PKOC SPECIFICATION

PKOC NFC Transport Profile

Public Key Open Credential for NFC access cards

Version: 2.0.1

Status / Classification: Public — Working Draft, not approved for implementation

Date: August 13, 2026

Maintained by: Physical Security Interoperability Alliance (PSIA)

Supersedes: PKOC NFC Card Specification v1.1

Document Revision History

Version	Date	Author	Description
1.1	Dec 8, 2023	PSIA SCI Working Group	PKOC NFC Card Specification v1.1: the two-command Standard Mode protocol (SELECT + AUTHENTICATE).
2.0.1	Aug 13, 2026	PSIA SCI Working Group (PKOC-CVC and Validated Mode contributed by Taglio LLC; EV-P Profile command parameters contributed by Mike Zercher, INID Readers USA)	Introduced PKOC Validated Mode, the SE V2 and EV-P card profiles, and reorganized NFC operation around three card profiles with common transaction processes separated from profile-specific commands. Added an explicit statement that Standard Mode alone satisfies PKOC conformance and that PKOC conformance is vendor-neutral. Fully backward compatible with v1.1.

Table of Contents

Glossary

This glossary lists terms specific to the NFC transport profile. General PKOC terms (Credential, Reader, PACS, Trust Anchor, ECDSA, Derived Identifier, and so on) are defined in the PKOC Core Specification and are not repeated here.

Term	Definition
AID	Application Identifier. The baseline PKOC application AID is `A0 00 00 08 98 00 00 01`. Additional profiles may define another application AID in the applicable profile section.
APDU	Application Protocol Data Unit; the ISO/IEC 7816-4 command/response unit exchanged between Reader and Card.
BER-TLV	Basic Encoding Rules Tag-Length-Value; the TLV encoding used for NFC command and response data.
Card	The NFC smart card or device acting as the PKOC Credential.
Card Issuer	The entity that provisions Cards and signs PKOC-CVCs; the NFC realization of the Core's Provisioning / Issuer Backend.
Card Profile	The specification detail for a specific card type (SE V1, SE V2, or EV-P).
EV-Capable Card	A Card that supports the EV APDU commands defined in VDE-AR-E 2532-100 (VDEReadData, VDEECDSASign).
EV-P Profile	The card profile that runs PKOC over an EV-capable card personalized for PKOC. It uses the PKOC AID, a PKOC-CVC in File 0, and NIST P-256, with the two EV APDU commands (VDEReadData, VDEECDSASign). It supports Standard Mode and may support Validated Mode.
IIR	Issuer Identification Reference; a 16-character identifier embedded in a PKOC-CVC that names the Issuer Key used to sign it and the JWK `kid` used to locate that key.
INTERNAL AUTHENTICATE	The SE V2 APDU that asks the Card to sign a 32-byte Reader challenge using the private key corresponding to the PKOC public key in the PKOC-CVC.
JWKS	JSON Web Key Set (RFC 7517); the format in which public Issuer Keys are published.
NFC Transport Profile	This specification, in contrast to other Transport Profiles such as BLE.
PKOC-CVC	PKOC Card Verifiable Certificate; the baseline card-bound attestation certificate defined in Core Section 5. It carries the PKOC public key and may carry extension credentials.
SE V1 Profile	The v1.1-compatible NFC card profile using SELECT and AUTHENTICATE. It supports Standard Mode only.
SE V2 Profile	The card profile identified by a successful GET DATA (INFO) response of `02 00`. It retrieves the PKOC-CVC with GET DATA and authenticates using INTERNAL AUTHENTICATE. It supports Standard Mode and Validated Mode.
Standard Mode	The operating mode in which proof of possession is verified and the PKOC public key, PKOC Credential, or PKOC Derived Identifier is trusted through prior PACS enrollment. PKOC-CVC is not required.
Validated Mode	The operating mode in which the Reader performs the complete Validation process using a PKOC-CVC before accepting authentication or outputting a credential.

Term	Definition
VDEReadData	The EV-P Profile command used to read File 0, which contains the PKOC-CVC.
VDEECDSASign	The EV-P Profile command used to sign a Reader challenge.
Wiegand	A legacy PACS signaling format; PKOC-CVC extension credentials allow a Reader to emit legacy identifiers for Wiegand or OSDP panels.

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 conformance does not require a PKOC-CVC. Other methods of establishing trust **MAY** be used in Standard Mode but are outside the scope of this specification.

Validated Mode uses a PKOC-CVC and the complete Validation process defined in Section 3.4.

Standard Mode alone is sufficient for PKOC conformance. Validated Mode, the PKOC-CVC, Issuer Keys, Trust Anchors, and PKOC-CVC extension credentials are **OPTIONAL** capabilities. A Card or Reader that implements only Standard Mode — including an unmodified PKOC NFC v1.1 Card or Reader — is fully PKOC conformant. Where this document states a **MUST**, **MUST NOT**, **SHALL**, or **SHALL NOT** in connection with a PKOC-CVC, Validated Mode, an Issuer Key, or a Trust Anchor, that requirement applies only to an implementation that has elected to support the optional capability in question. Such a requirement is conditional on that election 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 Card, Reader, or Issuer — the application AID, the command set, the certificate format, the trust model, and the credential derivation — is specified in this document and the Core, and **MAY** be implemented by any party without licence, royalty, registration, or approval. A conforming Reader configured for a given profile **MUST** accept any Card that meets that profile's requirements, regardless of its manufacturer, applet supplier, card operating system, or Issuer.

1. Introduction

1.1 Relationship to the PKOC Core Specification

This document is a Transport Profile of the PKOC Core Specification. It defines the PKOC NFC protocol for physical-access Readers and Cards operating within a PACS. Transport-independent concepts — including the PKOC credential model, cryptographic baseline, key encodings, PKOC-CVC format, registration, and common Credential and 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 NFC Transport Profile

This profile defines three card profiles — SE V1, SE V2, and EV-P — and two operating modes — Standard Mode and Validated Mode. It specifies baseline application detection, common high-level processes, profile-specific command mappings, Card and Reader provisioning, authentication, validation, credential output, Issuer-key management, and NFC-specific requirements. The SE V1 Profile is backward compatible with PKOC NFC v1.1.

1.3 Changes in This Version

Version 2.0.1 reorganizes NFC operation around three card profiles and separates the common transaction processes from profile-specific commands. Section 6 defines baseline application detection. The SE V1 Profile preserves the v1.1 command flow, the SE V2 Profile adds the V2 functions, and the EV-P Profile uses standards-based EV functions. Standard Mode and Validated Mode remain independent of the selected profile. Backward compatibility and vendor neutrality are stated in Section 1.4.

Additional card profiles, including a profile for cards pre-provisioned for electromobility, are expected in a future revision. Section 2.2 and Core Section 8.2 define how a profile is added; adding one requires no change to the profiles defined here.

1.4 Backward Compatibility and Vendor Neutrality

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

- **PKOC NFC Card Specification v1.1.** The SE V1 Profile (Section 7) is the v1.1 protocol, unchanged: the same PKOC application AID, the same SELECT command and response, the same AUTHENTICATE command and response TLVs, the same NIST P-256 baseline, and the same credential derivation. A v1.1 Card and a v1.1 Reader remain fully conformant to this version without modification, and a Reader built to this version interoperates with a v1.1 Card exactly as a v1.1 Reader does.
- **PKOC BLE.** This profile makes no change of any kind to the PKOC BLE Transport Profile or to the PKOC BLE Specification v3.1.2 that it supersedes. The BLE wire protocol is unchanged and byte-compatible; no BLE implementation is affected by anything in this document.
- **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 v1.1, on every profile and every transport. Enrolled credential records do not need to be re-derived, re-issued, or re-enrolled.
- **Additive only.** The SE V2 Profile, the EV-P Profile, Validated Mode, the PKOC-CVC, Issuer Keys, and extension credentials are additions. They are elected through Reader configuration and Card personalization and impose no obligation on an implementation that does not use them.

PKOC is an open, royalty-free standard, and this profile is written so that it stays that way. Every element of it **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 applet, chip, card operating system, certificate hierarchy, key-distribution service, registry, or

intellectual property. Any organization **MAY** act as a Card Issuer and operate its own Issuer Keys; PKOC defines no central issuing authority and no approval process.

2. NFC Architecture

A card profile defines the Card application and the functions used to obtain the PKOC public key, authenticate the Card, and produce a Reader-to-PACS credential. Section 6 defines baseline application detection, and Section 3 defines the common transaction processes. Each profile section maps those processes to its commands and **MAY** define profile-specific detection or public-key retrieval behavior.

Other standards-based profile implementations **MAY** be supported when the Reader-to-PACS output conforms to the PKOC Credential and PKOC Derived Identifier rules in Core Section 4. A profile that does not support a PKOC-CVC **MUST** document how the PKOC public key is retrieved. Any such profile **MUST** be implementable by any party from published specifications alone and **MUST NOT** make PKOC conformance dependent on a vendor-controlled applet, key, certificate hierarchy, or key-distribution service.

2.1 Operating Modes

Mode	Conformance	Trust establishment
Standard Mode	Required	The Reader verifies proof of possession of the PKOC private key. The PKOC public key, PKOC Credential, or PKOC Derived Identifier is trusted through prior PACS enrollment. A PKOC-CVC is not required. Other methods of establishing trust **MAY** be used but are outside the scope of this specification.
Validated Mode	Optional	The Reader performs the complete PKOC-CVC Validation process before accepting authentication or outputting a credential.

In both operating modes, the Reader **MUST** verify the Card challenge signature using the PKOC public key obtained through the selected profile.

A Reader operating in Validated Mode **MUST** successfully complete the Validation process in Section 3.4 and **MUST NOT** fall back to Standard Mode after validation fails.

2.2 Card Profiles

The NFC Transport Profile supports multiple card profiles.

Profile	Application detection	Supported modes	Description
SE V1	Baseline application detection in Section 6.	Standard	The PKOC public key and challenge signature are returned together by the profile Authentication function.
SE V2	Baseline application detection in Section 6, including capability information.	Standard and Validated	The certificate containing the PKOC public key is retrieved separately from the Authentication function.
EV-P	Baseline PKOC application (Section 6.1); an EV-P card returns `9000` with no version data.	Standard; Validated when supported	The two EV APDU commands (VDEReadData, VDEECDSASign) are used with the PKOC AID and a PKOC-CVC in File 0 to retrieve the public key and authenticate the Card, using NIST P-256.

An implementation **MAY** support more than one card profile. Future revisions **MAY** define additional profiles consistent with this architecture.

2.3 Credential Types

In Standard Mode, the Reader **MUST** output a PKOC Credential or PKOC Derived Identifier derived in accordance with Core Section 4.

In Validated Mode, the Reader **MUST** output a PKOC Validated Credential, PKOC Validated Derived Identifier, or PKOC-CVC extension credential, according to Reader configuration. A PKOC-CVC extension credential is an issuer-provided value carried in Certificate Extension Data and can represent a legacy PACS credential such as an ISO/IEC 14443 UID or a Wiegand-compatible identifier.

A PKOC Validated Derived Identifier **SHOULD** have a length of 8 to 31 octets (64 to 248 bits). It **MAY** be 4 to 7 octets (32 to 56 bits) when the issuer attests its uniqueness for the IIR of the PKOC-CVC, as defined in Core Section 4.6.

Profile or key use	Supported algorithms
SE V1 PKOC key	PKOC cryptographic baseline defined in Core Section 3: NIST P-256 with SHA-256.
SE V2 PKOC key	ECC on a supported curve, including P-256, P-384, or Brainpool; or RSA 2048, 3072, or 4096.
EV-P PKOC key	NIST P-256 (secp256r1), aligning with the SE V1 / PKOC V1 baseline.
PKOC-CVC Issuer key	Supported ECC curves, including P-256, P-384, P-521, or Brainpool; RSA 2048, 3072, or 4096; or ML-DSA-44, ML-DSA-65, or ML-DSA-87.

A Reader **MUST** reject a transaction when it does not support the PKOC public-key or Issuer-key algorithm required by that transaction.

3. High-Level Process Overview

The common processes are Profile Detection, Get PKOC-CVC when applicable, Validation in Validated Mode, Authentication, and Reader to PACS. Profiles **SHOULD** perform the processes in the order shown; a profile **MAY** define another order. The selected profile defines the commands and functions used to perform each process.

3.1 Profile and Mode Flows

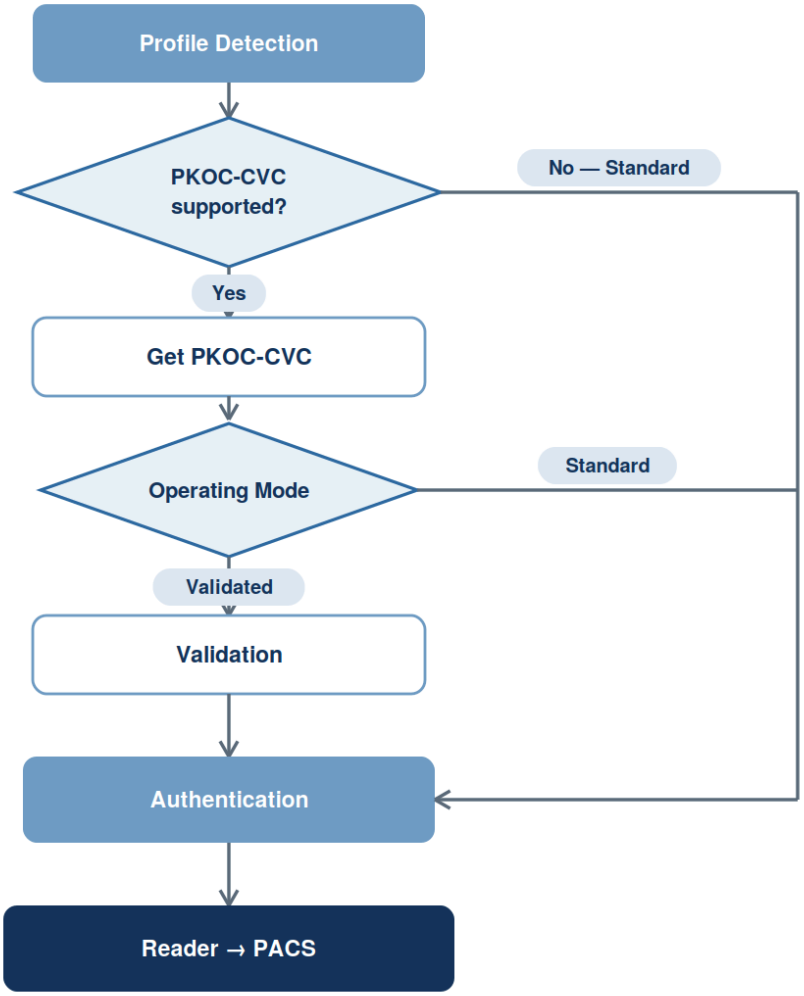

Figure 3-1. Common NFC transaction processes.

3.2 Profile Detection

The Reader **MUST** select a profile permitted by its configuration. Profile Detection **MUST** use the baseline application detection in Section 6 and **MAY** also use additional detection defined by the selected profile. The selected profile determines how the remaining processes are performed.

3.3 Get PKOC-CVC

The Get PKOC-CVC process obtains a PKOC-CVC containing the PKOC public key. In Standard Mode, a profile **MAY** use this process to retrieve the public key without validating the PKOC-CVC. In Validated Mode, the Reader performs the Get PKOC-CVC process followed by the Validation process.

A profile that does not use a PKOC-CVC in Standard Mode defines its public-key retrieval method in the profile section.

3.4 Validation

In Validated Mode, the Reader **MUST** validate the PKOC-CVC. Validation **SHOULD** be completed before the Authentication process; a profile **MAY** define another process order.

The Reader **MUST** fail the transaction if a required PKOC-CVC check, or any optional check it elects to perform, fails. The Reader **MUST** extract the IIR from the PKOC-CVC and select the configured Issuer Key whose identifier matches the IIR. If no matching Issuer Key is configured, the Reader **MUST** fail the transaction. The Reader **MUST** verify the PKOC-CVC signature using the selected Issuer Key. The Reader **SHOULD** verify that the PKOC-CVC conforms to the certificate profile in Core Section 5, including required fields, field order, and encoding. The Reader **MAY** verify that the current date is within the PKOC-CVC validity period.

3.5 Authentication

The Authentication process applies to every profile. The Reader **MUST** send a fresh challenge, the Card **MUST** sign that challenge using the private key corresponding to the transaction public key, and the Reader **MUST** verify the returned signature before deriving or outputting a credential. The selected profile defines how the process is implemented.

3.6 Reader to PACS

The Reader **MUST** derive credentials in accordance with Core Section 4. In Standard Mode, it **MUST** output a PKOC Credential or PKOC Derived Identifier. In Validated Mode, it **MUST** output a PKOC Validated Credential, PKOC Validated Derived Identifier, or PKOC-CVC extension credential, according to configuration.

When configured to output a PKOC-CVC extension credential, the Reader **MUST** locate the extension matching the configured OID and **MUST** fail the transaction if that OID is absent. Reader-to-PACS signaling and output encoding, including Wiegand or OSDP, are outside the scope of this profile.

4. NFC Common Requirements

These requirements supplement the common Credential, Reader, and cryptographic requirements in the PKOC Core Specification. A conforming NFC implementation **MUST** satisfy both. All profiles **MUST** implement these requirements.

4.1 Card Requirements

1. The Card **MUST** sign the Reader-supplied challenge using the private key corresponding to the PKOC public key used for the transaction.
2. The Card **MUST** return the signature using the encoding required by the selected profile and algorithm.
3. The Card **MUST NOT** transmit the PKOC private key.

4.2 Reader Requirements

1. The Reader **MUST** be configured with the accepted card profiles and operating modes.
2. The Reader **MUST** generate a fresh, unpredictable challenge for every authentication transaction.
3. The Reader **MUST** verify the Card signature using the PKOC public key obtained through the selected profile before deriving or outputting a credential.

4.3 Validated Mode Reader Requirements

Validated Mode is optional. These requirements apply only to a Reader that supports it; a Reader that supports only Standard Mode is fully conformant and is not subject to this section.

1. A Reader operating in Validated Mode **MUST** be provisioned with the Issuer Keys required to validate supported PKOC-CVCs.

2. The Reader **MUST** complete the Validation process before accepting the Authentication result and **MUST NOT** fall back to Standard Mode after validation fails.
3. When configured to output a PKOC-CVC extension credential, the Reader **MUST** implement the requirements in Sections 3.6 and 5.4.

4.4 Cryptographic Requirements

The Card and Reader **MUST** satisfy the cryptographic requirements in the Core and in the selected profile.

5. Reader and Card Provisioning

This section defines NFC-specific provisioning. The general registration process is defined in Core Section 6.

5.1 PKOC Key Pairs

A Card **MUST** be provisioned with the key pair or key pairs required by each supported profile. Profile-specific key provisioning requirements are defined in Sections 7 through 9.

5.2 PKOC-CVC

A Card supporting Validated Mode **MUST** be provisioned with a PKOC-CVC compliant with Core Section 5 and signed by a Card Issuer using a supported Issuer algorithm. The selected profile **MAY** override the baseline PKOC-CVC retrieval command and, when it does, **MUST** define how the PKOC-CVC is stored and retrieved.

A PKOC-CVC is **OPTIONAL** for SE Profiles. A SE Card that does not support Validated Mode is not required to carry one and is fully PKOC conformant without one. Any organization **MAY** act as a Card Issuer and sign PKOC-CVCs with its own Issuer Key; no license, registration, central authority, or vendor approval is required.

5.3 Issuer Keys

A Reader supporting Validated Mode **MUST** be provisioned with the Issuer Keys required for the Cards it supports. Each Issuer Key **MUST** be identified by the IIR used in the PKOC-CVC. The Reader **MUST** provide a mechanism to add, replace, and remove Issuer Keys. The provisioning and update mechanism **MAY** be defined by the deployment and is otherwise outside the scope of this profile.

5.4 PKOC-CVC Credential Types

A Reader supporting Validated Mode **MUST** support the following PKOC-CVC extension credential types defined in Core Section 5.7. If the Reader is configured to output one of these types and the corresponding OID is absent, the Reader **MUST** fail the transaction.

Name	OID	DER type	Length
UUID	1.3.6.1.4.1.59685.8.1	OCTET STRING	16 bytes
4ByteUID	1.3.6.1.4.1.59685.8.2	OCTET STRING	4 bytes
7ByteUID	1.3.6.1.4.1.59685.8.3	OCTET STRING	7 bytes
10ByteUID	1.3.6.1.4.1.59685.8.5	OCTET STRING	10 bytes
BinaryID	1.3.6.1.4.1.59685.8.7	BIT STRING	Variable

5.5 Reader Configuration

A Reader **MUST** be configured with the following settings. How configuration is stored is outside the scope of this profile.

Configuration item	Conformance	Description
Accepted card profiles	**MUST** (always)	SE V1, SE V2, EV-P, or an allowed combination.
Accepted PKOC Operating Modes	**MUST** (always)	Standard, Validated, or both, subject to the selected profile.
Profile Application AID	**MUST** when the deployment supports more than one AID for a profile	The AID used to select the configured profile.
Reader Output Credential Type	**MUST** (always)	The credential type output in the Reader-to-PACS process.
CVC Credential OID	**MUST** when output type is a PKOC-CVC extension credential	The OID of the extension credential defined in Section 5.4.
Identifier Bit Length	**MUST** when output type is a PKOC Derived Identifier or PKOC Validated Derived Identifier	The bit length of the identifier to output.

6. Profile Detection Commands

This section defines profile detection commands. Supported applications **MUST** respond to SELECT. The baseline PKOC application uses the SELECT command in Section 6.1 and GET DATA (INFO) in Section 6.2. Other profiles **MAY** define another application AID and **MAY** define additional detection functionality.

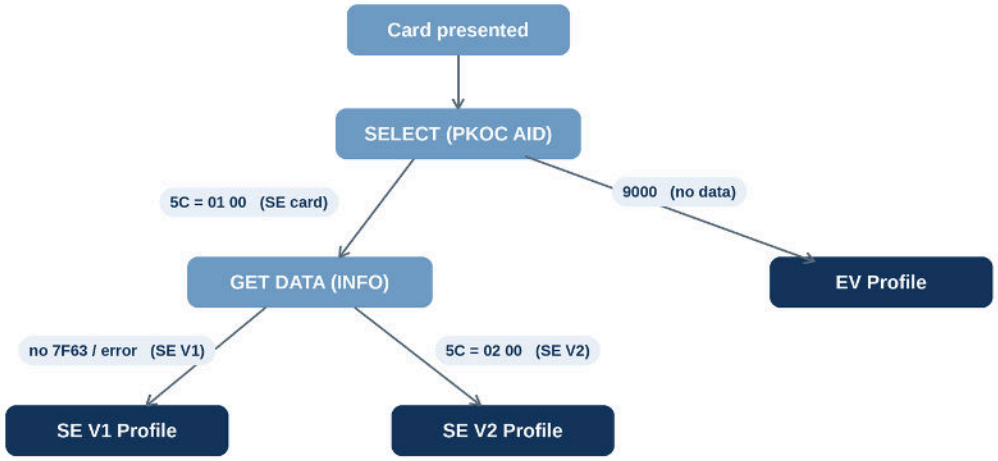

Figure 6-1. Baseline application detection and profile selection.

6.1 SELECT Application

The following APDU selects the baseline PKOC application.

Field	Value	Description
CLA	00	Command class.
INS	A4	SELECT.

Field	Value	Description
P1	04	Select by DF name (AID).
P2	00	First or only occurrence.
Lc	08	Length of data field.
Data	A0 00 00 08 98 00 00 01	PKOC Application Identifier.
Le	00	Maximum expected response length.

The following table describes the required response for the different card profiles on a successful select (Status word 9000).

Profile	Select Application Response Data	GET DATA (INFO)
SE V1	**MUST** contain tag `5C` with protocol version `01 00`	**MAY** continue to GET DATA (INFO)
SE V2	**MUST** contain tag `5C` with protocol version `01 00`	**MUST** continue to GET DATA (INFO)
EV-P	**MUST NOT** contain tag `5C` with protocol version	**MAY** continue to GET DATA (INFO)

Note: The SELECT command and its `01 00` response are unchanged from PKOC NFC v1.1.

Status words include `9000` success, `6700` wrong length, `6B00` incorrect P1-P2, `6D00` instruction not supported, `6E00` invalid CLA, and `6F00` general error.

6.2 GET DATA (INFO)

GET DATA (INFO) is part of baseline application detection and queries PKOC protocol capability information. It does not select the operating mode. This command **MUST** be used to confirm support for Profile SE V2 after a successful Select Application command.

Field	Value	Description
CLA	00	Command class.
INS	CA	GET DATA.
P1 / P2	7F / 63	Tag 7F63 — PKOC Protocol Info object.
Le	00	Maximum expected response length.

Status words include `9000`, `6A88`, `6D00`, `6E00`, and `6A86`.

The response **MUST** be a constructed `7F63` TLV containing inner tag `5C` with value `02 00`. The `7F63` object **MAY** contain additional TLVs. The Reader **MUST** ignore unrecognized elements.

The following table describes the response codes and actions for the different card profiles.

Response Code	Action
9000	The cards **SHOULD** proceed with the SE V2 profile.
6D00 or 6E00	The card does not support the SE V2 profile and **SHOULD** proceed with the SE V1 profile
Any other response code	The card **MUST** terminate the transaction

Example response: `7F 63 04 5C 02 02 00` followed by `9000`. Status words include `9000`, `6A88`, `6D00`, `6E00`, and `6A86`.

An error status word from GET DATA (INFO) **MUST NOT** cause the Reader to abort the transaction or re-issue SELECT. A Card **MUST** remain selected after responding to an unsupported GET DATA (INFO), so that a PKOC NFC v1.1 Card is unaffected by profile detection.

Example response: `7F 63 04 5C 02 02 00` followed by `9000`.

7. PKOC SE V1 Profile Commands

The SE V1 Profile supports Standard Mode only. Profile Detection uses the baseline application detection in Section 6; GET DATA (INFO) is not used. The following table maps the common processes to the SE V1 functions.

Process	SE V1 implementation
Profile Detection	Baseline application detection in Section 6.
Get PKOC-CVC	Not used. The PKOC public key is returned by the Authentication process.
Validation	Not supported by this profile.
Authentication	**AUTHENTICATE** (profile-specific).
Reader to PACS	Credential derivation according to Core Section 4.

Profile requirements

- The Card **MUST** return the baseline SELECT response defined in Section 6.1.
- The Card **MUST** implement the PKOC cryptographic baseline defined in Core Section 3 and **MUST** be provisioned with a conforming Standard-mode key pair.
- The Card **MUST** implement AUTHENTICATE as defined in Section 7.1.
- The Reader **MUST** perform baseline application detection as defined in Section 6. A Reader that accepts both SE profiles uses GET DATA (INFO) to identify the Card as SE V1 (Section 6.2); once the SE V1 Profile has been selected, the Reader **MUST NOT** use GET DATA (INFO) as part of the SE V1 transaction.
- The Reader **MUST** send AUTHENTICATE and **MUST** verify the response using the public key returned in tag `5A`.
- The Reader **MUST** implement the PKOC cryptographic baseline defined in Core Section 3.

7.1 AUTHENTICATE

AUTHENTICATE requests the Card to compute an ECDSA signature over the Reader-supplied Transaction Identifier using the SE V1 private key and to return the signature and public key.

Field	Value	Description
CLA	80	Manufacturer-specific class.
INS	80	AUTHENTICATE.
P1 / P2	00 / 01	Fixed.
Lc	38	Length of data field.
Data	5C (version) + 4C (Transaction ID) + 4D (Reader ID)	Protocol Version `01 00`; 16-byte Transaction Identifier; 32-byte Reader Identifier. The Reader Identifier is not included in the signature and **MAY** be zero.

Field	Value	Description
Le	00	Maximum expected response length.

The Reader **MUST** generate a fresh, unpredictable 16-byte Transaction Identifier for each authentication session. The Card **MUST** sign the value of tag `4C`, excluding the tag and length octets.

Tag	Length	Description
5A	65	Uncompressed P-256 public key encoded as `04 ‖ X ‖ Y`.
9E	64	ECDSA signature over the Transaction Identifier encoded as raw `R ‖ S`.

The signature **MUST** conform to the Core Section 3 baseline. The Reader **MUST** parse the response TLVs, **MUST** accept them in any order, and **MUST** ignore unrecognized TLVs.

7.2 Sequence Diagram for SE V1 Standard Mode

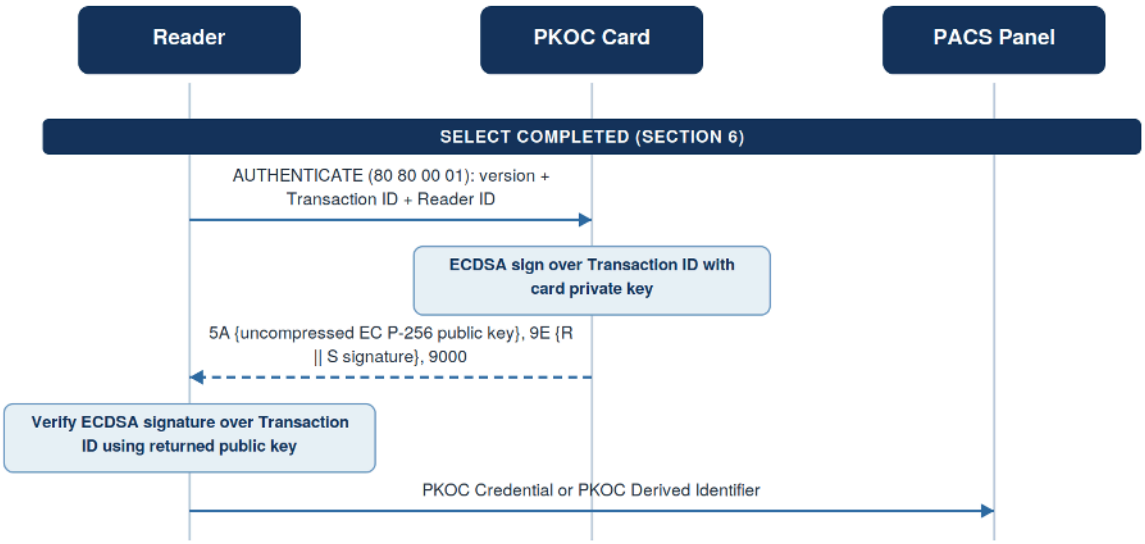

Figure 7-1. SE V1 Standard Mode authentication sequence.

8. PKOC SE V2 Profile Commands

The SE V2 Profile uses a PKOC-CVC in Standard Mode and Validated Mode. Profile Detection uses baseline SELECT and GET DATA (INFO). Standard Mode and Validated Mode use the same profile functions; Validated Mode adds the Validation process. The following table maps the common processes to the SE V2 commands.

Process	SE V2 implementation
Profile Detection	Baseline application detection using SELECT and GET DATA (INFO), Section 6.
Get PKOC-CVC	**GET DATA (PKOC-CVC).**
Validation	The Validation process in Section 3.4, in Validated Mode.
Authentication	**INTERNAL AUTHENTICATE.**
Reader to PACS	Credential derivation according to Core Section 4, or PKOC-CVC extension output.

Profile requirements

- The Card **MUST** return the baseline SELECT response defined in Section 6.1 and the GET DATA (INFO) response defined in Section 6.2.
- The Reader **MUST** require a valid GET DATA (INFO) response containing `02 00` before continuing. Any other result **MUST** fail the SE V2 transaction, and the Reader **MUST NOT** fall back to SE V1. This applies once the SE V2 Profile has been selected; a Reader performing profile detection under Section 6.2 treats the same result as identification of an SE V1 Card.
- The Card **MUST** be provisioned with a supported SE V2 key pair and a PKOC-CVC compliant with Core Section 5. The public key in the PKOC-CVC **MUST** correspond to the private key used by INTERNAL AUTHENTICATE.
- The Card and Reader **MUST** implement GET DATA (PKOC-CVC) as defined in Section 8.1 and INTERNAL AUTHENTICATE as defined in Section 8.2.
- The Reader **MUST NOT** assume that tag `9E` has a fixed length.
- An SE V2 Card **MAY** also implement the SE V1 AUTHENTICATE command for backward compatibility. If the SE V2 key does not conform to the SE V1 baseline, the Card **MUST** use a separate SE V1 key pair.

8.1 GET DATA (PKOC-CVC)

GET DATA (PKOC-CVC) returns the PKOC-CVC containing the public key used by INTERNAL AUTHENTICATE. P1-P2 form tag `7F21`.

Field	Value	Description
CLA	00	Command class.
INS	CA	GET DATA.
P1 / P2	7F / 21	Tag 7F21 — Card Verifiable Certificate.
Le	00	Short-APDU maximum; extended Le **MAY** be used.

The response payload **MUST** be a single BER-TLV object with tag `7F21` containing the complete PKOC-CVC. If the object exceeds the maximum response length, the Card **MUST** use ISO/IEC 7816-4 response chaining and return status word `61XX`. The Reader retrieves remaining data with GET RESPONSE:

Field	Value	Description
CLA	00	Command class.
INS	C0	GET RESPONSE.
P1 / P2	00 / 00	Parameter bytes.
Le	XX	Value from the preceding `61XX` status word.

The Reader **MUST** repeat GET RESPONSE until the Card returns `9000` and **MUST** concatenate all response data into exactly one complete `7F21` object. Fragments **MAY** split the tag, length, or value. Status words include `9000`, `61XX`, `6700`, and `6A88`.

In Standard Mode, the Reader uses the returned PKOC-CVC only as the public-key container. In Validated Mode, the Reader performs the Validation process as defined in Section 3.4.

8.2 INTERNAL AUTHENTICATE

INTERNAL AUTHENTICATE computes a signature over the Reader-supplied 32-byte challenge using the private key corresponding to the PKOC public key in the PKOC-CVC.

Field	Value	Description
CLA	00	Command class.
INS	88	INTERNAL AUTHENTICATE.
P1	00	No specific information.
P2	00	Selects the default SE V2 private key. Alternative P2 values **MUST NOT** be used.
Lc	20	32 bytes.
Data	32-byte challenge	Fresh random nonce from the Reader.
Le	00	Maximum expected response length.

The response is tag `9E` containing the SignatureValue over the challenge. Its format and length **MUST** be determined by the PKOC public-key algorithm in the PKOC-CVC. For ES256 it **MUST** be 64-byte raw `R ‖ S`. For RS256 it **MUST** be an RSASSA-PKCS1-v1_5 signature using SHA-256. Other supported algorithms **MUST** use their applicable algorithm profile. A Reader **MUST NOT** assume a fixed length for tag `9E`. Status words include `9000`, `6985`, `6A86`, and `6A88`.

8.3 Sequence Diagram for SE V2

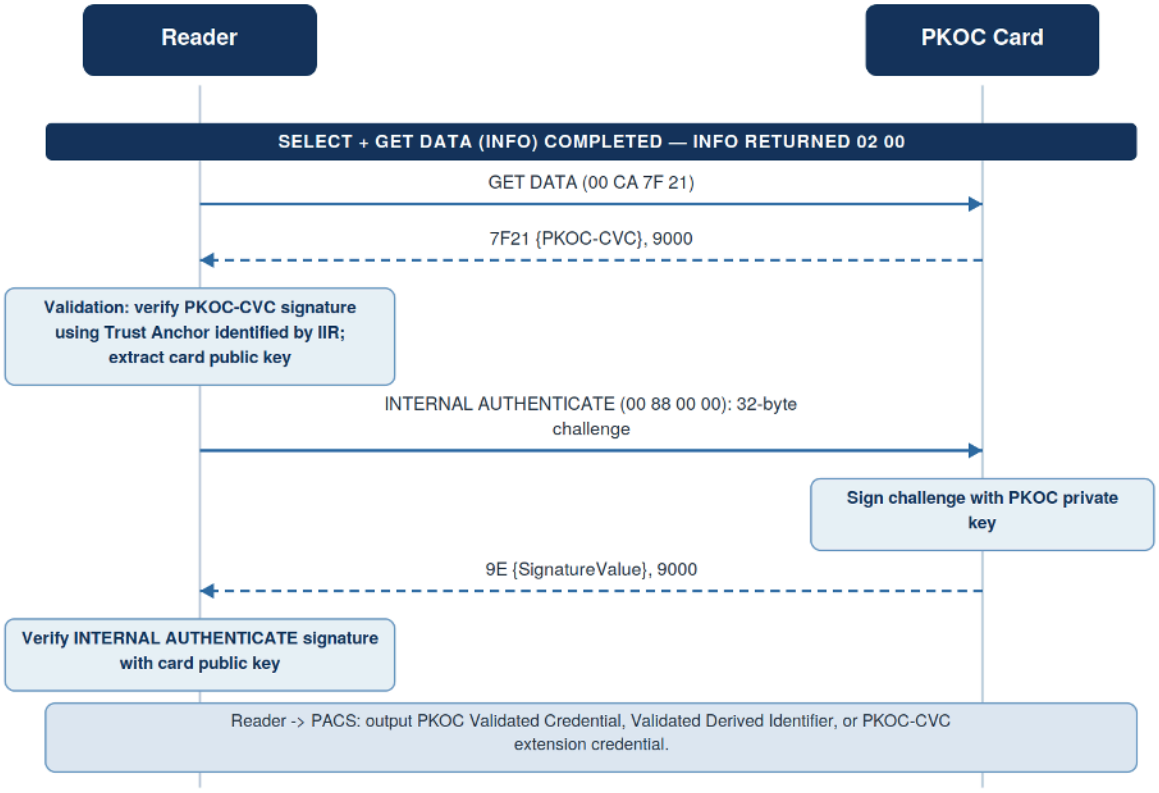

Figure 8-1. SE V2 authentication sequence. In Standard Mode, the validation step is omitted and the same commands are used.

9. PKOC EV-P Profile Commands

The EV-P Profile borrows only the two EV APDU commands — VDEReadData and VDEECDSASign — from VDE-AR-E 2532-100 Table A.1. The other commands in that table are not used by PKOC. It uses the PKOC

application (PKOC AID), a PKOC-CVC in File 0, and NIST P-256; it does not use the native EV certificate, application, or curve.

The EV-P Profile **MUST NOT** use an application AID other than the baseline PKOC AID, **MUST NOT** use a curve other than NIST P-256 for the PKOC key, and **MUST NOT** treat a vendor-issued or chip-manufacturer-issued card certificate as a PKOC trust artifact. In Validated Mode, trust is established solely by a PKOC-CVC verified against a PKOC Issuer Key, which any organization **MAY** operate. Nothing in this profile requires a particular vendor's applet, chip, card operating system, certificate hierarchy, key-distribution service, or licence, and a conforming Reader **MUST** accept any Card meeting the requirements of this section regardless of its supplier.

> **Note:** A card that instead exposes the VDE electromobility AID is not an EV-P card and is outside the scope of this profile; such a card typically uses brainpoolP256r1 and the native EV certificate. Additional EV command information is available in VDE-AR-E 2532-100 and vendor documentation such as NXP AN14223 (informative).

The following table maps the common processes to the EV-P functions.

Process	EV-P implementation
Profile Detection	Baseline application detection and any additional detection defined in Section 9.1.
Get PKOC-CVC	VDEReadData reading File 0 (data file `0x00`), which **MUST** contain a PKOC-CVC.
Validation	The Validation process in Section 3.4, in Validated Mode.
Authentication	VDEECDSASign.
Reader to PACS	Credential derivation according to Core Section 4.

Profile requirements

- The Card **MUST** support selection of the baseline PKOC application AID (Section 6.1) and **MUST** support VDEReadData and VDEECDSASign.
- The Card **MUST** be provisioned with a NIST P-256 (secp256r1) key pair. The private key corresponding to the retrieved PKOC public key **MUST** be used by VDEECDSASign.
- File 0 (data file `0x00`) **MUST** be present and **MUST** contain a PKOC-CVC with an uncompressed NIST P-256 public key; the Card **MAY** support Validated Mode.
- The Card **MUST** make the complete PKOC-CVC retrievable by VDEReadData, using extended length or length negotiation where the PKOC-CVC exceeds the 256-octet short-APDU maximum (Section 9.2).
- The Reader **MUST** perform baseline application detection and any additional Profile Detection defined in Section 9.1.
- The Reader **MUST** use VDEReadData and VDEECDSASign as defined in Sections 9.2 and 9.3 and **MUST** support NIST P-256.
- The VDEECDSASign response **MUST** contain a 64-byte ECDSA signature encoded as raw `R ‖ S`, with R and S each encoded as a 32-byte unsigned big-endian integer. ASN.1 DER signature encoding **MUST NOT** be used.

9.1 EV-P Profile Basis and Application Selection

The Reader **MUST** select the baseline PKOC application AID (Section 6.1). An EV-P card **MUST** return `9000` with no additional response data to SELECT, which distinguishes it from an SE card (which returns the `01 00` version TLV). The EV-P Profile does not use the VDE electromobility AID; a card that exposes that AID is not an EV-P card.

9.2 GET PKOC-CVC (VDEReadData)

VDEReadData reads File 0, which contains the PKOC-CVC used for the transaction.

Field	Value	Description
CLA	80	Proprietary class.
INS	02	VDEReadData.
P1	00	File number (File 0).
P2	00	Reserved.
Le	00	Short-APDU maximum; extended Le **MAY** be used.

VDEReadData is a Case 2 command: no Lc and no command data field are present. An Le of `00` requests the short-APDU maximum of 256 octets; a 3-octet extended Le **MAY** be used where the Card supports it.

File 0 **MUST** be present and **MUST** contain a PKOC-CVC. A PKOC-CVC **MAY** exceed 256 octets — the example in Core Section 5.9 is 265 octets — so the Reader **MUST** be able to retrieve a PKOC-CVC larger than the short-APDU maximum. Where the Card returns `6CXX`, the Reader **MUST** re-issue VDEReadData with Le set to `XX`. Where the Card returns `61XX`, the Reader **MUST** retrieve the remaining data with GET RESPONSE as defined in Section 8.1. The Reader **MUST** assemble exactly one complete `7F21` object before parsing it, and **MUST** fail the transaction if the complete object cannot be retrieved.

When the PKOC-CVC is returned, the Reader parses the `7F21` object and extracts the NIST P-256 PKOC public key. The Card **MAY** support Validated Mode.

In Validated Mode, the Reader performs the Validation process in Section 3.4 using the PKOC-CVC from File 0.

Status words include `9000`, `61XX`, `6700`, `6982`, `6985`, `6A82`, `6A86`, and `6CXX`.

9.3 AUTHENTICATE (VDEECDSASign)

VDEECDSASign requests the Card to sign a Reader-supplied challenge using the private key corresponding to the retrieved PKOC public key.

Field	Value	Description
CLA	80	Proprietary class.
INS	03	VDEECDSASign.
P1	0C	Reserved.
P2	09	Reserved.
Lc	20	Length of data field.
Data	32-byte challenge	Fresh random nonce from the Reader.
Le	00	Maximum expected response length.

The Card computes SHA-256 over the challenge and signs the digest using ECDSA on NIST P-256 (secp256r1). The response **MUST** contain a 64-byte signature encoded as raw `R ‖ S`, with R and S each 32-byte unsigned big-endian integers. ASN.1 DER signature encoding **MUST NOT** be used.

The Reader **MUST** verify the signature using the retrieved PKOC public key before deriving or outputting a credential.

Because the retrieved public key is NIST P-256, the Reader **MUST** derive the PKOC Credential and any PKOC Derived Identifier using PKOC Credential V1 (the X-coordinate) as defined in Core Section 4.3, consistent with the SE V1 baseline. A given key therefore produces the same PKOC Credential on an EV-P Card as it does on an SE V1 Card and on a v1.1 Card. The Card UID **MUST NOT** be used as the PKOC identifier.

Status words include `9000`, `6700`, `6982`, `6985`, `6A86`, `6A87`, and `6C00`.

9.4 Sequence Diagram for EV-P Profile

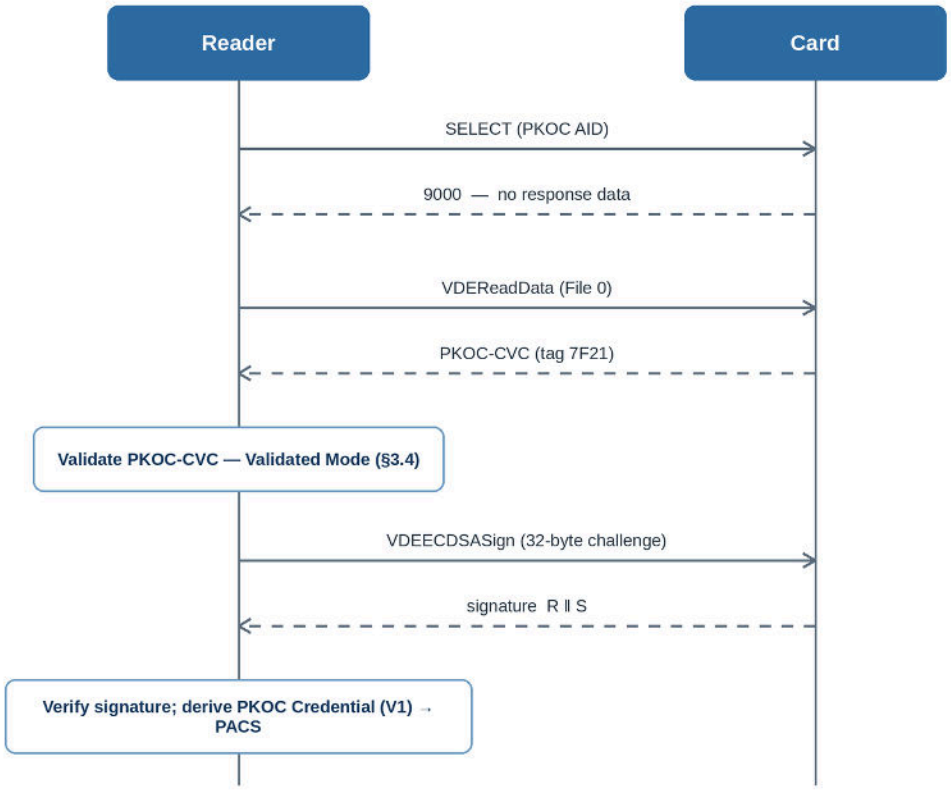

Figure 9-1. PKOC EV-P Profile authentication sequence.

10. PKOC Key Management

Standard Mode establishes trust through prior PACS enrollment and does not require an Issuer Key. Validated Mode uses Issuer Keys as Trust Anchors to verify PKOC-CVC signatures.

Support for Validated Mode, and therefore for Issuer Keys, is **OPTIONAL**. Any organization **MAY** operate its own Issuer Keys and issue its own PKOC-CVCs; PKOC defines no central issuing authority, no registry, and no approval process, and Issuer-key operation is not restricted to any vendor.

A Reader supporting Validated Mode **MUST** be provisioned with the required Issuer Keys before validation is performed and **MUST** provide a mechanism to add, replace, and remove those keys.

10.1 Standard Issuer Keys

Card Issuers that provide PKOC-CVCs using a Standard IIR **MUST** publish the public Issuer Keys in a single file at an issuer-controlled HTTPS endpoint protected by a valid server certificate. The Issuer Key **MUST** have an associated key identifier that exactly matches the Standard IIR using a case-sensitive comparison.

The file **MUST** identify the algorithm used to sign the PKOC-CVC. The file format **SHOULD** be a JSON Web Key Set (JWKS, RFC 7517).

10.2 Private Issuer Keys

An organization **MAY** issue PKOC-CVCs using a private trust service and **MAY** distribute Issuer Keys through a public or private mechanism. The organization **SHOULD** use the JWKS format. Private Issuer Keys **MUST NOT** be included in a public Standard-IIR JWKS resource.

11. References

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

- **PSIA NFC Card 1.1** — PSIA NFC Card Specification v1.1 (Dec 2023): the SE V1 SELECT + AUTHENTICATE protocol.
- **VDE-AR-E 2532-100:2021-07** — Requirements for authentication for the use of electric mobility supply facilities.
- **NXP AN14223** — MIFARE DUOX for EV charging use cases, Rev. 1.0, 10 July 2025 (informative).
- **ISO/IEC 14443-4:2018** — Contactless proximity objects, transmission protocol.
- **ISO/IEC 7816-4:2020** — Organization, security, and commands for interchange.
- **ISO/IEC 7816-6:2023** — Interindustry data elements for interchange.
- **FIPS 204** — Module-Lattice-Based Digital Signature Standard (ML-DSA).
- **RFC 5639** — ECC Brainpool Standard Curves.
- **RFC 7517 / RFC 7518** — JSON Web Key, JSON Web Key Set, and JSON Web Algorithms.
- **RFC 8017** — PKCS #1 v2.2.
- **RFC 8615** — Well-Known Uniform Resource Identifiers.
- **RFC 9964** — ML-DSA for JOSE and COSE.
- **NIST CSOR** — Computer Security Objects Register.
- **PKOC Core Specification**, Version 2.0.1 — PSIA.

End of PKOC NFC Transport Profile, Version 2.0.1.