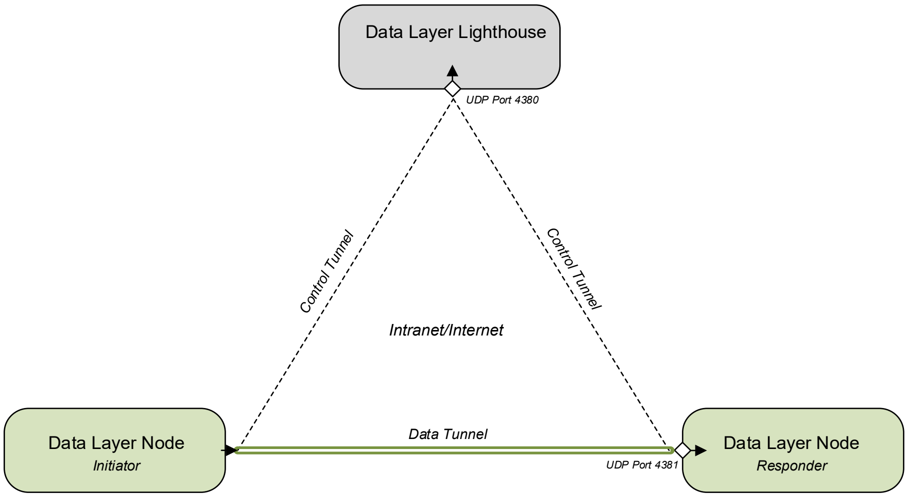
```
sudo nmap -Pn -sU -p 67,68,4381 -sV 192.168.1.78
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-02 18:47 CEST
Nmap scan report for InovaMesh.station (192.168.1.78)
Host is up.

PORT     STATE         SERVICE VERSION
67/udp   open|filtered dhcps
68/udp   open|filtered dhcpc
4381/udp open|filtered unknown

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 115.45 seconds
```

## Topologies of Stealth and Non-Stealth Modes

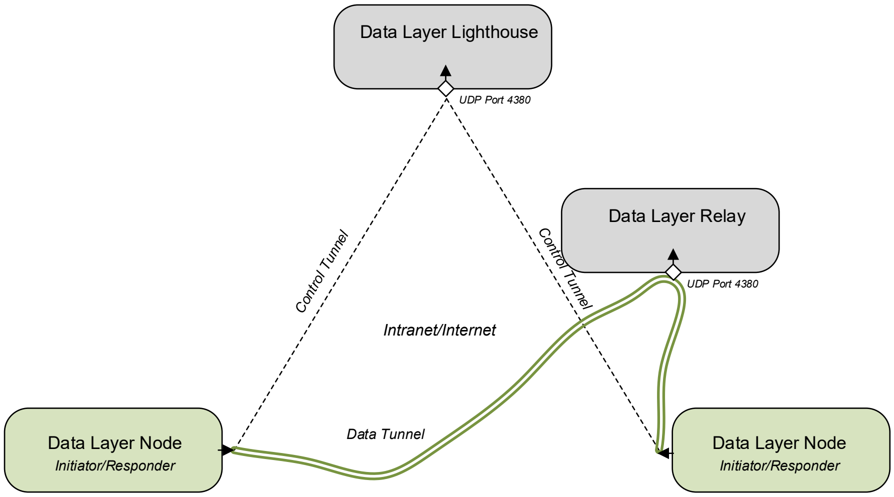
The diagram illustrates the nominal operating mode of the InovaMesh Data Layer, where any two nodes (in a Full Mesh topology) can establish a direct tunnel over the intranet or Internet for data exchange, under the supervision of the LightHouse. In this configuration, both nodes expose their UDP port (e.g. 4381) to accept incoming tunnel creation requests.



The diagram illustrates the Stealth mode of operation for the Initiator node (in a Full Mesh topology), where, under the supervision of the LightHouse, a direct tunnel is established to the Responder node over the intranet or Internet for data exchange. In this configuration, only the Responder node exposes its UDP port to accept incoming tunnel creation requests. The Initiator node does not expose any ports: it operates in Stealth mode.



Finally, this diagram shows the Stealth mode of operation for both nodes (in a Full Mesh topology), where, under the supervision of the LightHouse, an indirect tunnel is established through the Relay over the intranet or Internet for data exchange. In this configuration, neither node exposes a UDP port to accept tunnel creation requests: both operate in Stealth mode.


Data Layer Lighthouse
UDP Port 4380
Control Tunnel
Data Layer Relay
Control Tunnel
UDP Port 4380
Intranet/Internet
Data Tunnel
Data Layer Node
Initiator/Responder
Data Layer Node
Initiator/Responder