
PKOC SPECIFICATION

# PKOC BLE Transport Profile

*Public Key Open Credential over Bluetooth Low Energy*

**Version:** 2.0.1

**Status / Classification:** Public — Approved for open implementation

**Date:** August 13, 2026

**Maintained by:** Physical Security Interoperability Alliance (PSIA)

**Supersedes:** PKOC BLE Specification v3.1.2

## Document Revision History

| Version | Date | Author | Description |
|---|---|---|---|
| 3.0.0 | Oct 15, 2024 | PSIA SCI Working Group | Added ECDHE Perfect Forward Secrecy flow, AES-CCM encryption, and protocol versioning. |
| 3.1.1 | Apr 15, 2026 | PSIA SCI Working Group | Aligned ECDHE with BSI TR-03111. Added explicit KDF, AES-CCM parameters, IV format, signature-input normalization, and site-key provisioning. |
| 3.1.2 | Jun 11, 2026 | PSIA SCI Working Group | Introduced per-reader signing keys with a Site Issuer trust anchor, the Reader Certificate TLV (0x10), discovery-and-pin caching, and a revocation mechanism. Legacy shared-Site-Key model retained for backward compatibility. |
| 2.0.1 | Aug 13, 2026 | PSIA SCI Working Group | Re-issued as a Transport Profile of the PKOC Core Specification 2.0.1. Transport-independent content moved to the Core. Front-matter order standardized (Revision History → Contents → Glossary → Conformance). High-level interaction diagram added and placed before the requirement lists; registration handled by the Core. Requirement lists and formatting aligned with the NFC profile. Supersedes PKOC BLE v3.1.2; the wire protocol is unchanged. Added an explicit statement that Validated Mode and the Reader Certificate are optional and that conformance is vendor-neutral. |

# Table of Contents

## Glossary

This glossary lists terms specific to the BLE profile. General PKOC terms (Credential, Reader, PACS, Trust Anchor, ECDSA, and so on) are defined in the PKOC Core Specification and are not repeated here. Terms marked “Legacy” apply to the shared-Site-Key model retained for backward compatibility (Appendix B).

| Term | Definition |
|---|---|
| AES-CCM | Advanced Encryption Standard in Counter with CBC-MAC mode; the authenticated encryption scheme used by the ECDHE flow. |
| ATT MTU | The negotiated Attribute Protocol Maximum Transmission Unit for a BLE connection. |
| BLE | Bluetooth Low Energy; the short-range, low-power radio link used by this profile. |
| BSI TR-03111 | German Federal Office for Information Security Technical Recommendation 03111, “Elliptic Curve Cryptography,” whose ECKA-DH construction this profile follows. |
| ECKA-DH | Elliptic Curve Key Agreement using Diffie-Hellman, as defined in BSI TR-03111 §4.3.1. |
| GATT | Generic Attribute Profile; the BLE service/characteristic framework carrying PKOC messages. |
| Hardware Attestation | A vendor-signed proof that a key was generated inside a genuine secure element and cannot be exported. Used at reader enrollment (Appendix A). |
| Reader Certificate | A Site Issuer-signed structure binding a Reader Public Key to a Reader Location Identifier, presented by the reader during the ECDHE handshake. Format in §7.1. This is the BLE realization of the Core's Validated trust model. |
| Reader Location Identifier | A 16-byte opaque identifier for a reader's physical location, asserted in TLV 0x0D and bound into the Reader Certificate subject. |
| Reader Signing Key | A per-reader long-term EC key pair generated in hardware-backed secure storage; its public half is bound into the Reader Certificate and it signs the ECDHE handshake. |
| Site Issuer Key | A per-site long-term EC key pair held by the site operator (typically in an HSM). Its private half signs Reader Certificates and revocation lists; its public half is the per-site Trust Anchor provisioned to credentials. |
| Site Issuer Identifier | A 16-byte opaque identifier for the Site Issuer, carried in TLV 0x0E. |
| Site Key (Legacy) | The shared per-site EC key pair used by the 3.1.1 flow; superseded by the Site Issuer Key plus per-reader Reader Signing Keys. |
| TLV | Type-Length-Value; the binary message encoding used by the BLE protocol. |

## Conformance

The key words **MUST**, **MUST NOT**, **REQUIRED**, **SHALL**, **SHALL NOT**, **SHOULD**, **SHOULD NOT**, **RECOMMENDED**, **MAY**, and **OPTIONAL** in this document are to be interpreted as described in RFC 2119.

An implementation is conformant to a given profile only if it satisfies every **MUST** and **MUST NOT** requirement defined by the PKOC Core Specification and by that profile. Requirements are highlighted in **bold** throughout this document.

**Standard Mode alone is sufficient for PKOC conformance.** Validated Mode, the Reader Certificate, the Site Issuer, and Trust Anchors are **OPTIONAL** capabilities. A Device or Reader that implements only Standard

Mode — including an unmodified PKOC BLE v3.1.2 implementation — is fully PKOC conformant. Where this document states a **MUST**, **MUST NOT**, **SHALL**, or **SHALL NOT** in connection with Validated Mode, a Reader Certificate, a Site Issuer, or a Trust Anchor, that requirement applies only to an implementation that has elected to support the optional capability in question, and is never in itself a condition of PKOC conformance.

No requirement in this profile may be satisfiable only through a particular vendor's product. Every element needed to build a conforming Device, Reader, or Site Issuer is specified in this document and the Core, and **MAY** be implemented by any party without licence, royalty, registration, or approval.

# 1. Introduction

## 1.1 Relationship to the PKOC Core Specification

This document is a **Transport Profile** of the PKOC Core Specification. It defines how PKOC credentials are exchanged over Bluetooth Low Energy (BLE) between a Device (a smartphone or wearable acting as the Credential) and a Reader (an access-control reader). All transport-independent concepts — the PKOC credential model, cryptographic foundations, key encodings, the credential registration process, and the common Credential/Reader requirements — are defined in the Core and are not repeated here. Where this profile is silent on a transport-independent matter, the Core governs.

## 1.2 Scope of the BLE Profile

This profile specifies the BLE-specific binding: advertisement and connection, the GATT service, TLV framing, MTU and fragmentation, the two transmission flows (ECDHE Perfect Forward Secrecy and Un-Obfuscated), the per-reader signing-key mechanism with Reader Certificates, error handling, and the BLE device and reader requirements.

## 1.3 Changes in This Version

This edition re-issues the former PKOC BLE Specification v3.1.2 as a profile of the Core. The **wire protocol is unchanged from 3.1.2**; a 3.1.2 implementation is byte-compatible with this profile. The changes are structural and editorial: transport-independent material now lives in the Core; the high-level process overview (Section 3) is placed before the requirement lists; registration is described once in the Core and referenced here; and the document has been organized to parallel the NFC profile — explicit Operating Modes and Credential Types (Section 2), a common-requirements structure with a distinct Validated Mode Requirements section (Section 4), and consistent language, tables, and formatting — so the set reads as one standard.

The substantive protocol capability carried forward from 3.1.2 is the **per-reader signing-key architecture**: each reader holds a unique signing key in hardware-backed storage, and a Site Issuer signs a Reader Certificate binding that key to the reader's location. Credentials trust a single per-site Site Issuer public key. This bounds the compromise of any one reader to that reader. The legacy shared-Site-Key model (3.1.1) remains supported for backward compatibility and is specified in Appendix B.

## 1.4 Backward Compatibility and Vendor Neutrality

**Nothing in this version changes, deprecates, or invalidates any deployed PKOC implementation.**

- **PKOC BLE v3.1.2.** The wire protocol is unchanged and byte-compatible. A 3.1.2 Device and a 3.1.2 Reader remain fully conformant to this version without modification. The changes in this edition are structural and editorial.
- **PKOC BLE v3.1.1.** The legacy shared-Site-Key model remains supported and is specified in Appendix B.
- **PKOC NFC.** Changes made in the NFC Transport Profile — including its card profiles, its optional PKOC-CVC, and its EV support — have no effect of any kind on BLE implementations. The two profiles share only the Core's credential derivation and cryptographic baseline, neither of which has changed.
- **Credential stability.** A given NIST P-256 public key yields the same PKOC Credential and the same PKOC Derived Identifier in this version as in v3.1.2, on every profile and every transport. Enrolled credential records do not need to be re-derived, re-issued, or re-enrolled.
- **Additive only.** Validated Mode, the Reader Certificate, and the Site Issuer trust anchor are optional elections and impose no obligation on an implementation that does not use them.

PKOC is an open, royalty-free standard. Every element of this profile **MAY** be implemented by any party without licence, royalty, or restriction. This profile **MUST NOT** be implemented, extended, or profiled in a way that makes PKOC conformance dependent on a particular vendor's product, secure element, certificate hierarchy, key-distribution service, registry, or intellectual property. Any organization **MAY** act as a Site Issuer and operate its own Trust Anchor; PKOC defines no central issuing authority and no approval process.

# 2. BLE Architecture

This section defines the BLE-specific architecture: the operating modes, the mapping of Core roles onto BLE link roles, and the credential output the Reader produces for the PACS.

## 2.1 Operating Modes

| Mode | Conformance | Trust establishment |
|---|---|---|
| Standard Mode | Required | The Reader verifies proof of possession of the PKOC private key; the Credential's public key (or a derived identifier) is trusted through prior PACS enrollment. No Reader Certificate is required. |
| Validated Mode | Optional | The Reader presents a Site Issuer-signed Reader Certificate that the Credential validates against its provisioned Trust Anchor, authenticating the Reader to the Credential before the credential is presented. |

In both modes the Reader **MUST** verify the Credential's challenge signature against the presented PKOC public key. In Validated Mode the Credential **MUST** validate the Reader Certificate (Section 7) against its Trust Anchor and **MUST** abort the transaction if validation fails.

## 2.2 Roles

This profile maps the Core roles onto BLE link roles:

| Core role | BLE realization |
|---|---|
| Credential | A **Device** (smartphone/wearable) operating as a BLE **peripheral**, advertising the PKOC service and receiving connections. |
| Reader | An access-control reader operating as a BLE **central**, scanning for and connecting to Devices. |
| Provisioning / Issuer Backend | The PACS cloud/provisioning service that enrolls credentials, distributes the Site Issuer public key, and publishes revocation lists. |

In the Validated trust model, the BLE profile authenticates the **Reader** to the Device: the reader presents a Site Issuer-signed Reader Certificate and signs the handshake with its per-reader signing key, and the Device verifies both against the per-site Trust Anchor it received at enrollment (Core Section 6).

## 2.3 Credential Types

The Reader derives its Reader-to-PACS output from the Credential's PKOC public key per Core Section 4. Because the BLE flows carry the full PKOC public key, the Reader can output either the full PKOC Credential or a shorter PKOC Derived Identifier, according to configuration. The BLE Validated model attests the Reader rather than the Credential, so the credential output type is the same in both operating modes; Validated Mode adds Reader authentication and, in the ECDHE flow, session encryption. Reader-to-PACS signaling is out of scope for this profile.

# 3. High-Level Process Overview

This section presents the end-to-end process before the detailed requirements and wire format, so the flow is understood first. The common processes are connection setup, flow selection, Reader validation (in Validated Mode), mutual authentication, credential presentation, and Reader to PACS. Enrollment (one-time, from the credential's perspective) is detailed in Appendix A; the runtime exchange follows Section 6.

## 3.1 ECDHE Perfect Forward Secrecy Flow

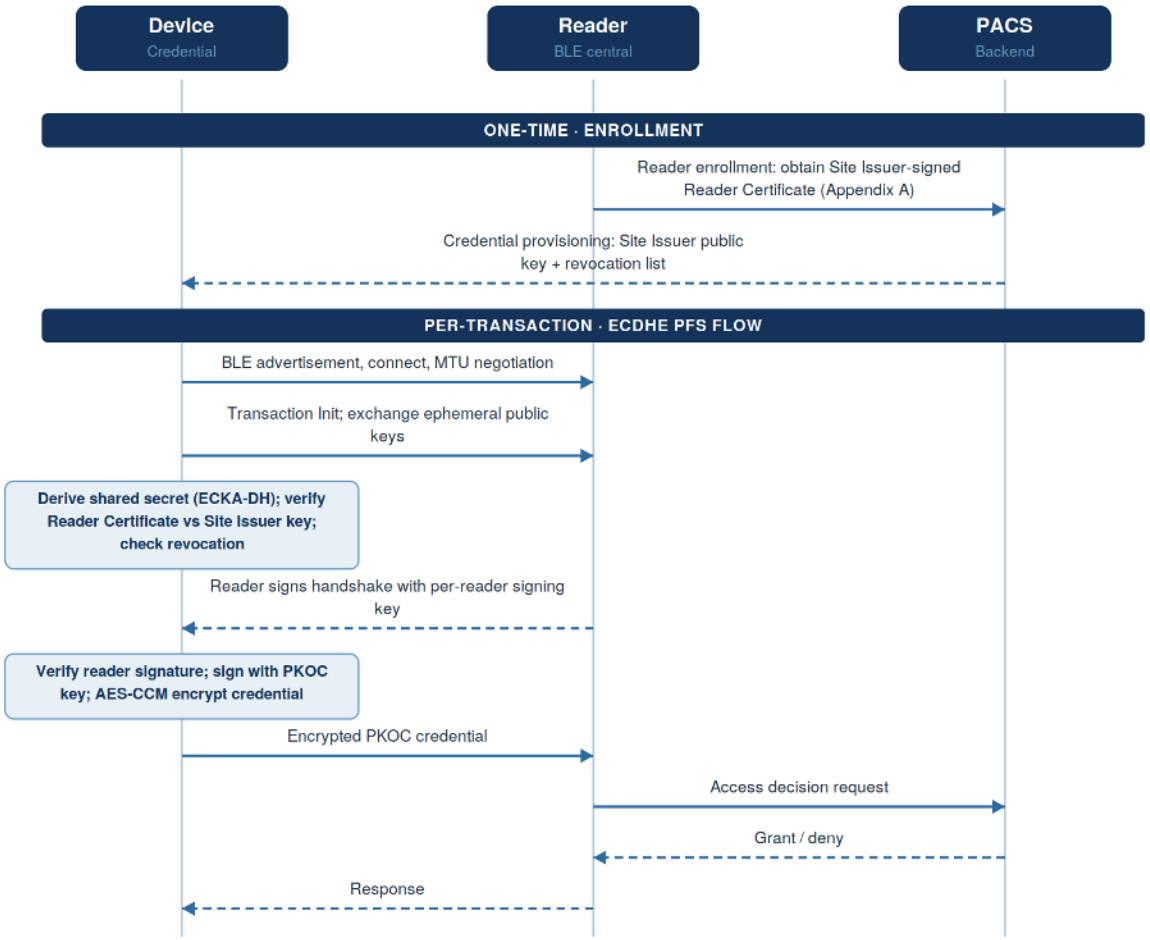


*Figure 3-1. High-level BLE interaction for the ECDHE Perfect Forward Secrecy flow, including one-time enrollment and the per-transaction handshake.*

1. **Reader enrollment (one-time).** The reader generates its Reader Signing Key and obtains a Site Issuer-signed Reader Certificate (Appendix A).
2. **Credential provisioning.** The Device receives the Site Issuer public key and an initial revocation list during credential enrollment (Core Section 6).
3. **Connection setup.** Advertisement, connection, and MTU negotiation.
4. **ECDHE handshake.** Ephemeral key exchange and shared-secret derivation.
5. **Reader Certificate verification.** The Device verifies the certificate against the Site Issuer public key, checks revocation, and caches or updates the pinned reader key.
6. **Mutual authentication.** Reader and Device exchange and verify signatures.
7. **Encrypted credential presentation.** The Device sends its AES-CCM-encrypted PKOC credential.

8. **Access decision and response.** The Reader consults the PACS and returns the result.

### 3.2 Un-Obfuscated Flow

The Un-Obfuscated flow omits session encryption. The Reader issues a challenge (its compressed ephemeral public key); the Device signs it with the PKOC private key and returns the PKOC public key, signature, and optional last-update time in plaintext; the Reader verifies and reports the decision. The message-level sequence is shown in Section 8.

## 4. BLE Common Requirements

These requirements are in addition to the common Credential, Reader, and cryptographic requirements of the Core (Core Section 7). A conformant BLE implementation **MUST** satisfy both. All BLE implementations **MUST** satisfy Sections 4.1, 4.2, and 4.4; Validated Mode adds the requirements in Section 4.3.

### 4.1 Device Requirements

1. The Device **MUST** support BLE peripheral mode, capable of receiving connection requests from a Reader acting as BLE central.
2. The Device **MUST** implement the PKOC GATT service with the UUIDs specified in Section 5.
3. The Device **MUST** support at minimum the Un-Obfuscated flow (Section 6.3).
4. The Device **SHOULD** support the ECDHE Perfect Forward Secrecy flow (Section 6.2) when the reader requests it.
5. The Device **MUST** generate fresh ephemeral keys using a CSPRNG for every ECDHE transaction and **MUST** discard them immediately after the transaction.
6. The Device **MUST** maintain backward compatibility with legacy readers as described in Appendix B.

### 4.2 Reader Requirements

1. The Reader **MUST** support BLE central mode, capable of scanning for and connecting to Devices.
2. The Reader **MUST** implement the PKOC GATT client with the UUIDs specified in Section 5.
3. The Reader **MUST** support at minimum the Un-Obfuscated flow and **SHOULD** support the ECDHE Perfect Forward Secrecy flow.
4. The Reader **MUST** generate fresh ephemeral keys using a CSPRNG for every ECDHE transaction and **MUST** discard ephemeral and session keys immediately after each transaction.
5. The Reader **MUST** verify the Credential's challenge signature against the presented PKOC public key before deriving or outputting a credential.

### 4.3 Validated Mode Requirements

Validated Mode is optional. These requirements apply only when the BLE profile is operated in Validated Mode (per-reader signing keys with Reader Certificates, Section 7). An implementation that supports only Standard Mode is fully conformant and is not subject to this section.

1. A Reader operating in Validated Mode **MUST** generate its Reader Signing Key in hardware-backed secure storage; the Reader Signing Private Key **MUST NOT** be exportable.
2. A Reader operating in Validated Mode **MUST NOT** possess the Site Issuer Private Key, which is held only by the site operator.
3. A Reader operating in Validated Mode **MUST** store a Site Issuer-signed Reader Certificate and **MUST** transmit it (TLV 0x10) during the ECDHE handshake alongside the Digital Signature (TLV 0x03).
4. A Reader operating in Validated Mode **MUST** sign the ECDHE handshake input (Section 6.2.2) with the Reader Signing Private Key, not the Site Issuer Private Key.

5. A Device operating in Validated Mode **MUST** verify a received Reader Certificate (TLV 0x10) against the stored Site Issuer public key and **MUST** reject the transaction with error 0x07 if the certificate subject does not match the Reader Location Identifier in TLV 0x0D.
6. A Device operating in Validated Mode **MUST** honor certificate validity windows (reject expired or not-yet-valid certificates with error 0x09) and **MUST** reject a transaction whose Reader Location Identifier appears in the cached revocation list (error 0x08).
7. A Device **MAY** cache verified Reader Public Keys (discovery-and-pin, Section 7.2); caching is **OPTIONAL**, and a conforming Device **MAY** perform full verification on every transaction.

### 4.4 Cryptographic Requirements

In addition to the Core cryptographic requirements:

1. Signatures transmitted over BLE **MUST** be raw R ‖ S (64 bytes); DER-encoded signatures **MUST** be converted per Core Section 3.3.
2. For the ECDHE flow, key agreement **MUST** follow BSI TR-03111 §4.3.1 (ECKA-DH). The shared secret is `Z_AB = SHA-256(x_S)`, where `x_S` is the X-coordinate of the shared ECDH point, used directly as the 256-bit AES-CCM key.
3. Reader Certificate and revocation-list signatures **MUST** use ECDSA-SHA256 on NIST P-256, transmitted as raw R ‖ S.

## 5. BLE Exchange

### 5.1 BLE Advertisement

The Reader scans for Devices; the Device (peripheral) advertises the PKOC service UUID. Upon detecting the PKOC service, the Reader (central) initiates a connection.

PKOC BLE Service UUID: `0000ABCD-0000-1000-8000-00805F9B34FB` (example; see the official PSIA registry for the assigned UUID). The advertisement contains at minimum the PKOC service UUID and, optionally, truncated reader-identity information.

### 5.2 GATT Service Structure

The PKOC GATT service contains two characteristics:

| Characteristic | Properties | Description |
|---|---|---|
| PKOC Write | Write, Write Without Response | Device-to-Reader data channel. |
| PKOC Notify | Notify, Indicate | Reader-to-Device data channel. |

The Device writes TLV-encoded messages to PKOC Write; the Reader responds via PKOC Notify. The Device **MUST** subscribe to notifications/indications before initiating a transaction.

### 5.3 TLV Encoding

All messages use Type-Length-Value encoding: **Type** is 1 byte; **Length** is 1–3 bytes (`0x00–0x7F` as a single byte, `0x81 <len>` for one length byte, `0x82 <len_hi> <len_lo>` for a two-byte big-endian length); **Value** is Length bytes. TLVs may be concatenated in one packet and may be nested.

### 5.4 TLV Type Definitions

| Type | Name | Description |
|---|---|---|
| 0x01 | PKOC Public Key | Device long-term PKOC public key (65 bytes, uncompressed). |

| Type | Name | Description |
|---|---|---|
| 0x02 | Reader Ephemeral Public Key (Compressed) | Reader ephemeral EC public key, compressed (33 bytes). |
| 0x03 | Digital Signature | 64-byte raw ECDSA signature (R ‖ S). |
| 0x04 | Response | Reader access-decision response. |
| 0x05 | Transaction Initiation | Device initiates a new transaction. |
| 0x06 | Error | Error code and optional description. |
| 0x07 | Device Ephemeral Public Key (Uncompressed) | Device ephemeral EC public key, uncompressed (65 bytes). |
| 0x09 | Last Update Time | Timestamp of last credential update (8 bytes, Unix epoch ms). |
| 0x0A | Credential Identifier | Opaque credential identifier from the provisioning backend. |
| 0x0C | Protocol Identifiers | Protocol version and feature negotiation (Section 5.6). |
| 0x0D | Reader Location Identifier | 16-byte opaque reader-location identifier; bound into the Reader Certificate subject. |
| 0x0E | Site Issuer Identifier | 16-byte opaque Site Issuer identifier (also the legacy Site Key identifier). |
| 0x10 | Reader Certificate | Site Issuer-signed certificate binding a Reader Public Key to a Reader Location Identifier (Section 7.1). Typically 138 bytes. |
| 0x40 | Encrypted Data | AES-CCM encrypted payload (ciphertext ‖ 16-byte tag). |

Types 0x08, 0x0B, and 0x0F are reserved for future use.

## 5.5 MTU and Fragmentation

The default ATT MTU (23 bytes) is too small for PKOC messages. The Device and Reader **MUST** negotiate an ATT MTU of at least 185 bytes at connection setup. If negotiation fails, implementations **MUST** fragment and reassemble at the application layer: the sender splits the TLV stream into (MTU − 3)-byte chunks, prefixing each with a 1-byte indicator (`0x00` first, `0x01` continuation, `0x02` last); the receiver reassembles until the last-fragment indicator.

> **Note:** The Reader Certificate TLV (0x10) is approximately 141 bytes on the wire. Combined with the other TLVs in the reader's response, the total may exceed a single BLE write at the negotiated MTU, so implementations **MUST** support fragmentation.

## 5.6 Protocol Identifiers (TLV 0x0C)

The Protocol Identifiers TLV negotiates version and features:

| Offset | Field | Size | Description |
|---|---|---|---|
| 0 | Spec Version Major | 1 | e.g., 0x03 for the 3.x wire format. |
| 1 | Spec Version Minor | 1 | e.g., 0x01 for 3.1.x. |
| 2–3 | Vendor Sub-Version | 2 | Vendor-specific. |
| 4–5 | Feature Bits | 2 | bit 0 = AES-CCM support; bit 1 = Reader Certificate support. |

Wire-format example for both AES-CCM and Reader Certificate support:

```
0x0C 0x06 0x03 0x01 0x00 0x00 0x00 0x03
```

A per-reader-capable reader sets bit 1 and transmits TLV 0x10. A legacy reader does not set bit 1 and does not transmit TLV 0x10; the Device then falls back to the legacy verification path (Appendix B).

### 5.7 Error Handling

On error, the detecting party **MUST** send an Error TLV (0x06) and close the BLE connection.

| Code | Name | Description |
|---|---|---|
| 0x01 | Unsupported Protocol Version | The requested protocol version is not supported. |
| 0x02 | Signature Verification Failed | A received digital signature did not verify. |
| 0x03 | Decryption Failed | AES-CCM tag validation failed. |
| 0x04 | Unknown Site Issuer | The Site Issuer Identifier is not recognized. |
| 0x05 | Session Expired | The encryption counter has rolled over. |
| 0x06 | Invalid TLV | Malformed TLV data received. |
| 0x07 | Reader Certificate Verification Failed | Certificate did not verify against the Site Issuer key, or subject did not match TLV 0x0D. |
| 0x08 | Reader Certificate Revoked | The Reader Location Identifier appears in the cached revocation list. |
| 0x09 | Reader Certificate Expired | The certificate validity window does not include the current time. |
| 0xFF | General Error | Unspecified error. |

## 6. Transmission Flows

### 6.1 Flow Selection

The profile defines two flows. The **ECDHE Perfect Forward Secrecy** flow uses ephemeral keys and AES-CCM and, in the per-reader model, authenticates the reader via its Reader Certificate; it is **RECOMMENDED** for new deployments. The **Un-Obfuscated** flow is simpler and unencrypted, suitable where PFS is not required. The Reader signals the flow by the TLVs it returns to the Transaction Initiation: ephemeral key plus Protocol Identifiers indicates ECDHE; a compressed ephemeral key alone indicates Un-Obfuscated. Within ECDHE, the presence of the Reader Certificate TLV and its feature bit selects the per-reader path versus the legacy path (Appendix B).

### 6.2 ECDHE with Perfect Forward Secrecy Flow

This flow establishes a secure channel using ECKA-DH Anonymous key agreement (BSI TR-03111 §4.3.1). Per-session ephemeral keys are discarded immediately, providing Perfect Forward Secrecy. In the per-reader model the Reader presents a Reader Certificate and signs with its unique signing key, bounding compromise to individual readers.

#### 6.2.1 Prerequisites

The Device **MUST** already hold the site Trust Anchor (the Site Issuer public key; or the legacy Site public key) before initiating an ECDHE transaction. It is provisioned out-of-band per Core Section 6 (cloud enrollment, QR code, or administrator configuration). TLV 0x0E identifies which anchor to use but does not carry the key. Reader provisioning is described in Appendix A.

### 6.2.2 Signature Input

To prevent man-in-the-middle attacks, both parties sign the identical 96-byte input, concatenated in this exact order and never reordered between parties:

| Data | Length | Source |
|---|---|---|
| Site Issuer Identifier | 16 | Reader/site configuration. |
| Reader Location Identifier | 16 | Reader configuration. |
| Device Ephemeral Public Key X | 32 | Counterparty. |
| Reader Ephemeral Public Key X | 32 | Local ephemeral key pair. |
| Total | 96 | |

The X component is bytes 1–32 of the 65-byte uncompressed key (excluding the `0x04` prefix). The signature is `ECDSA-SHA256(private_key, SHA-256(signature_input))`. In the per-reader model the Reader signs with its Reader Signing Private Key; in the legacy model it signs with the shared Site Private Key. The Device always signs with the PKOC Private Key.

### 6.2.3 Shared Secret Derivation

1. Compute the ECDH shared point `S = d_local × Q_remote` on NIST P-256.
2. Extract `x_S`, the 32-byte big-endian X-coordinate of S.
3. Derive `Z_AB = SHA-256(x_S)` — the 256-bit AES-CCM key.

### 6.2.4 AES-256-CCM Parameters

| Parameter | Value |
|---|---|
| Algorithm | AES-256-CCM (NIST SP 800-38C). |
| Key | Z_AB (32 bytes). |
| Tag length | 16 bytes (128 bits). |
| Nonce/IV length | 12 bytes. |
| Length field size (q) | 3 bytes (messages up to 2^24 − 1 bytes). |
| AAD | Empty. |

The 12-byte IV is `00 00 00 00 00 00 00 01 ‖ Counter (4 bytes, big-endian)`, where Counter starts at `0x00000001` and increments after every encrypted message. Each party keeps its own counter; on rollover the session is invalidated. Encrypted payloads are carried in the Encrypted Data TLV (0x40) as ciphertext ‖ tag, and the tag **MUST** be validated before processing plaintext.

### 6.2.5 Flow Description

Steps marked (per-reader) apply only to the per-reader signing-key path. The legacy fallback is in Appendix B. The message sequence is shown in Section 8.

1. The Device has recognized the Reader and holds the site Trust Anchor (Section 6.2.1).
2. Device: sends Transaction Initiation (0x05); generates ephemeral EC keys.
3. Reader: generates ephemeral EC keys and sends Protocol Identifiers (0x0C), Reader compressed ephemeral PK (0x02), Reader Location Identifier (0x0D), Site Issuer Identifier (0x0E), and — per-reader — Reader Certificate (0x10).
4. Device: sends its uncompressed ephemeral PK (0x07) and Protocol Identifiers (0x0C); derives Z_AB; and — per-reader — verifies the Reader Certificate against the Site Issuer public key, matches the subject to TLV 0x0D, checks the validity window and revocation list, and extracts the Reader Public Key.

On any failure it sends error 0x07/0x08/0x09 and closes. It **MAY** use its cache to skip full re-verification (Section 7.2).

5. Reader: derives Z_AB; signs the 96-byte input with its Reader Signing Private Key (per-reader) or the legacy Site Private Key; sends Digital Signature (0x03).
6. Device: verifies the reader signature against the Reader Public Key from the certificate (per-reader) or the legacy Site public key. On failure it returns error 0x02 and closes. On success it signs the identical 96-byte input with the PKOC Private Key, AES-CCM-encrypts the PKOC public key (0x01), signature (0x03), and last-update time (0x09), and sends the Encrypted Data TLV (0x40); per-reader, it updates its certificate cache.
7. Reader: validates the CCM tag, decrypts, verifies the PKOC signature against the Device's PKOC public key, and returns the Response TLV (0x04).
8. Both parties **MUST** discard ephemeral keys and Z_AB immediately, ensuring Perfect Forward Secrecy. Long-term signing keys are unaffected.

### 6.3 Un-Obfuscated Flow

The Un-Obfuscated flow has no session encryption and is unchanged from earlier versions. The signature input is the Reader's 33-byte compressed ephemeral public key (TLV 0x02); the Device computes `ECDSA-SHA256(PKOC_Private_Key, SHA-256(reader_compressed_ephemeral_PK))`.

1. Device sends Transaction Initiation (0x05).
2. Reader generates an ephemeral EC key pair and sends its compressed ephemeral PK (0x02).
3. Device signs the reader's compressed ephemeral PK and sends PKOC public key (0x01), Digital Signature (0x03), and optional Last Update Time (0x09).
4. Reader verifies the signature against the PKOC public key; on success it evaluates access and sends Response (0x04); on failure it sends Error (0x06) and closes.

## 7. Per-Reader Signing Keys and Reader Certificates

The per-reader signing-key mechanism is the BLE realization of the Core's Validated trust model. Each reader holds a unique signing key; a Site Issuer signs a Reader Certificate binding that key to the reader's location; credentials trust one per-site Site Issuer public key. Reader enrollment is described in Appendix A.

### 7.1 Reader Certificate Format

A Reader Certificate is a fixed-offset, fixed-length 138-byte structure signed by the Site Issuer Private Key, stored in the reader's non-secret flash and transmitted as TLV 0x10.

| Offset | Length | Field | Description |
|---|---|---|---|
| 0 | 1 | Version | 0x01 for this profile. |
| 1 | 16 | Subject Reader Location Identifier | **MUST** match TLV 0x0D. |
| 17 | 16 | Issuer Site Issuer Identifier | **MUST** match TLV 0x0E. |
| 33 | 4 | Not-Before | Unix epoch seconds, big-endian (0 = no lower bound). |
| 37 | 4 | Not-After | Unix epoch seconds, big-endian (0xFFFFFFFF = no upper bound). |
| 41 | 33 | Reader Public Key | Compressed P-256. |
| 74 | 64 | Signature | Raw R ‖ S, ECDSA-SHA256 by the Site Issuer Private Key over offsets 0–73. |

The Site Issuer signs the first 74 bytes (SHA-256 then ECDSA on P-256); the 64-byte signature is placed at offset 74. On the wire the TLV header is `0x10 0x81 0x8A` (138-byte value), 141 bytes total. To verify, the Device checks the version, matches issuer and subject to TLV 0x0E/0x0D, validates the time window, verifies the signature over bytes 0–73 against the Site Issuer public key, and confirms the location is not revoked; on any failure it returns error 0x07/0x08/0x09.

### 7.2 Certificate Caching (Discovery-and-Pin)

A Device **MAY** cache verified Reader Public Keys to reduce per-transaction work: it discovers a reader's identity through full certificate verification on first encounter, then pins the reader's public key for fast verification on later encounters. Caching is **OPTIONAL**.

| Cache field | Length | Description |
|---|---|---|
| Reader Location Identifier | 16 bytes | Cache key (from certificate subject). |
| Reader Public Key | 33 bytes | Compressed P-256, from the certificate. |
| Site Issuer Identifier | 16 bytes | Anchor the entry was verified against. |
| Certificate Fingerprint | 32 bytes | SHA-256 over the 138-byte certificate; detects replacement. |
| Not-After | 4 bytes | Entry expiry, copied from the certificate. |

On a hit — fingerprint matches, issuer matches, Not-After is in the future, and the location is not revoked — the Device **MAY** skip full signature verification and use the pinned key. Any mismatch (especially a fingerprint change from a replaced reader) forces full re-verification and a cache update. Devices **SHOULD** cache at least 256 readers (LRU eviction). The cache is a performance optimization, not a trust mechanism: an attacker cannot forge a certificate that both matches a legitimate fingerprint and verifies against the Site Issuer key.

### 7.3 Reader Certificate Revocation

An operator may revoke a reader before its certificate expires. Revocation is a Site Issuer-signed list distributed over the provisioning channel (Core Section 6).

| Field | Length | Description |
|---|---|---|
| Version | 1 byte | 0x01. |
| Issuer Identifier | 16 bytes | Site Issuer that authored the list. |
| Issued Time | 4 bytes | Unix epoch seconds, big-endian. |
| Entry Count | 2 bytes | Number of entries, big-endian. |
| Entries | 20 bytes each | 16-byte Reader Location Identifier + 4-byte revocation timestamp. |
| Signature | 64 bytes | Raw R ‖ S, ECDSA-SHA256 by the Site Issuer over all preceding fields. |

Devices **SHOULD** refresh the list periodically (24 hours recommended). Before honoring a list the Device **MUST** verify its signature and **MUST** accept it only if its Issued Time exceeds the cached list's (rollback protection). During every per-reader transaction the Device **MUST** check the Reader Location Identifier against the cached list and, if present, send error 0x08 and close. At 20 bytes per entry, a 1,000-entry list is roughly 20 KB — easily shipped over the provisioning channel.

## 8. Sequence Diagrams

The following message-level sequences correspond to the flows in Section 6.

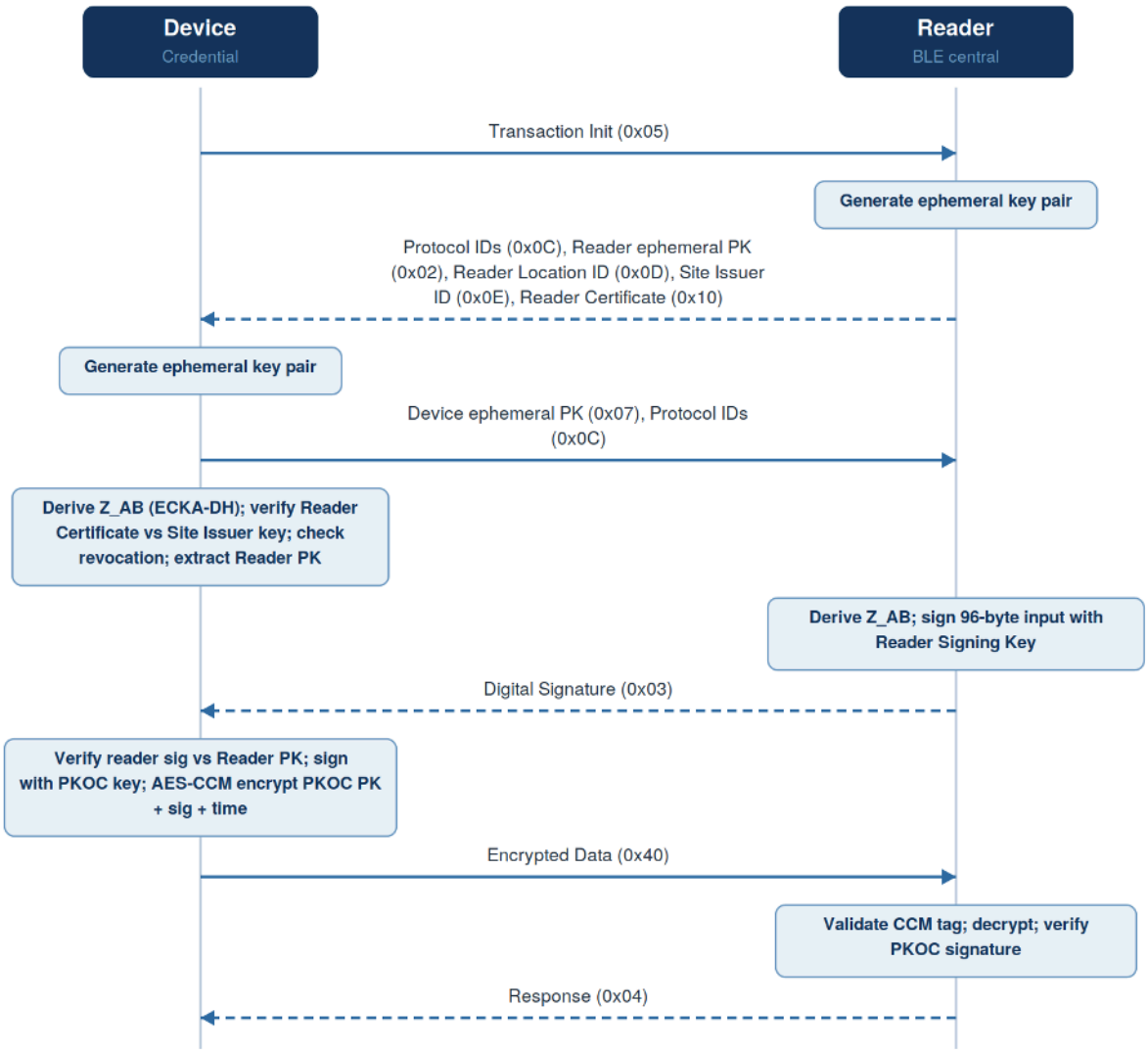


*Figure 8-1. ECDHE Perfect Forward Secrecy flow (per-reader signing-key path).*

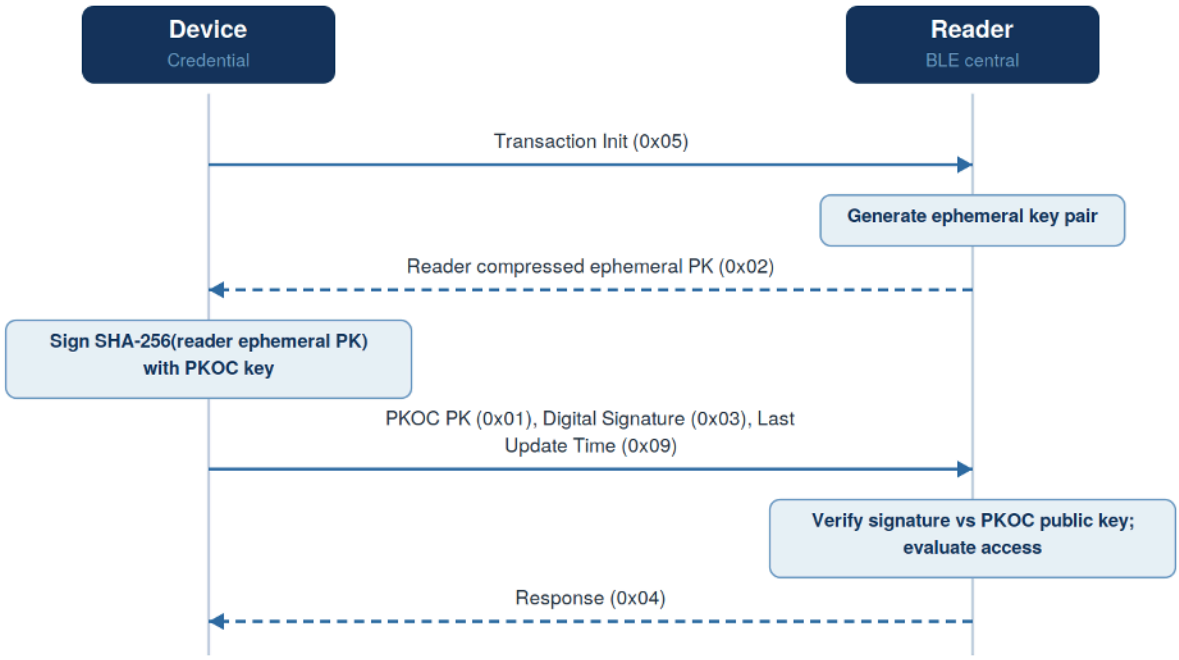


*Figure 8-2. Un-Obfuscated flow.*

> **Note:** The legacy (3.1.1) ECDHE sequence is identical to Figure 8-1 except that the Reader Certificate (0x10) is not sent, TLV 0x0E carries the Site Key identifier, and the reader signs with the shared Site Private Key while the Device verifies against the legacy Site public key. See Appendix B.

## Appendix A. Operational Phases for Per-Reader Signing Keys

This appendix is informative. It describes the deployment pipeline that places a unique Reader Signing Key and a Site Issuer-signed Reader Certificate into each reader, producing the preconditions for the per-reader path in Section 6.2. The pipeline has four phases, each performed by the party naturally positioned to do it.

| Phase | Performed by | When |
|---|---|---|
| 1. Factory Provisioning | Reader Manufacturer | At manufacture, per reader. |
| 2. Site Operator CA Setup | PACS / Site Operator | Once per site. |
| 3. On-Site Enrollment | Installer (authenticated to PACS) | At installation, per reader. |
| 4. Steady-State Operation | Device, Reader, PACS | Every transaction. |

### A.1 Key Separation

- The Reader Signing Private Key never leaves the reader's secure element.
- The Site Issuer Private Key never leaves the PACS backend HSM.
- The PKOC Private Key never leaves the credential's secure element.
- The manufacturer sees no site-specific information; per-reader factory output is identical regardless of customer.

## A.2 Phase 1 — Factory Provisioning

Per reader on the production line, the secure element generates an internal P-256 key pair and exports the public key plus a vendor-signed hardware-attestation blob proving the key is non-exportable. Firmware writes both to non-secret flash; the reader is labeled with a serial (QR/NFC) encoding the serial and a public-key fingerprint. The manufacturer publishes a signed attestation manifest of {serial, public key, attestation chain} tuples. Per-reader cost is a few seconds; no site keys touch the factory.

## A.3 Phase 2 — Site Operator CA Setup

Once per site, the operator generates the Site Issuer key pair in an HSM (the private key never leaves), configures a CA service to sign Reader Certificates from authenticated installer tools, publishes the Site Issuer public key through the credential-provisioning channel, and establishes a revocation-list publication channel. Typical integration is about a week of PACS-vendor effort.

## A.4 Phase 3 — On-Site Enrollment

At install, the binding between the manufacturer-provided reader key and the site operator's authority is created. Two patterns are supported.

### A.4.1 Tablet-Assisted Enrollment

1. Installer mounts and powers the reader, then taps (NFC) or scans (QR) it with a PACS-authenticated tablet.
2. The tablet reads {serial, Reader Public Key, attestation blob} and submits them with installer identity and the intended Reader Location Identifier to the PACS backend over TLS.
3. The backend confirms the serial and public key against the manufacturer manifest, validates the attestation chain, and confirms installer authorization for the location.
4. The Site Issuer (in the HSM) signs a Reader Certificate (Section 7.1) and returns it.
5. The tablet writes the certificate to the reader; the reader self-tests (signs a value and verifies against its certificate) and stores it. Failure aborts enrollment.

End-to-end this is roughly 15–30 seconds per reader — small relative to mounting and wiring. It applies to all form factors, including battery-powered readers without network connectivity.

### A.4.2 Network Self-Enrollment

For network-connected readers (PoE/Wi-Fi), the reader boots, locates the PACS enrollment endpoint, and submits {serial, Reader Public Key, attestation blob, MAC}. The backend matches a pre-staged enrollment record, validates attestation, signs a Reader Certificate, and returns it; the reader self-tests and stores it. This removes the tablet step but requires network coordination ahead of the install.

## A.5 Phase 4 — Steady-State Operation

After enrollment the reader operates without further CA contact. Every transaction follows Section 6.2: the reader presents its certificate and signs with its Reader Signing Private Key; the Device verifies against the Site Issuer public key and pins the reader key (Section 7.2). The Site Issuer is offline except for enrollments, replacements, and revocation publications — the trust root for thousands of readers is never exposed to runtime traffic.

## A.6 Reader Replacement

A replacement reader has a different key pair. It is enrolled exactly as in A.4, receiving a new certificate for the same Reader Location Identifier; the failed reader's location is added to the revocation list. On next encounter the credential's cached fingerprint no longer matches, so it performs full re-verification (which

succeeds because the new certificate is Site Issuer-signed) and updates its cache. The user observes no change.

### A.7 Multi-Vendor Reader Fleets

A site may mix manufacturers. Each manufacturer publishes its own attestation manifest under its own chip-vendor root; the PACS trusts each root only at enrollment time. The Site Issuer signs certificates for all readers regardless of make, so every reader chains to the same single Site Issuer and the credential sees exactly one Trust Anchor per site. An operator can blacklist a suspect manufacturer root without affecting already-enrolled readers.

### A.8 Site Issuer Key Rotation

Periodic rotation (e.g., every five years, or on suspected compromise) re-issues all Reader Certificates and re-provisions the new Site Issuer public key to credentials. During the transition credentials hold both old and new anchors. This is the only per-reader operation whose cost scales with the number of credentials, but it is a scheduled maintenance event rather than an emergency: individual reader compromise is handled by revocation, not rotation.

## Appendix B. Backward Compatibility with Legacy BLE (3.1.1)

This appendix specifies interoperation between per-reader (this profile) and legacy shared-Site-Key (3.1.1) implementations during transitions.

### B.1 Version Negotiation

Negotiation uses the Protocol Identifiers TLV (Section 5.6). The Spec Version bytes are `0x03 0x01` for all 3.1.x revisions; the per-reader capability is signaled solely by Feature Bits Low bit 1 (Reader Certificate support).

| Reader | Device | Operative path |
|---|---|---|
| Per-reader (bit set) | Per-reader (bit set) | Per-reader ECDHE signing-key path. |
| Per-reader (bit set) | Legacy (bit clear) | Reader falls back to the legacy path (B.3). |
| Legacy (bit clear) | Per-reader (bit set) | Device falls back to legacy verification if a legacy Site public key is provisioned (B.2). |
| Legacy | Legacy | Legacy shared-Site-Key ECDHE (unchanged). |

### B.2 Device Fallback (Per-Reader Device, Legacy Reader)

1. The Device confirms the reader did not set bit 1 and no Reader Certificate (0x10) was sent.
2. The Device looks up the legacy Site public key for the Site Issuer Identifier (TLV 0x0E).
3. If present, it verifies the reader signature directly against the legacy key (the 3.1.1 path) and completes normally.
4. If absent, it sends error 0x04 (Unknown Site Issuer) and closes.

For mixed deployments, credentials should hold both the Site Issuer public key and the legacy Site public key for each site, keyed by the same Site Issuer Identifier.

### B.3 Reader Fallback (Per-Reader Reader, Legacy Device)

- **Policy A (recommended for high-security sites):** the reader rejects legacy devices outright, requiring all devices to be updated first.
- **Policy B (transition mode):** the reader holds both a Reader Signing Key and a copy of the legacy Site Private Key; on detecting a legacy device it signs with the legacy key and omits the certificate. This reintroduces the legacy blast radius for the transition window and **SHOULD** be minimized. Once the fleet is migrated, the legacy Site Private Key **SHOULD** be wiped from all readers.

### B.4 Sample Code Compatibility

Legacy sample code remains correct for the Un-Obfuscated flow and for implementations operating in legacy ECDHE mode. Per-reader ECDHE with certificate verification is illustrated in a separate PSIA implementation guide.

## Appendix C. Sample Code

Reference samples for the Un-Obfuscated flow — key generation in the Android Keystore and iOS Secure Enclave, DER-to-raw conversion, TLV encoding, and BLE transmission (CoreBluetooth / Android BLE) — are published with prior PKOC BLE releases and remain applicable. Samples for the per-reader ECDHE flow with Reader Certificate verification are published as a separate PSIA implementation guide; the normative behavior is the specification text in Sections 6.2 and 7.1, which governs over any sample.

## References

This profile builds on the references in the PKOC Core Specification and adds:

- **BSI TR-03111** — Elliptic Curve Cryptography (ECKA-DH key agreement).
- **NIST SP 800-38C** — CCM Mode for Authentication and Confidentiality.
- **Bluetooth Core Specification** — GATT, ATT, and BLE link layer.
- **PKOC Core Specification**, Version 2.0.1 — PSIA.

*End of PKOC BLE Transport Profile, Version 2.0.1.*