Thread Security Architecture and Features

Authors: Esko Dijk, Alban Notin, Lance Looper, Levy Toredjo

This Thread Technical white paper is provided for reference purposes only. The full technical specification is available publicly. To gain access, please follow this link: https://www.threadgroup.org/ThreadSpec.

If there are questions or comments on this technical paper, please send them to help@threadgroup.org.

Table of Contents

1. Introduction
2. What is Thread?
3. Thread's Layered Security Architecture
4. Secure Network Joining and Device Authentication
5. Per-Link Encryption and Authentication
6. Secure Routing and Relay
7. Device and Network Lifetime Management
8. Strong Key Management
9. Protection Against Denial of Service (DoS) Attacks
10. Conclusion

1. Introduction

Connected devices are now an integral part of our lives. At home, at the office, in our cars, and virtually anywhere else we find ourselves, connectivity is ubiquitous. As the number of devices grows, so does our dependence on the security of these networks. For OEMs, security has to be built into the way devices join a network, authenticate, communicate, route traffic, manage keys, and remain protected over time. A weak connectivity layer can expose devices to unauthorized access, eavesdropping, replay attacks, denial-of-service attempts, or long-term credential risk.

Thread is positioned to play a vital role in enabling secure, efficient, and interoperable IoT communication.

In this white paper, we provide an overview of the Thread wireless connectivity standard, its security features, and the design principles that make it secure. It explains how Thread's architecture and security mechanisms can be used to help protect IoT devices and systems throughout their lifecycle, from commissioning and daily operation to reconfiguration and decommissioning.

This paper is intended for industry decision makers involved in connectivity and IoT networking strategy. It is not intended to be a tutorial or a how-to manual for system integrators, installers, or developers.

2. What is Thread?

Thread is a low-power, wireless mesh networking protocol developed by the Thread Group, an industry consortium that includes major players such as Amazon, Apple, Google, Qualcomm, Samsung, and Siemens. It is designed to provide reliable, secure, and scalable RF connectivity for smart devices in homes and buildings. Thread is designed with security as a core feature, helping devices and communications remain protected against security threats.

Thread operates on the IEEE 802.15.4 radio standard, which specifies the physical and medium access control layers for wireless personal area networks. By using a mesh topology, Thread allows devices to communicate with each other over a multi-hop network, improving range, reliability, and robustness.

Key features of Thread include:

- **Low power consumption:** Optimized for battery-powered devices.
- **Mesh networking:** Mains-powered devices can relay data on behalf of other devices, to extend the effective range and reliability of the network. Such devices are called Thread Mesh Extenders.
- **Self-healing:** If a Thread device goes offline, the mesh network automatically reroutes data through other available network paths.

- **IPv6 support:** Thread uses the Internet Protocol version 6 (IPv6), enabling a vast address space while simplifying device addressing, as detailed further below.
- **Interoperability:** Thread is designed to work with existing Internet Protocol (IP) based networks, ensuring easy integration with other networking standards.
- **Designed for reliable and secure data transport:** Thread incorporates AES-128 encryption for secure communications, with robust key management, and mechanisms like per-link encryption and secure device commissioning to ensure data integrity and confidentiality at all times.

Thread's choice for natively supporting IPv6 enables it to leverage the following essential IPv6 features:

- **Vast address space:** IPv6 vastly increases the address space compared to Internet Protocol version 4 (IPv4), enabling the deployment of a large number of devices in IoT ecosystems without address exhaustion or address conflicts.
- **Seamless interoperability:** IPv6 facilitates direct communication across different network technologies and device types, enabling seamless integration of Thread-enabled devices with other (non-Thread) IP-based devices.
- **Simplified address management:** IPv6's hierarchical addressing simplifies device addressing and network management, which is vital for scalability in dense IoT deployments.
- **IPv6 addressability and auditability:** Each Thread device has a unique IPv6 address, allowing event messages to be routed per device to a central log server. This makes it easier to implement per-device audit logging when required by security policies or by higher-level frameworks.

A key benefit of natively supporting IPv6 is taking advantage of all the robust (security) protocols, products and infrastructure that have been widely deployed to support the world's most critical applications. Such protocols are typically defined by the Internet Engineering Task Force (IETF).

The figure below shows how a Thread mesh network (center) can be seamlessly embedded in an existing building infrastructure where multiple IP and non-IP building-automation protocols already operate side by side. Thread integrates with a corporate backbone IPv6 network with a building management system (BMS). A key component to enable this is the Thread Border Router: a "gatekeeper" that links a Thread mesh network to other IP networks such as Ethernet or Wi-Fi. The Thread Border Router acts as an IPv6 router, IPv6/IPv4 translator, and a standardized security boundary that connects the Thread mesh network to adjacent IP networks (such as Ethernet or Wi-Fi). Because Thread is native IPv6, the Border Router forwards standard IPv6 packets without application-layer translation. Furthermore, the Thread Specification standardizes mandatory packet filtering rules—including strict ingress filtering (forwarding only traffic destined for configured on-mesh IPv6 prefixes), mesh-local address confinement, and source-address anti-spoofing—shielding the low-power mesh from unauthorized external traffic.

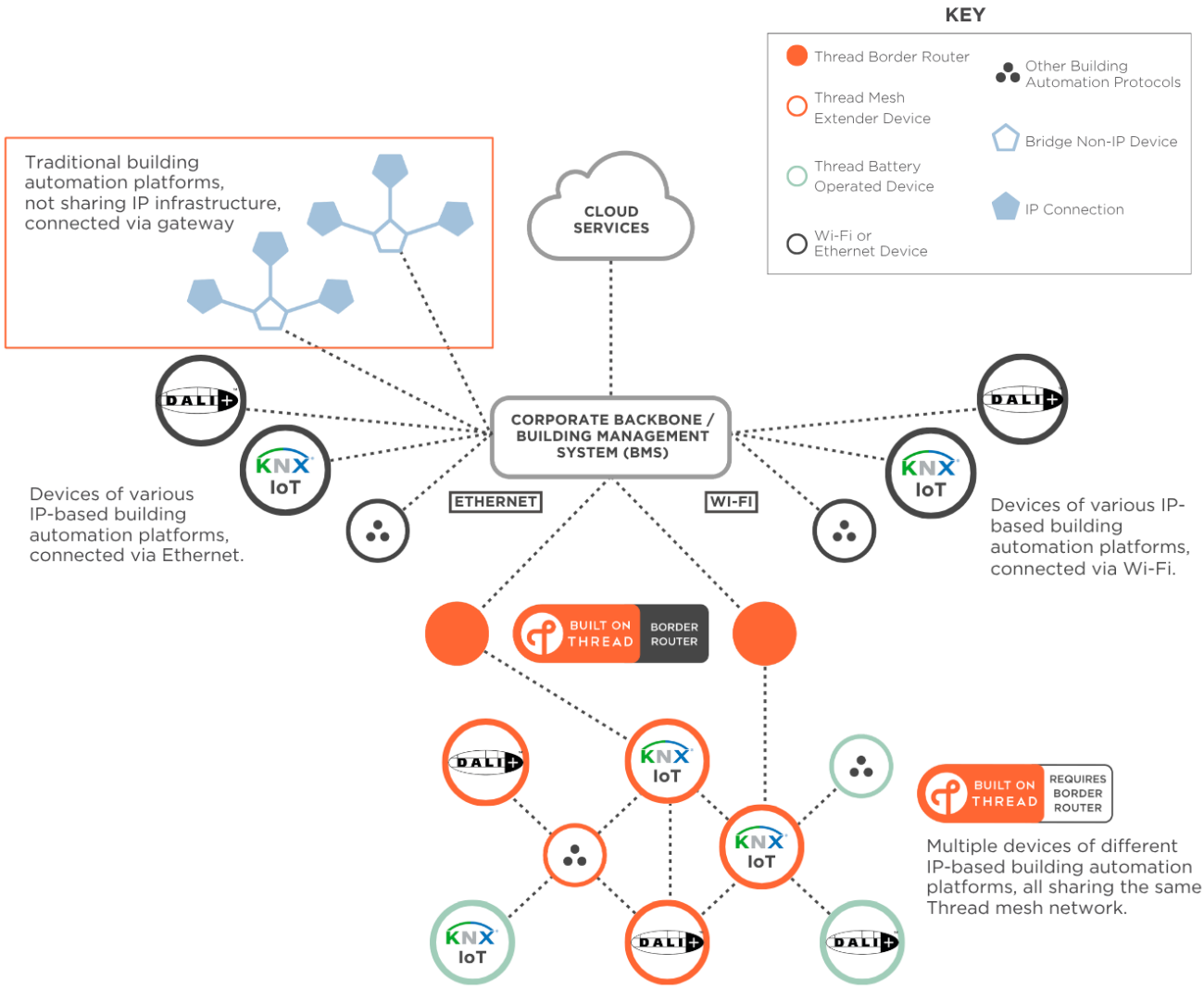

Fig. 1. Example of a Smart Building Network Topology Integrating a Thread Network (center), connected via Internet Protocol to a Building Management System (BMS)

3. Thread's Layered Security Architecture

Thread employs a layered security architecture. It implements 6LoWPAN to efficiently support IPv6 addressing on top of IEEE 802.15.4 radio technology. **Secure commissioning with authentication** complements these measures to support robust data and device protection.

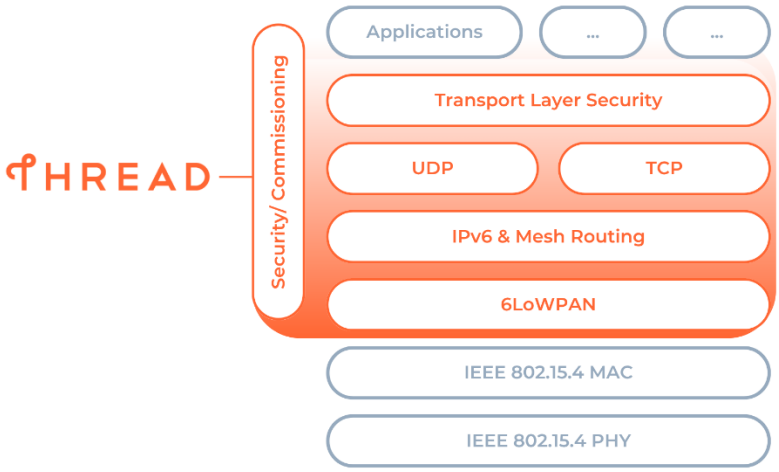

Fig. 2. The stack diagram above shows Thread's layered security architecture

During normal network operation, Thread always provides data-link security. This includes AES-CCM* (128-bit key, 32-bit MIC) at the MAC layer on every hop, ensuring both confidentiality and message integrity. Because Thread is natively IPv6-based, developers can implement additional security at the transport layer or application layer when end-to-end encryption is required.

This means standard IPv6-based security protocols such as TLS, DTLS, or OSCORE can be used for true **end-to-end protection and authentication** when needed. Commercially available Thread components typically already include support for one or more of these protocols.

This layered approach enables flexible security strategies. Vendors can tailor encryption and authentication to their specific use cases while relying on Thread's link-layer security foundation.

4. Secure Network Joining and Device Authentication

When a new device joins a Thread network, it must go through an authentication process. Thread is designed so that only authorized devices can join the network, using a combination of cryptographic techniques and strict security protocols.

Role of the Commissioner

- The **Commissioner** is the trusted entity that decides which Thread devices can be added onto the network. A mobile device such as a smartphone or tablet can operate as a Commissioner, even when it has no Thread radio itself.
- It ensures that only authorized devices can join, preventing any unauthorized access. The Commissioner can optionally audit the new device using for example online methods of device attestation, before granting access. Attestation is a cryptographic method to validate that a new device is a genuine, certified device that has not been tampered with. Other methods for performing this audit are also enabled.

- Typically, it authorizes the new Thread device based on a passphrase or a pre-provisioned key. Authentication using X509 certificates is also available for specific implementations targeting commercial applications.

Secure Device Enrollment

- **Passphrase-based approach:** Standard Thread commissioning establishes a secure DTLS session authenticated via Elliptic Curve J-PAKE (EC-JPAKE) using the Joining Device Credential (passphrase or QR code). EC-JPAKE is a zero-knowledge password-authenticated key exchange (PAKE) protocol that ensures passphrases are never transmitted over the air and protects against offline dictionary attacks.
- **Commissioning Traffic Relay:** Uncommissioned devices communicate with the Commissioner via stateless CoAP joiner relays through on-mesh Thread Mesh Extenders (over UDP port 1000), securely isolating unauthenticated traffic until provisioning is complete.
- **Certificate-based approach:** As stated above, in some implementations, a certificate-based model may be used to authenticate both the Commissioner and the new device, in order to meet stricter security requirements in certain markets.
- **Pre-shared key for the Commissioner (PSKc)**: A PSKc is a derived key based on the Commissioning Credential password that lets a trusted entity establish a commissioner session and become the Commissioner, with proper authentication.

Device Authentication and Secure Session Establishment

- Datagram Transport Layer Security (DTLS): Any communication between a new Thread device and a Commissioner is protected by DTLS, a standard transport-layer security protocol defined by the Internet Engineering Task Force (IETF). This protocol enables various forms of device authentication, including any one of the secure device enrollment methods listed above.
- Thread builds on Elliptic Curve Cryptography (ECC) for trusted authentication, as part of DTLS secure session establishment.

New Thread Device Commissioning/Provisioning Process

- To start the commissioning and provisioning process for a new Thread device, a Commissioner first enables access to new devices in the Thread network. This may be all new devices, or designated new devices.
- Next, the new Thread device discovers that the network is enabled and sends a request to join the network.
- The Commissioner receives the request via the network and validates the new device's credentials. It can optionally audit the new device if this is supported for the type of device. The process to audit a new device may be required by an IoT ecosystem standard, such as Matter, to ensure the device is authentic and certified.
- Once the new device is authenticated and approved, it is provisioned with network-specific credentials (e.g., the Thread Network Key) to enable secure communication with other Thread devices in the network.

5. Per-Link Encryption and Authentication

Thread employs mandatory standard AES-128 encryption for all communications on the network, ensuring that messages exchanged between devices are always encrypted and

protected from eavesdropping. This per-link encryption together with the message authentication code (MAC) inside each message, protects both the integrity and confidentiality of data transmitted over the Thread network.

Per-Link Security Features

- **Data Integrity & Confidentiality:** AES-CCM* ensures data cannot be altered or read without authorization.
- **Replay Attack Protection:** Thread devices maintain neighbor frame-counter tables to strictly reject duplicate, out-of-order, or decremented frame counters, preventing packet replay attacks.

Per-Link Versus End-to-End Security

- Each Thread Mesh Extender forwarding multi-hop traffic decrypts and authenticates incoming link-layer frames to inspect standard IPv6 network routing headers for next-hop forwarding, then re-encrypts the frame for transmission to the next neighbor.
- When higher-layer end-to-end security (e.g., TLS, DTLS, Matter CASE, OSCORE) is applied, the application payload remains completely encrypted and opaque to intermediate routing nodes.
- End-to-End Encryption (E2EE) requires an additional protocol such as TLS, DTLS, or OSCORE at the transport or application layer. These protocols ensure data remains encrypted from the originating device to the final recipient, even when the originator or recipient is not a Thread device but an external IPv6/IPv4 device.

6. Secure Routing and Relay

Thread's mesh network is inherently resilient and self-healing. As data passes hop-by-hop over radio links, mains-powered devices act as Thread Mesh Extenders, relaying messages to their destinations.

Thread implements robust security mechanisms to protect communications at each hop while maintaining network reliability:

- **Contributes to Mesh Network Resilience**: If a Thread Mesh Extender is switched off or fails, the network dynamically reroutes traffic.
- **Supports End-to-End Protected Data:** True end-to-end encryption (E2EE) and confidentiality can be accomplished using an optional end-to-end security protocol of choice. End-to-end protected data remains protected from eavesdropping or tampering while the Thread Mesh Extender performs routing tasks. Thanks to the layered IP architecture, the transported data can remain end-to-end protected even when one of the communication endpoints is located outside the Thread network, for example on the internet.

Thread's robust per-link encryption, dynamic routing, and resilience against radio interference make it an ideal choice for secure and reliable IoT communication. For heightened data confidentiality and protection against advanced threats, deploying a higher-layer end-to-end encryption protocol is recommended.

7. Device and Network Lifetime Management

Thread incorporates several mechanisms to prevent devices from becoming vulnerable over time:

- **Renewal of keys:** Periodic renewal of link encryption keys reduces the risk of long-term cryptographic exposure.
- **Secure decommissioning:** Since Thread 1.4, the optional Thread Commissioning over Authenticated TLS (TCAT) feature allows a Commissioner to securely wipe network credentials from a device when it is removed from a Thread network.
- **Network-wide key change:** The function of network-wide key change, which can be used, for example, after the removal of a potentially compromised device, ensures that any removed devices cannot exploit their old credentials for network access.

This combined approach of mechanisms maintains network integrity and resilience even in challenging security scenarios that involve long lifetimes.

8. Strong Key Management

Thread employs key management protocols to ensure that cryptographic link keys are handled securely and rotated regularly, enhancing network security and resilience. For demanding use cases, the default key management can be customized using configuration and/or vendor-specific features:

- **Periodic Key Sequence Rotation:** Thread networks support automatic key rotation (by default every 28 days, configurable by an administrator) by incrementing the Key Sequence Counter to derive fresh link keys. This mitigates cryptographic exposure and prevents frame counter exhaustion during long network uptimes.
- **Network-Wide Key Change:** If a device is permanently decommissioned or compromised, the network can perform a Network Key update to revoke network access and ensure removed devices cannot participate in future communications.
- **Secure key storage:** Thread provides flexibility for module/system-on-chip (SoC) vendors to implement tailored solutions for secure key storage. While the Thread certification process does not mandate specific storage methods, it allows vendors to leverage advanced technologies such as hardware security modules (HSMs) or secure memory solutions, enabling them to exceed baseline security standards and differentiate their products in the market.

Best practices in **secure key storage** and device security in connected products ensure that Thread's built-in key management remains effective.

9. Protection Against Denial-of-Service Attacks

To protect against Denial of Service (DoS) attacks, Thread implements several countermeasures designed to enhance the resilience of the network and mitigate malicious activity.

- **Rate limiting:** Thread networks impose rate limits on control plane traffic, joiner requests, and diagnostic messaging to mitigate flood-based denial-of-service attacks and minimize the impact of excessive traffic on constrained mesh devices.
- **Thread-specific network monitoring:** While Thread already supports per-device diagnostics information for network performance and status, it provides flexibility for vendors to develop tailored attack detection and mitigation systems on top. This allows vendors to integrate custom solutions that meet the specific security needs of their applications, enhancing overall network defense.
- **Standard IP network monitoring:** Because every Thread packet is native IPv6, the same network traffic monitoring, analytics, and anomaly-detection already proven on wired and Wi-Fi IP networks can be applied at or behind the Thread Border Router for real-time threat monitoring, detection, and automated mitigation.
- **Path selection:** Thread's mesh routing algorithms prioritize stable and best-quality paths, while dynamically adapting to changing network conditions. This self-healing capability ensures robust communication and makes it more difficult for potential attackers to disrupt network traffic through localized RF interference or localized jamming, as traffic dynamically reroutes around affected nodes.
- **Traffic prioritization:** Thread prioritizes critical network management traffic over regular data traffic to ensure network integrity even under high-load conditions.

Thread's built-in mechanisms provide a strong foundation for mitigating the effects of DoS attacks, but achieving the most comprehensive protection requires device vendors to integrate additional functionality, such as attack detection and response systems. Such functionality will typically reside on a more powerful network device, such as a Thread Border Router.

10. Conclusion

The Thread wireless connectivity standard offers a complete package of advanced security features, including a layered security architecture, link-layer encryption, secure commissioning, key rotation, and compatibility with standard IP-based internet security protocols like TLS and DTLS. This comprehensive set of features makes Thread a future-ready secure foundation for a wide range of IoT devices that require local connectivity as well as connectivity to the wider internet.

Thread's security design and alignment with widely adopted security best practices position it as an ideal solution for secure, scalable and future-proof IoT networks.