Scaling the Mesh: Best Practices for Securing DigiMesh Environments

Daniel Romero | March 12, 2026

Table of Contents

Disclaimer: The statements and observations in this document are based on the information available at the time of writing. They are subject to revision and may change in future updates or subsequent versions.

1. Introduction

Wireless mesh networking plays a key role in modern Industrial IoT (IIoT) deployments. In large or complex environments where cabling is impractical or costly, mesh networks provide reliable, self-healing communication with strong coverage and built-in redundancy, making them well suited for industrial environments.

At NCC Group, our experience in vulnerability assessments and security research across both public and proprietary network protocols, as well as embedded devices, has consistently reinforced one key principle: all devices and protocols must be properly hardened to achieve comprehensive security. In practice, most implementations tested contain small, or sometimes significant, gaps that, if addressed, could greatly strengthen the overall security of the environment.

In real-world deployments achieving both security and operational efficiency usually means leveraging multiple networking technologies such as Zigbee, LoRaWAN, and the one we focus on today: DigiMesh.

DigiMesh is a proprietary peer-to-peer networking topology developed by Digi International. Unlike traditional mesh networks that rely on dedicated "coordinator" or "router" nodes, DigiMesh treats every node as an equal and every node can help move data along. Even better, all nodes can sleep, which is a huge win for power-sensitive installations.

DigiMesh is widely used in scenarios where reliability, coverage, and autonomous operation are critical:

- **Agriculture**: soil monitoring, irrigation control and livestock tracking rely on DigiMesh's long sub-GHz range and synchronized sleep to run for years on battery power.
- **Critical infrastructure**: power substations, pipelines and remote monitoring require the kind of self-healing redundancy that DigiMesh's mesh routing provides.
- **Smart cities**: street lighting, parking sensors, environmental monitoring benefit from DigiMesh's ability to scale to hundreds of nodes while maintaining reliable data aggregation back to a central gateway.
- **Industrial and remote telemetry**: mining, oil & gas, water management require deterministic, low-latency delivery, capabilities that DigiMesh's provide.

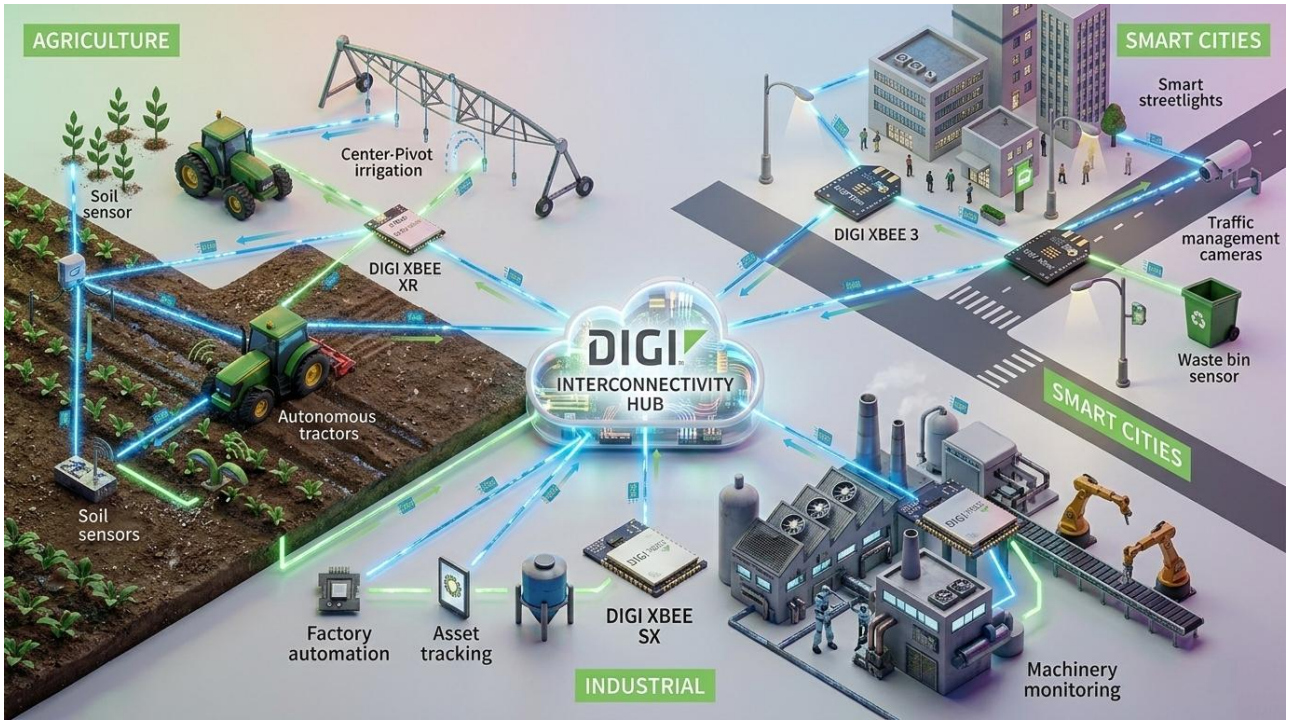

Figure 1: DigiMesh networking

The table below provides a high-level comparison of Zigbee, DigiMesh, and LoRaWAN to highlight their relative strengths and differences.

ZIGBEE VS. DIGIMESH VS. LORAWAN HIGH-LEVEL COMPARISON TABLE [1][2]

	Zigbee	DigiMesh	LoRaWAN
Topology	Hierarchical (coordinator, router, end device)	Flat mesh - all nodes equal	Star-of-stars (gateway)
Sub-GHz Support	No - 2.4 GHz only, typically <3 km	Yes (868/900 MHz) - up to 40+ km range	Yes - very long range, up to 15+ km
Encryption Key Length	128-bit AES	128-bit AES (256-bit AES is available in newer firmware versions)	128-bit AES
Sleeping routers	No	Yes	Yes
Payload Size	Up to 256 bytes when using unicast messaging	Up to 1024 bytes	Up to 242 bytes (DR-dependent)
Interoperability	Open standard, multi-vendor	Proprietary - Digi ecosystem only	Open standard (LoRa Alliance)

[1] *Wireless Mesh Networking: ZigBee vs. DigiMesh: https://www.astutegroup.com/wp-content/uploads/2024/02/Digi-International-wp_zigbeevsdigimesh.pdf*

[2] *IoT Wireless Sensor Networks: DigiMesh vs LoRaWAN: https://ncd.io/blog/iot-wireless-sensor-networks-digimesh-vs-lorawan/#elementor-toc__heading-anchor-0*

2. DigiMesh Modules - Technical Detail

Digi offers several XBee module variants for DigiMesh deployments. These variants differ not only in features and performance, but also in their compliance with regional frequency regulations, including the EU, the Americas, and APAC.

Understanding the distinctions in hardware capabilities, performance, and supported feature sets is essential, whether you are selecting the appropriate module for development or deployment, or defining the scope and limitations of a module during a security assessment.

The table below summarizes the most frequently observed modules during our testing activities:

	XBee 3[3]	XBee SX[4][5]	XBee XR[6][7]
Potential Use Case	Global deployments, Smart Building, IoT	Agriculture / Industrial / Critical Infrastructure	Agriculture / Industrial / Critical Infrastructure
Frequency	2.4 GHz	868 / 900 MHz	868 / 900 MHz
Max Distance	<1.6 km (PRO)	Up to ~14.5 km (868) / ~28 km (900)	Up to ~17 km
Encryption	128-bit AES (ECB) / 256-bit AES (CTR)	128-bit AES (ECB) / 256-bit AES (CTR) PRO: 256-bit AES (CBC or CTR)	128-bit AES (ECB) / 256-bit AES (CTR)
Secure Access	Yes	*No *PRO: Remote AT Command Password available*	Yes
Built-in Digi TrustFence[8][9] security	Yes	No	No
Signed OTA Firmware Updates	Yes	No	No

* *It should be noted that the modules listed above may support different features depending on their PRO variants or regional deployments.*

[3] *Digi XBee® 3 DigiMesh 2.4 (Rev. M): https://docs.digi.com/resources/documentation/digidocs/pdfs/90002277.pdf*

[4] *XBee® SX 868 (Rev. G): https://docs.digi.com/resources/documentation/digidocs/pdfs/90001538.pdf*

[5] *XBee®/XBee-PRO SX (Rev. N): https://docs.digi.com/resources/documentation/digidocs/pdfs/90001477.pdf*

[6] *Digi XBee® XR 900 (Rev. D): https://docs.digi.com//resources/documentation/digidocs/pdfs/90002474.pdf*

[7] *Digi XBee® XR 868 (Rev. E): https://docs.digi.com/resources/documentation/digidocs/pdfs/90002461.pdf*

[8] *Who Is Responsible for IoT Device Security? https://www.digi.com/blog/post/who-is-responsible-for-iot-device-security*

[9] *Digi TrustFence: https://www.digi.com/solutions/by-technology/trustfence*

3. Broad Security Best Practices for DigiMesh Environments

3.1. Firmware updates

Keeping DigiMesh modules on the latest firmware is one of the most fundamental security controls, updates fix known vulnerabilities and can introduce new security capabilities with no hardware change required.

Beyond reactive patching, organizations should establish internal processes to ensure modules are consistently maintained throughout their operational lifetime: keep a firmware version inventory for all deployed nodes, subscribe or monitor the Digi's security advisories[10] and release notes, and validate new firmware in a lab environment before network-wide rollout.

Digi provides firmware packages and release notes through its support hub (for example for the XBee XR 868 at Digi's hub[11]) with associated release notes[12]. Additionally, XCTU software automatically detects available firmware updates when a module is connected. For large fleets, it is recommended to centralize authenticated OTA firmware updates across nodes simultaneously.

3.2. Encryption Key Management and Rotation

Network encryption is only as strong as the key management practices surrounding it. A hardcoded, default or never-rotated key is a critical vulnerability, if any node is compromised, the entire network's historical traffic may be decryptable, and both network devices and the broader operational environment could be exposed to compromise.

Therefore, it is recommended to implement the following measures to establish a secure and resilient environment:

- **Strong, randomly generated keys:** Never use default or weak keys. Generate keys using a cryptographically secure random number generator and use the full available key length: 32 hex characters (128-bit) for legacy modules, or 64 hex characters (256-bit) for newer modules.
- **Key rotation**: Establish a rotation schedule proportional to the deployment's risk profile. High-security or critical infrastructure deployments should rotate at least quarterly and immediately upon any suspected compromise.
- **Secure provisioning**: Keys should be set once during commissioning via direct physical connection, never embedded in source code, configuration files, or transmitted over unencrypted channels.
- **Unique key per network segment:** Avoid key reuse across deployments. When the same hardware is used in different network or firmware configurations, ensure that cryptographic material is never shared between segments.

Finally, it is recommended to define and document the procedure for responding to a suspected key compromise: which nodes to replace, how to provision new keys, and how to verify network re-keying success.

[10] Digi Security Center: https://www.digi.com/resources/security

[11] Hub - Digi XBee XR 868: https://hub.digi.com/support/products/digi-xbee/digi-xbee-xr-868/

[12] XBee XR 868 Release Notes: https://hub.digi.com/dp/path=/support/asset/xbee-xr-868-release-notes/

3.3. Hardware & Firmware Security

While physical access to DigiMesh modules may appear to be a minor concern, especially when access to a critical deployment site is already considered a higher risk, these devices are often installed in remote or exposed environments. In such locations, physical compromise could lead to data breaches, data leakage, or even full network compromise.

Basic mitigation strategies include:

- **Secure boot**: On XBee 3 modules with Digi TrustFence, enable secure boot to prevent execution of unauthorized firmware. Only firmware signed with the provisioned key will boot, so unsigned images are rejected at startup.
- **Anti-tampering**: Security screws, tamper-evident seals, conformal PCB coating, and epoxy potting are practical deterrents that substantially increase the cost and detectability of physical attacks. For the highest-security deployments, fully potting the PCB can further enhance protection by eliminating opportunities for invasive side-channel attacks, including techniques such as physical probing and reverse engineering.
- **Intrusion detection**: Where possible and the environment requires it, deploy enclosures with intrusion detection switches connected to a monitored GPIO or external controller, configured to alert or trigger memory erasure upon unauthorised opening.
- **Disable unused interfaces**: Disable debug interfaces and unnecessary GPIO pins that could be used for side-channel attacks or unauthorized firmware extraction.

3.4. Disabling Unnecessary Features

Some DigiMesh modules provide additional features to enhance usability; however, any enabled feature not required by the application unnecessarily increases the attack surface. The following services should be reviewed and disabled where not needed:

- **MicroPython and file system:** Disable MicroPython if edge scripting is not required. An active interpreter allows arbitrary code upload and execution on the module. If MicroPython is not used, also disable the file system, as it serves no purpose without the interpreter and represents an additional exposure.
- **Bluetooth**: Disable the BLE interface if it is not required for commissioning or device management. Bluetooth introduces a short-range wireless attack surface that is independent of the DigiMesh RF stack and often overlooked in network security assessments.
- **Device discovery and broadcast:** Disable or restrict network discovery and broadcast features to prevent passive enumeration of network topology, node addresses, and device identifiers by unauthorized listeners.
- **Remote management**: Remote management interfaces should be disabled by default and only enabled through deliberate, authenticated configuration.
- **Over-The-Air updates**: Consider disabling OTA update mechanisms in deployments where firmware updates are performed exclusively through controlled local processes.

3.5. Monitoring

A secure network is a monitored network. DigiMesh modules expose rich diagnostic capabilities that, when integrated into a monitoring pipeline, can provide early warning of network anomalies:

- **RSSI monitoring**: Unexpected changes in Received Signal Strength Indicator (RSSI) can indicate node movement, RF jamming attempts, or rogue nodes in proximity.
- **Transmission error rate tracking**: A sudden increase in transmission errors (ER or TR AT commands) can indicate interference, a degraded node, or a jamming attack.
- **Unexpected topology changes**: Log network discovery results regularly. New, unexpected node addresses appearing in the network should trigger investigation.

It should be noted that as deployments grow, ad hoc monitoring becomes increasingly unmanageable. Centralizing telemetry in a dedicated system, whether a private solution, a SCADA platform, or an IoT operations tool, enables events to be correlated across all nodes, allows threshold alerts to be applied consistently, and supports the maintenance of historical baselines for anomaly detection.

4. XBee Module Hardening – Best Security Practices

4.1. AES Encryption

All DigiMesh modules support AES encryption, enabled via **EE=1** and configured with the **KY** AT command. However, not all encryption modes offer the same level of protection and choosing the right mode matters. While CBC and ECB are common, they leave specific backdoors open that modern networks cannot afford:

- **CBC (Cipher Block Chaining):** CBC lacks the nonce/counter that CTR mode uses to make each encryption unique. Without it, a captured encrypted frame can be re-transmitted later and accepted as legitimate because the receiver has no way to detect it has already been seen. Additionally, if Initialization Vectors (IV) are reused or predictable, an attacker can identify when the same plaintext has been sent, partially exposing message structure.

- **ECB (Electronic Codebook):** ECB encrypts each block independently, so identical plaintext blocks always produce identical ciphertext. An attacker passively observing traffic can infer data structure and communication patterns over time without ever breaking the key. It also provides no integrity protection, since individual ciphertext blocks can be modified without the receiver detecting tampering.

Therefore, for all modern mesh deployments, **AES-256 CTR** is the recommended choice. It provides the highest level of confidentiality while natively supporting replay attack resistance. By using a unique, incrementing counter (a "nonce") for every packet, it ensures that even if an attacker captures a valid encrypted command and tries to broadcast it again later, the network will reject it as a duplicate.

AT Command Configuration	Description
`AT EE 1`	`Enable encryption`
`AT KY [Your_64_Hex_Characters]`	`Set your 256-bit Key`
`AT WR`	`Commit changes`

Note that when **C8** (Compatibility Options) bit 2 is cleared, encryption and decryption use the full 256-bit KY value (all 64 ASCII characters) with CTR mode. If **C8** bit 2 is set, AES security may be downgraded either by reducing the key length to 128 bits or by changing the encryption mode (ECB or CBC), depending on the XBee module.

4.2. Secure Sessions

Modules such as the XBee XR and XBee 3 support Secure Sessions, a session-based authentication mechanism that provides protection beyond network-level AES encryption. Instead of transmitting a password with each frame, Secure Sessions use a Secure Remote Password (SRP) exchange.

Secure Sessions rely on a salt and verifier derived from a password, ensuring the password itself is never transmitted over the air. This mechanism restricts sensitive operations, such as remote AT commands and data delivery, to nodes with an authenticated session.

Note that Digi recommends using the XCTU software, instead of setting the values directly via AT commands, to set a password, which will then automatically generate the salt and verifier parameters.

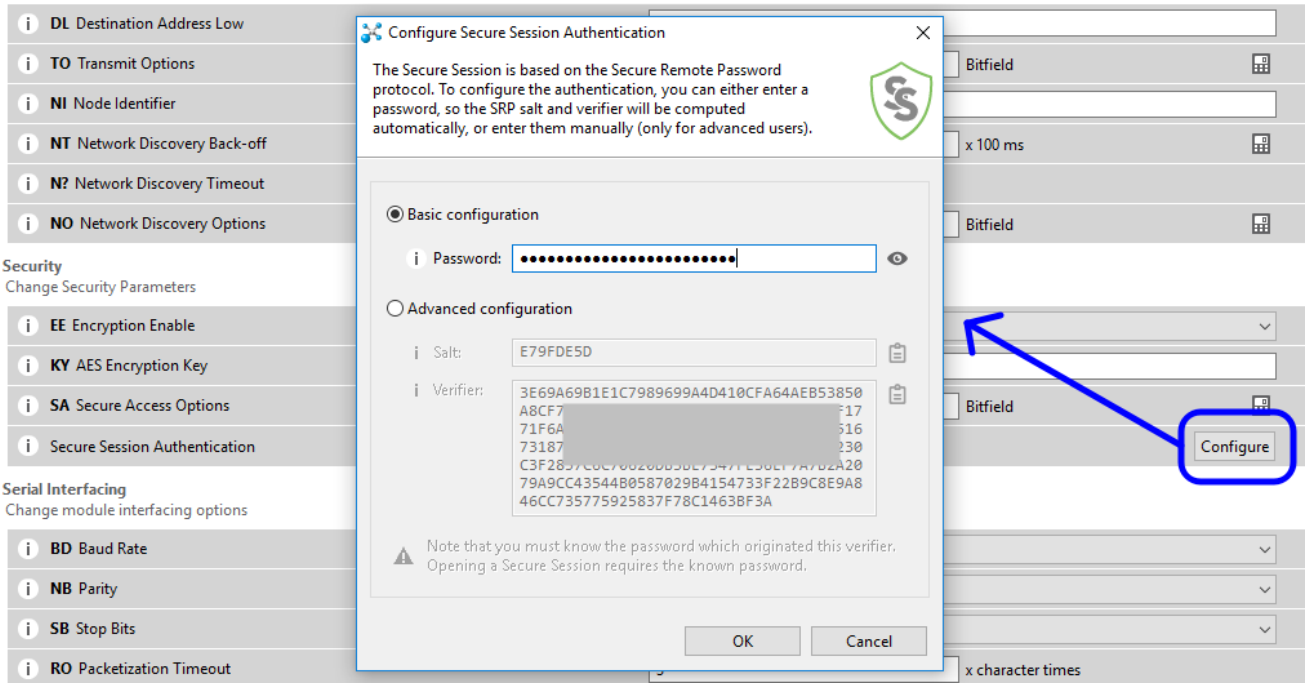

Figure 2: Configuring Secure Session authentication in Digi XCTU

Secure Sessions configuration via AT commands:

AT Command Configuration	Description
`AT *S [salt_value]`	Set the SRP salt
`AT *V [verifier_part_1]`	Set verifier chunk 1
`AT *W [verifier_part_2]`	Set verifier chunk 2
`AT *X [verifier_part_3]`	Set verifier chunk 3
`AT *Y [verifier_part_4]`	Set verifier chunk 4
`AT SA [bitmask]`	Enable Secure Access restrictions: - Remote AT Commands - Serial Data
`AT EE 1`	Ensure encryption is enabled
`AT WR`	Commit changes

4.3. Secure Remote AT Commands

Other modules such as XBee/XBee-PRO SX RF do not implement secure sessions but implement the **KZ** AT command which allows setting a password to inhibit transmission and reception of remote AT commands without including the password in the request. When you set **KZ**, it disables “0x17 - Remote AT Command” frames and “0x18 - Secure Remote AT Command” frames must be used instead.

AT Command Configuration	Description
`AT EE 1`	Ensure encryption is enabled
`AT KZ [password]`	Set the remote AT command password (ASCII string)
`AT WR`	Commit changes
`AT FR`	Reset the module

4.4. Secure OTA Updates

Over-The-Air (OTA) firmware updates are operationally critical for maintaining the security and reliability of deployed networks. However, if not properly secured, they can also introduce a significant attack vector, as an unprotected OTA mechanism could allow an attacker to deploy malicious firmware to field devices.

The XBee 3 is currently the only DigiMesh module that supports cryptographically authenticated OTA updates. Through the **FK** (File System Public Key) AT command, a public key can be securely stored on the module and subsequently used to verify the digital signature of incoming firmware, ensuring that only trusted and authorized images are accepted.

AT Command Configuration	Description
`AT EE 1`	`Ensure encryption is enabled`
`AT FK <public_key>`	`Set the public key (65-byte ECDSA)`
`AT WR`	`Commit changes`

In addition, TrustFence enforces anti-rollback protection, preventing devices from being downgraded to older, potentially vulnerable firmware versions. Together, these mechanisms provide the strongest OTA security posture available within the XBee portfolio.

This feature is not available in other DigiMesh modules such as XBee SX or XBee XR. These devices support unsigned OTA updates through the GPM interface using FIRMWARE_VERIFY (0x05) and FIRMWARE_VERIFY_AND_INSTALL (0x06) over “0x11 - Explicit API” frames, where image integrity is checked prior to installation.

However, there is no image signing, no source authentication, and no anti-rollback protection at the GPM protocol level. Consequently, firmware is not cryptographically validated, and a node with network access and knowledge of the GPM frame format could potentially push arbitrary firmware to an unprotected target. In this setup, confidentiality relies solely on network-level AES encryption.

The image below shows an example of how to verify a firmware image previously loaded into the GPM on a target device with serial number “0013A200407402AC”:

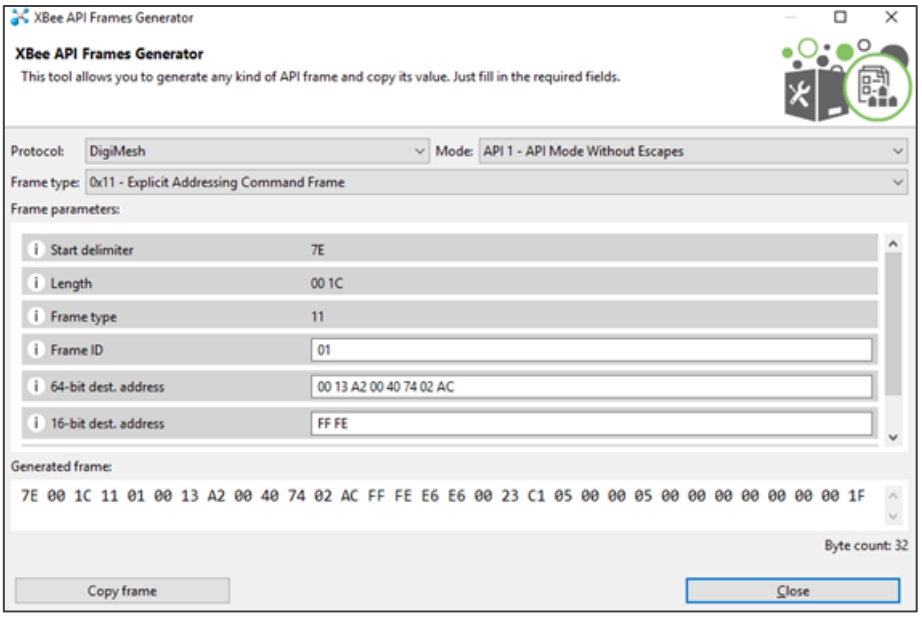

Figure 3: XBee API frames generator

4.5. Changing Defaults

Besides the recommendations pointed out in this document, small configuration changes applied consistently across all nodes can meaningfully reduce the network's exposure to passive reconnaissance and opportunistic attacks.

- **Network ID (ID) and Operating Channel (CH):** DigiMesh nodes communicate only with devices sharing the same Network ID and channel. Using non-default values for these parameters adds a simple but effective layer of access control, reducing unintended cross-network communication and making network discovery attacks more difficult.

AT Command Configuration	Description
`AT ID <value>`	`Set the network ID (0x0 – 0x7fff)`
`AT CH <value>`	`Set the operating channel (0xb – 0x1a)`
`AT WR`	`Commit changes`

** Note that CH AT command is only available in the XBee 3 module*

- **Node Identifier (NI):** Use a naming convention that does not reveal device function to a passive observer. Use opaque identifiers such as “device_57F2” over descriptive names like “critical_turbine_sensor”.

AT Command Configuration	Description
`AT NI <name>`	`Set the node identifier`
`AT WR`	`Commit changes`

- **Command mode lockdown (CC, CT and GT):** Command mode provides a configuration interface that, if accessed through the physical serial connection, could allow an attacker to reconfigure an XBee module, including disabling encryption, modifying network parameters, or redirecting communications. Properly restricting and hardening access to command mode significantly increases resistance to physical attacks and unauthorized local reconfiguration.

AT Command Configuration	Description
`AT CC <string>`	`Change the default “+++” entry sequence`
`AT CT <value>`	`Sets the command mode timeout`
`AT GT <value>`	`Set the required period of silence before and after the command sequence`
`AT WR`	`Commit changes`

- **Minimum transmit power (PL):** Set transmit power to the minimum level required to maintain reliable communication. Unnecessary RF range increases the network's physical footprint and the number of potential eavesdroppers or attackers within reception distance.

AT Command Configuration	Description
`AT PL <value>`	`Sets the power level (0x0 – 0x4)`
`AT WR`	`Commit changes`

- **Do not use the Network Discovery (ND) AT command:** The **ND** command broadcasts node addresses and topology information network-wide. Restrict it to the commissioning phase and do not use it in production to prevent passive enumeration.

- **Disable unused I/O sampling and flow control:** Every active I/O line and reporting feature that the application does not use is an unnecessary exposure. GPIO pins left in default states can be manipulated by an attacker with physical access and leak physical state information.

AT Command Configuration	Description
`AT D6 0`	`Disable the DIO6/RTS pin`
`AT D7 0`	`Disable the DIO7/CTS pin`
`AT IC 0`	`Disable I/O change detection reporting (if not required)`
`AT IR 0`	`Disable I/O sampling rate`
`AT WR`	`Commit changes`

- **Disable Features (DM):** The **DM** command defines a bit-field mask that allows specific DigiMesh protocol features to be disabled at the firmware level. These features include FOTA updates, Trace Route, NACK responses, SRP authentication (both client- and server-side), and aggregator updates.

AT Command Configuration	Description
`AT DM <value>`	`Disable the selected function (0x0 – 0x1F)`
`AT WR`	`Commit changes`

- **API mode:** Using **(AP=1 or AP=2)** instead of transparent (AT) mode provides significantly stronger control, visibility, and resilience against misuse.

5. Conclusions

Selecting the appropriate wireless mesh protocol is never a trivial decision. Each deployment presents unique requirements in terms of topology, power constraints, data throughput, scalability, and security posture. When DigiMesh is the appropriate choice, implementing security correctly from the outset becomes essential.

DigiMesh networks are commonly deployed in demanding environments such as agriculture, industrial automation, smart city infrastructure, and other critical systems, where the consequences of a security breach extend far beyond data loss. In these contexts, a compromised node is not merely an IT issue; it can disrupt operations, introduce safety risks, and potentially trigger cascading failures across interconnected infrastructure.

This document has provided a comprehensive reference addressing the full spectrum of potential attack surfaces, from network-level protections such as encryption key management, firmware lifecycle control, and physical hardware security to protocol-level hardening measures including Secure Sessions, remote AT command protection, and secure OTA update mechanisms.

Adhering to the best practices outlined here establishes the foundation for DigiMesh deployments that are not only functional but secure, resilient, and prepared to withstand both current and evolving threats. In critical environments, security is as much an operational discipline as it is a technical requirement, and it is always more effective when designed in from the beginning rather than retrofitted after deployment.

Finally, we must not overlook system integrators, asset owners, and operators. Even if they are not directly responsible for configuring or deploying DigiMesh networks, it is essential that they understand whether the system has been properly secured. The technical best practices in this document translate into a set of basic but critical questions that should be asked of any party responsible for deploying or maintaining a DigiMesh environment:

- Are modules running the latest firmware?
- Is encryption active and are keys properly managed?
- Are nodes physically protected against unauthorised access?
- Is remote configuration access restricted or protected by secure sessions?
- Are OTA firmware updates authenticated and controlled?
- Have factory defaults been changed to minimise the exposure of potential attacks?
- Is the network actively monitored to detect anomalies or attacks?
- Are security procedures documented and regularly reviewed?

6. References and Further Reading

Digi International - Digi TrustFence Device Security Framework.
https://www.digi.com/solutions/by-technology/trustfence

Digi XBee® 3 DigiMesh 2.4 (Rev. M)
https://docs.digi.com/resources/documentation/digidocs/pdfs/90002277.pdf

XBee® SX 868 (Rev. G)
https://docs.digi.com/resources/documentation/digidocs/pdfs/90001538.pdf

XBee®/XBee-PRO SX (Rev. N)
https://docs.digi.com/resources/documentation/digidocs/pdfs/90001477.pdf

Digi XBee® XR 900 (Rev. D)
https://docs.digi.com//resources/documentation/digidocs/pdfs/90002474.pdf

Digi XBee® XR 868 (Rev. E)
https://docs.digi.com/resources/documentation/digidocs/pdfs/90002461.pdf

Who Is Responsible for IoT Device Security?
https://www.digi.com/blog/post/who-is-responsible-for-iot-device-security

Digi TrustFence
https://www.digi.com/solutions/by-technology/trustfence

7. About Us

Daniel Romero

Daniel is currently Senior Principal Security Consultant and European Research Lead at NCC Group. He has more than sixteen years of experience as security consultant and has worked at some important companies delivering and managing complex, high-technology security projects.

During his career, he has worked with several high-profile clients and has successfully delivered numerous security and research-driven assessments. These have included embedded systems and product security assessments (including firmware reverse engineering, fuzzing, hardware security review, protocol and cryptographic analysis, and exploitation), compiled applications, low-level code reviews, and device assessments involving wireless technologies and protocols such as LoRaWAN, mesh networks, BLE, and other RF technologies.

In addition, he has maintained ongoing involvement in vulnerability research and hardware hacking, alongside web and mobile application security assessments and internal and external network security engagements.

NCC Group

NCC Group is a people-powered, tech-enabled global cyber resilience and software escrow business.

Driven by a collective purpose to create a more secure digital future, over 2,000 colleagues across Europe, North America, and Asia Pacific harness their collective insight, intelligence, and innovation to deliver cyber resilience to clients across the public and private sector.

With decades of experience and a rich heritage, NCC Group is committed to developing sustainable solutions that continue to meet clients' current and future cyber security challenges.

Follow NCC Group on LinkedIn and at https://www.nccgroup.com/