Secure and Scalable Provisioning for Embedded Systems: A Comparative Study of Techniques, Trust Models, and Future Trends

Nikheel Vishwas Savant, *Senior Software Engineer, Reality Labs, Meta*

Abstract— **Provisioning is a foundational step in the deployment of embedded systems, enabling secure, authenticated, and scalable onboarding of devices into trusted ecosystems. While extensive research exists on individual protocols and cryptographic techniques, current literature lacks a unified perspective that bridges standards, algorithms, and real-world implementations. This paper presents a structured survey and comparative analysis of device provisioning strategies tailored for resource-constrained embedded systems. It categorizes common methods—manual, QR-code-based, out-of-band (OOB), and zero-touch—and analyzes the underlying key exchange schemes such as ECDH [4], PSK [9], and certificate-based provisioning [13]. Furthermore, it introduces a trust chaining model for scalable delegation, synthesizing practices from Matter [1], Thread Mesh [6], and industrial IoT architectures. By evaluating the trade-offs in security, scalability, user complexity, and hardware dependencies, the paper provides actionable guidance for system architects and researchers. It concludes with future directions including post-quantum security [7], cross-vendor onboarding standards like FIDO Device Onboarding (FDO) [15], and AI-assisted trust verification.**

Index Terms— **Provisioning, Embedded Systems, Secure Onboarding, Zero-Touch Provisioning (ZTP), Out-of-Band (OOB) Pairing, QR Code Provisioning, Trust Chain, Key Exchange, Elliptic Curve Diffie-Hellman (ECDH), Pre-Shared Key (PSK), Public Key Infrastructure (PKI), Bluetooth Low Energy (BLE), Matter, Thread, Trusted Platform Module (TPM), Post-Quantum Cryptography, FIDO Device Onboarding (FDO), IETF SUIT.**

I. INTRODUCTION

As embedded systems become increasingly pervasive across the Internet of Things (IoT), automotive, industrial automation, and healthcare domains, the challenge of securely and efficiently provisioning these devices has taken center stage. Provisioning refers to the process of securely enrolling a device onto a network, configuring its identity, credentials, and operating parameters [8]. A key subset of this is **secure onboarding**, which ensures that a device is authenticated, its software integrity is verified, and communication with the ecosystem is encrypted from the outset.

This process is particularly challenging for embedded platforms, which often operate under severe constraints in terms of memory, compute power, energy availability, and secure hardware. As a result, provisioning mechanisms must balance **cryptographic strength with operational efficiency**.

While several standards and frameworks offer solutions for identity and provisioning—such as IEEE 802.1AR for secure device identity [8], the Matter protocol for smart home ecosystems [1], and FIDO Device Onboarding (FDO) for cross-vendor automation [15]—they are often fragmented across different domains and implementations. This fragmentation makes it difficult for system designers to identify the most appropriate strategy for their embedded deployment.

This paper addresses this gap by providing a **comprehensive survey and comparative analysis** of provisioning methods used in embedded systems. It categorizes provisioning strategies into manual, QR-code-based, out-of-band (OOB), and zero-touch provisioning (ZTP), and analyzes their trade-offs in terms of security, scalability, user interaction, and hardware requirements. In addition, the paper explores **trust chaining** as a scalable delegation model, drawing inspiration from Thread Mesh [6] and certificate-based trust systems.

We also examine commonly used **key exchange algorithms**—including Elliptic Curve Diffie-Hellman (ECDH) [4], Pre-Shared Keys (PSK) [9], and certificate-based schemes using Public Key Infrastructure (PKI) [13]—and discuss their suitability for constrained devices.

In doing so, this paper aims to serve as a reference for engineers, system architects, and researchers seeking to design **scalable, secure, and interoperable provisioning pipelines** for embedded ecosystems.

II. METHODOLOGY OF LITERATURE SURVEY

To ensure a comprehensive and balanced survey of device provisioning techniques, this paper followed a structured literature review process across academic, industrial, and standardization domains. The goal was to identify provisioning mechanisms specifically relevant to **resource-constrained embedded systems**, evaluate their trade-offs, and contextualize them within emerging trust and security models.

A. Literature Sources:

Sources were selected from:

- **Peer-reviewed academic publications** indexed in IEEE Xplore, ACM Digital Library, and SpringerLink.
- **Industry whitepapers and specifications** from organizations such as the Connectivity Standards Alliance (CSA), Bluetooth SIG, FIDO Alliance, and IETF.
- **Technical standards and protocols**, including IEEE 802.1AR [8], Matter [1], TLS [13], HKDF [5], and FDO [15].

B. Search Criteria and Keywords

Key search terms included:

- *Device provisioning*, *secure onboarding*, *embedded systems security*, *BLE provisioning*, *ZTP embedded*, *trust chaining IoT*, *TPM key exchange*, and *post-quantum device identity*.

The literature selection emphasized works that:

1. Described **provisioning flows** or key exchange schemes applicable to **constrained environments**.
2. Focused on **security, scalability, or interoperability** challenges in embedded systems.
3. Included **real-world adoption** or integration with platforms such as AWS IoT [2], Azure DPS [3], or Apple HomeKit [10].

C. Evaluation Framework

Each provisioning method or algorithm was evaluated using the following criteria:

- **Security**: Resistance to spoofing, MITM, replay, and downgrade attacks.
- **Scalability**: Suitability for factory-scale or fleet-wide onboarding.
- **User Effort**: Level of manual interaction required.
- **Hardware Dependency**: Need for secure elements, cameras, NFC, or other peripherals.
- **Ecosystem Adoption**: Evidence of real-world implementation in consumer or industrial ecosystems.

The findings were synthesized into comparative tables and narrative analyses, structured to assist system architects in choosing provisioning strategies aligned with their deployment constraints.

III. Provisioning methods

Provisioning methods in embedded systems can be broadly categorized based on how device identity and secrets are transferred during the onboarding process. Each approach has specific trade-offs in terms of **security guarantees, ease of use, hardware requirements, and scalability**, especially in the context of constrained platforms.

A. Manual Provisioning

Manual provisioning involves physically connecting devices to a host (e.g., PC or manufacturing tester) using USB, UART, or JTAG interfaces to inject configuration data and security credentials.

- **Pros**: Complete control over the provisioning process; no network dependency; suitable for debugging or secure labs.
- **Cons**: Labor-intensive and error-prone at scale; impractical for large-scale deployments [9].

This method is commonly used in early-stage development or in **factory provisioning lines** where devices are not yet connected to a network.

B. QR Code / Barcode Provisioning

In this method, provisioning information (such as device identifiers, authentication tokens, or public keys) is encoded into a QR code or barcode and scanned by a companion app (e.g., smartphone or tablet) during onboarding. It is widely supported in ecosystems like Matter [1] and Apple HomeKit [10].

- **Pros**: Intuitive user experience; no need for physical data transfer interfaces.
- **Cons**: Relies on camera-equipped devices; can expose data if the QR code is unencrypted or tampered with.

To mitigate risks, some protocols like Matter encrypt the QR payload and include **unique setup codes** per device, ensuring stronger entropy during onboarding.

C. Out-of-Band (OOB) Pairing

OOB provisioning transfers secrets over a **separate physical or wireless channel** from the main communication path. Common mediums include **NFC**, **ultrasound**, **audio tones**, or **LED blinking**. This method is used in BLE Secure Connections [4] and FIDO2 hardware authenticators [11].

- **Pros**: Strong protection against man-in-the-middle (MITM) attacks; suitable for offline or peer-to-peer setups.
- **Cons**: Hardware-specific dependencies; limited to environments where both devices support the same OOB medium.

For example, in BLE, OOB pairing allows pre-exchange of cryptographic keys before Bluetooth communication begins, minimizing MITM attack vectors.

D. Zero-Touch Provisioning (ZTP)

Zero-touch provisioning automates onboarding using a **factory-installed unique identity**, such as a Trusted Platform Module (TPM) key, serial number, or certificate. Upon power-up and internet connectivity, the device contacts a cloud-based provisioning service such as **AWS IoT Core [2]**, **Azure Device Provisioning Service (DPS) [3]**, or **Google Cloud IoT**.

- **Pros**: Fully scalable and hands-free for end users; aligns well with large-scale enterprise and industrial deployments.
- **Cons**: Requires secure identity storage and connectivity; onboarding often relies on correct factory configuration [12].

These platforms support **ownership transfer** and **lifecycle management**, which are essential for multi-tenant or enterprise IoT deployments.

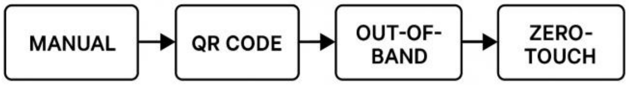

Fig. 1. *Device Provisioning Lifecycle*

IV. Provisioning algorithms and key exchanges

The security of a provisioning process heavily depends on the **underlying key exchange mechanism** used to establish trust and confidentiality between the device and its provisioning authority. Embedded systems often have limited computational resources, making it essential to select cryptographic schemes that are **both secure and lightweight**.

A. Elliptic Curve Diffie-Hellman (ECDH)

ECDH is a widely used asymmetric key exchange protocol that enables two parties to establish a shared secret over an insecure channel. Its small key sizes and computational efficiency make it ideal for constrained devices.

- **Use Cases**: BLE Secure Connections [4], Matter onboarding [1], and embedded TLS stacks.
- **Advantages**: Provides **forward secrecy** and minimal message overhead.
- **Limitations**: Requires random number generation and elliptic curve cryptography support, which may not be present in ultra-low-power MCUs without acceleration.

In BLE, ECDH is used in combination with Numeric Comparison or OOB to resist MITM attacks, while in Matter, it underpins secure commissioning via PASE and CASE modes.

B. Pre-Shared Key (PSK)-Based Provisioning

In PSK-based provisioning, both the device and the provisioning service share a secret key that was either manually entered or injected at manufacturing time.

- **Use Cases**: Legacy Wi-Fi WPA2-Personal networks, lightweight IoT deployments, and proprietary systems [9].
- **Advantages**: Computationally simple; no certificate chain validation needed.
- **Limitations**: **Difficult to rotate or revoke keys**, vulnerable to impersonation if the PSK is leaked.

This approach is suitable for **closed-loop deployments** or environments with no public network exposure.

C. Certificate-Based Provisioning

Certificate-based provisioning leverages **Public Key Infrastructure (PKI)** to authenticate the device using a certificate signed by a trusted Certificate Authority (CA). This is commonly used in enterprise and industrial settings.

- **Use Cases**: IEEE 802.1X [13], TLS 1.2/1.3 authentication in secure cloud onboarding, and X.509-based device identity.
- **Advantages**: Scalable trust model with revocation and rotation support; suited for multi-tenant systems.
- **Limitations**: Certificate management can be storage- and compute-intensive for embedded systems without secure elements or flash memory.

Secure bootloaders and TPM-backed identity schemes are increasingly used to support certificate validation with minimal attack surface.

D. Symmetric Key Derivation (e.g., HKDF)

HKDF (HMAC-based Extract-and-Expand Key Derivation Function) is a symmetric key derivation method used to derive multiple secure keys from a shared secret.

- **Use Cases**: BLE pairing [4], Matter commissioning flows [1], and TLS 1.3 key derivation [5].
- **Advantages**: Lightweight and fast; easy to implement in constrained firmware.
- **Limitations**: Relies on secure key pre-exchange (e.g., from ECDH or OOB).

HKDF is often combined with ECDH to derive session keys post-handshake and offers excellent modularity and cryptographic strength.

V. Provisioning chains and delegation of trust

In large-scale or distributed embedded environments, provisioning every device directly from a root authority can be impractical. A more scalable alternative is to use **trust chaining**, where a root-provisioned device can securely **delegate the authority** to onboard additional devices.

This approach is especially valuable in mesh or relay-based topologies (e.g., smart home or industrial clusters), where intermediate nodes assume partial provisioning responsibilities within a **bounded trust model**.

A. Concept of Trust Chaining

Trust chaining involves:

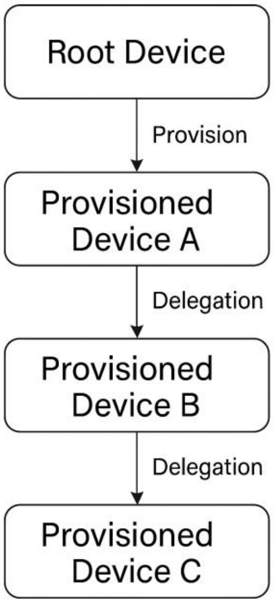

Fig. 2. *Trust Chain Delegation Flow*

1. A **root authority** (e.g., manufacturer, OEM cloud) issuing an identity or certificate to a **delegate node**
2. The delegate then **provisions or authenticates child devices** using derived keys, signed certificates, or cryptographic assertions
3. Each device in the chain can trace its identity **back to the root**, ensuring verifiability and revocation support

This model is conceptually similar to certificate hierarchies in PKI and is actively used in protocols like Matter and Thread Mesh networks [1][6].

B. Embedded System Use Cases

- **Thread and Matter Commissioning**: A controller device (e.g., smartphone) provisioned with the user's credentials securely onboards other devices in the home using **CASE (Certificate Authenticated Session Establishment)** [1].

- **BLE Relays or Gateways**: A BLE-enabled hub or gateway, once provisioned, generates or distributes session keys to peripherals (e.g., sensors, smart locks) over a secure BLE or Wi-Fi link.
- **Manufacturing Delegation**: A trusted test jig or fixture at a production line may act as a proxy for the OEM provisioning server, using **pre-loaded certificates** or device keys to enable offline onboarding.

C. Benefits and Risks

- **Pros**:
  - Reduces reliance on cloud connectivity for every onboarding event
  - Enables **offline or near-field provisioning**
  - Can scale across factory floors, homes, or industrial zones
- **Cons**:
  - Requires strong security boundaries between parent and child nodes
  - If a delegate is compromised, downstream devices may be exposed unless proper revocation or key rotation mechanisms are in place

To mitigate this, trust chaining schemes often employ **short-lived certificates**, **revocation lists**, or **attestation mechanisms** to preserve security across delegated boundaries.

VI. Comparative analysis

To guide selection among provisioning techniques, Table I presents a structured comparison across five key dimensions:

- **Security**: Resistance to common threats such as spoofing, man-in-the-middle (MITM), or replay attacks
- **Scalability**: Ease of deploying the method across a fleet or factory-scale operation
- **User Effort**: Degree of manual interaction or technical knowledge required during onboarding
- **Hardware Dependency**: Need for specific components (e.g., cameras, NFC chips, TPM)
- **Example Use Cases**: Real-world applications or protocol contexts

Table I — Comparative Summary of Provisioning Methods

Method	Security	Scalability	User Effort	Hardware Dependency	Example Use Cases
Manual	High	Low	High	Debug interfaces (USB, JTAG)	Factory-line flashing, lab diagnostics
QR Code	Med	Med	Low	Camera on user device	Matter, Apple HomeKit onboarding [1][10]
OOB (e.g., NFC)	High	Med	Med	NFC, audio, or optical hardware	BLE Secure Connections, FIDO2 [4][11]
Zero-Touch	High	High	None	TPM, secure boot, internet	AWS IoT Core, Azure DPS [2][3]
Trust Chain	High	High	Low	Root identity, certificate store	Thread, Matter CASE, BLE hubs [1][6]

This comparative framework enables designers to select provisioning strategies aligned with their device class, deployment scale, and threat model. For example, trust chaining is ideal for mesh networks or gateways in industrial settings, whereas QR code onboarding may suffice for consumer-grade devices with user-friendly requirements.

VII. Future directions

As embedded systems continue to evolve in scale and sophistication, provisioning techniques must also adapt to new security threats, operational requirements, and standardization needs. Several promising avenues for future research and development are outlined below.

A. Post-Quantum Secure Provisioning

With the advancement of quantum computing, existing cryptographic primitives like ECDH and RSA will become vulnerable. Provisioning schemes must begin transitioning toward **quantum-resistant algorithms**, such as those emerging from NIST's Post-Quantum Cryptography Standardization project [7]. This transition is particularly critical for long-lived IoT deployments where keys and certificates may need to remain secure for decades.

Research Opportunity: Design lightweight post-quantum onboarding protocols that are feasible for microcontroller-class devices.

B. AI-Based Anomaly Detection in Trust Propagation

Current provisioning flows assume secure and linear trust chains. However, in dynamic environments, compromised devices or misconfigured provisioning agents can lead to unauthorized access. **AI and ML-based anomaly detection** could be employed to monitor provisioning events and trust delegation paths for abnormal patterns.

Example: A model could flag deviations in expected onboarding flow—for example, an unusually high number of child device provisions from a single edge node.

C. Decentralized Trust Logs and Key Revocation via Blockchain

Revocation remains a weak point in provisioning schemes. Traditional Certificate Revocation Lists (CRLs) and Online Certificate Status Protocols (OCSP) are not optimal for offline or intermittently connected systems. Blockchain-based ledgers can offer **tamper-resistant trust logs** and distributed revocation visibility [14].

Research Opportunity: Evaluate the feasibility of hybrid models that combine local trust caches with globally auditable revocation registries for embedded platforms.

D. Standardized Cross-Vendor Provisioning Interfaces

Fragmentation among OEMs, cloud platforms, and connectivity protocols hinders seamless provisioning. Initiatives like **FIDO Device Onboarding (FDO) [15]** and **IETF SUIT [16]** aim to define standard schemas and lifecycle workflows across vendors and ecosystems. Adoption of such standards will be key to reducing integration friction.

Design Challenge: Mapping standard protocols (e.g., SUIT manifests) to constrained firmware environments with minimal overhead.

VIII. Real world use cases

While provisioning models may appear well-defined in theory, practical deployment in embedded systems often reveals critical constraints and vulnerabilities. This section outlines **real-world implementations**, engineering trade-offs, and known challenges in the field.

A. Real-World Use Cases

- **Consumer Smart Home (Matter Ecosystem)**
  - Devices are provisioned using QR codes with encrypted payloads.
  - A smartphone app scans the QR, establishes an ECDH-based session, and provisions credentials securely [1].
  - Trust chaining is used when smartphones or hubs onboard child devices such as lights or thermostats.

- **Industrial IoT (AWS IoT Zero-Touch Provisioning)**
  - Devices with a pre-installed X.509 certificate boot up and reach out to AWS IoT Core [2].
  - Factory tools inject device identity tied to the customer account.
  - TLS with mutual authentication ensures onboarding is secure, automated, and auditable.

- **Wearables and Mobile Accessories (BLE + OOB)**
  - Smartwatches, glasses, and fitness bands often use BLE with OOB channels such as NFC or visual pairing.
  - Secrets may be transmitted from the phone to the accessory to initiate bonding with minimum user input.
  - Hardware constraints drive adoption of symmetric key derivation (HKDF) with encrypted L2CAP [4].

- **Connected Vehicles**
  - Telematics units and ECUs may use TPMs to store identity keys securely.
  - Provisioning occurs via cellular or factory Wi-Fi using certificate-based mutual TLS [13].

IX. Implementation consideration and pitfalls

While provisioning techniques vary by application, several **recurring themes** emerge in real-world deployments:

- **Entropy and RNG Sources** Devices must use secure entropy sources for cryptographic operations. Inconsistent or weak RNGs can undermine ECDH or key derivation steps.

- **Secure Storage and Key Persistence** Storing credentials in raw flash is risky. Designers must use secure elements or encrypt-at-rest techniques. Devices with limited flash wear endurance must also handle key rotation carefully.

- **Cloud Dependency** Zero-touch provisioning methods require early boot access to cloud services. Lack of internet, incorrect time sync, or DNS failures can halt onboarding.

- **Hardcoded Keys and Uniform Secrets** Some deployments mistakenly use the same PSK or QR code across batches, creating systemic vulnerabilities.

- **Fallback to Insecure Modes** Devices that default to open BLE pairing or unencrypted Wi-Fi AP provisioning when onboarding fails present attack surfaces for adversaries.

- **Poor Revocation and Lifecycle Management** Devices lacking effective revocation mechanisms can continue trusting compromised certificates. Embedded systems with long lifespans are especially vulnerable to stale trust anchors.

- **Neglecting Offline Scenarios** Many provisioning systems assume consistent cloud connectivity. Air-gapped or field-deployed environments require alternate workflows—such as delegate-based provisioning or USB-based fallbacks.

X. Conclusion

Secure and scalable provisioning is a foundational requirement for embedded systems operating across consumer, industrial, and mission-critical domains. As devices become more connected and autonomous, provisioning strategies must ensure robust identity management, cryptographic integrity, and lifecycle trust—all while operating within tight resource constraints.

This paper presented a **comprehensive survey and comparative analysis** of provisioning methods tailored to embedded systems. It categorized provisioning mechanisms by approach—manual, QR code, out-of-band (OOB), zero-touch, and trust chaining—and analyzed their respective trade-offs in security, user effort, hardware requirements, and scalability. Real-world use cases from Matter, BLE, AWS IoT, and industrial automation were discussed, along with critical deployment considerations such as secure storage, entropy generation, revocation, and offline provisioning support.

By connecting cryptographic algorithms like ECDH, PSK, and certificate-based authentication to practical embedded implementations, the paper also outlined how key exchange protocols can be matched to hardware profiles and threat models. The introduction of a trust chaining model highlights a scalable path forward, enabling decentralized, resilient onboarding workflows.

Looking ahead, emerging challenges—such as post-quantum threats, AI-driven anomaly detection, blockchain-based revocation, and cross-vendor onboarding standards—offer rich opportunities for future research and innovation.

Ultimately, secure provisioning must be treated not as a one-time event but as an **ongoing capability**—designed with adaptability, transparency, and trust at its core.

References

[1] Connectivity Standards Alliance, *"Matter Protocol Specification v1.2,"* 2023.

[2] Amazon Web Services, *"AWS IoT Core Device Provisioning,"* [Online]. Available: https://docs.aws.amazon.com/iot/latest/developerguide/provision-devices.html
[3] Microsoft Azure, *"Azure Device Provisioning Service (DPS),"* [Online]. Available: https://learn.microsoft.com/en-us/azure/iot-dps/
[4] Bluetooth Special Interest Group (SIG), *"Bluetooth Core Specification v5.3,"* 2021.
[5] H. Krawczyk and P. Eronen, *"RFC 5869: HMAC-based Extract-and-Expand Key Derivation Function (HKDF),"*IETF, 2010.
[6] Google LLC, *"Nest Device Commissioning Architecture,"* 2022. [Whitepaper].
[7] National Institute of Standards and Technology (NIST), *"Post-Quantum Cryptography Standardization Project,"*[Online]. Available: https://csrc.nist.gov/projects/post-quantum-cryptography
[8] IEEE 802.1 Working Group, *"IEEE Std 802.1AR-2018: Secure Device Identity,"* IEEE, 2018.
[9] A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, *"LEAP: Efficient Security Mechanism for Resource-Constrained Devices,"* in ACM EmNets, 2004.
[10] Apple Inc., *"HomeKit Accessory Protocol Specification R2,"* 2021.
[11] FIDO Alliance, *"FIDO2 Technical Overview,"* 2020. [Whitepaper].
[12] S. Arshad, A. Azad, and M. Farooq, *"A Trustworthy Device Identity Provisioning System,"* IEEE Access, vol. 6, pp. 46992–47004, 2018.
[13] T. Dierks and E. Rescorla, *"RFC 5246: The Transport Layer Security (TLS) Protocol Version 1.2,"* IETF, 2008.
[14] S. Nakamoto, *"Bitcoin: A Peer-to-Peer Electronic Cash System,"* 2008.
[15] FIDO Alliance, *"FIDO Device Onboarding (FDO) Specification,"* 2022.
[16] IETF SUIT Working Group, *"Software Updates for Internet of Things (SUIT),"* IETF, 2023. [Online]. Available: https://datatracker.ietf.org/wg/suit/

Nikheel V. Savant (Senior Member, IEEE) received the B.E. degree in Electronics and Communication Engineering from B.V. Bhoomaraddi College of Engineering and Technology, Hubli, India, in 2013, and the M.S.E. degree in Embedded Systems from the University of Pennsylvania, Philadelphia, PA, USA, in 2016.

He has held engineering roles at Apple as a Wi-Fi Systems Software Engineer and at Tesla as a Vehicle Connectivity Intern, where he focused on wireless communication protocols and automotive telemetry. He is currently a Senior Software Engineer at Meta, where he leads the development and optimization of Bluetooth protocols for next-generation wearable and embedded platforms.

His research interests include Bluetooth protocol stacks, embedded wireless systems, AI-driven connectivity diagnostics, and low-power communication architectures. He received the Gold Medal for academic excellence during his undergraduate studies. Mr. Savant is a Senior Member of the IEEE and actively contributes to standardization efforts within the Bluetooth Special Interest Group (SIG).