EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL

Ensuring High Assurance of EUDI Wallets
& the NL National eID Scheme



NL-WI-conditions

NL EUDI Wallet Certification System

Conditions on the Underlying Mobile System for Wallet Instances in a remote HSM architecture

RC1.1

2026.10.20

Document reference	NL-WI
Version	RC1.1
Status	Release Candidate
Date	20/10/2026
Document type	Certification Scheme Conditions on the Underlying Mobile System for Wallet Instances in a remote HSM architecture
Parent framework	NL-CS (parts1/2/3/4)
Scheme Owner	Kingdom of the Netherlands Minister of Economic Affairs
Certification target	European Digital Identity Wallets
Primary audience	Scheme Owners, Evaluation Laboratories; CABs, EUDI Wallet providers
Secondary audience	EUDI Wallet ecosystem stakeholders
Assurance level <identity proofing CIR 2015/1502> and <evaluation, EU Regulation 2019/881>	High/High
Related Documents	NL-WI NL-WSCA (-conditions)

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

Version	Change	State	Date
RC1.0	Result of European Collaboration / Discussions	Release Candidate	2026-10-27
RC1.1	Release Candidate for publication following review round NL Certification System Mngt Board	Release Candidate	2026-10-31

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

Table of Contents

1 Introduction 5
2 Security controls 6
2.1 SC-01 – Vulnerability management 6
2.2 SC-02 – Protections against exploits 6
2.3 SC-03 – Trusted firmware 6
2.4 SC-04 – Secure random number generator 7
2.5 SC-05 – Secure cryptographic primitives 7
2.6 SC-06 – Isolated environment for keys 8
2.7 SC-07 – App sandbox 10
2.8 SC-08 – Integrity attestations 10
2.9 SC-09 – Secure lock screen 10
2.10 SC-10 – Stored data confidentially 11
2.11 SC-11 – Secure networking 11
2.12 SC-12 – User Device privacy 11
3 Abbreviations 13
4 References 14

1 Introduction

An EUDI Wallet Instance operates on a mobile device that forms the foundational security layer of the Wallet Unit. Although critical cryptographic operations and long-term key material may be protected by remotely operated Wallet Secure Cryptographic Applications and Devices, the Wallet Instance still relies on its mobile device to protect locally processed authentication factors, personal data, attestations, transaction information and security-relevant application data. The security of the Wallet Instance can therefore not be established solely through evaluation of the wallet application. It also depends on the security capabilities, configuration and lifecycle support of every mobile device on which the Wallet Instance is permitted to run.

This document list the requirements following a risk analysis that a mobile device, referred to as the User Device, must fulfil to provide an adequately secure operating environment for an EUDI Wallet Instance. The requirements address vulnerability management, protection against exploitation, trusted firmware, secure generation of random values, secure cryptographic implementations, hardware-backed isolation of cryptographic keys, application sandboxing, integrity attestation, device access control, protection of stored data, secure networking and preservation of user privacy.

Collectively, these controls are intended to protect the Wallet Instance against attackers with high attack potential, including attackers capable of remote exploitation or physical access to the device.

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

2 Security controls

2.1 SC-01 – Vulnerability management

2.1.1 Requirements

The UD shall receive regular, timely security updates addressing disclosed vulnerabilities in its first- and third-party components.

2.1.2 Evaluation criteria

The following criteria shall be used to ensure the fulfilment of this security control:

1. The OS version / security patch level of the UD shall be checked at runtime to ensure that the UD runs an OS with recent security patches. On iOS this can be done, for example, via the `UIDevice.current.systemVersion` Swift API and on Android via the `android.os.Build.*` APIs. Note that there may be additional versions, e.g., of the kernel, which may also be relevant.
2. The wallet's design shall be capable of revoking WIs in case the associated UD can no longer ensure its security, e.g., in case there are known exploits (cf. SC-02 – Protections against exploits) or UD is significantly out of date. This can be done, for example, when the WI is being associated with device information (device vendor, operating system version, etc.) in the Wallet Provider Backend. Device identification information can be obtained at runtime and may even be readable in an unspoofable way via integrity attestations (e.g., via ID attestations on Android; see also. SC-08 – Integrity attestations).

2.2 SC-02 – Protections against exploits

2.2.1 Requirements

The UD shall follow best practices to protect against hardware and software vulnerabilities in any of its components, like memory corruption attacks and supply chain attacks.

2.2.2 Evaluation criteria

The following criteria shall be used to ensure the fulfilment of this security control:

1. Due diligence shall be done to ensure that the UD implements protections against exploits, e.g., via review of platform documentation related to implemented mechanisms protection from local and remote attacks (e.g., regarding the presence of new or unpatchable exploits).

2.3 SC-03 – Trusted firmware

2.3.1 Requirements

The UD shall ensure that any code executed with elevated privileges (e.g., code of the bootloader, kernel, OS, and TEE) is trusted and cannot be rolled back to a less secure version.

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

2.3.2 Evaluation criteria

The following criteria shall be used to ensure the fulfilment of this security control:

1. On UDs where secure boot is not always enforced, its activation status shall be checked at runtime. On Android, for example, the verified boot status (key, locked, state, hash) can be checked using the APIs provided for verifying the integrity of apps (see SC-08 – Integrity attestations for details).
2. Due diligence shall be done, e.g., via review of platform documentation related to implemented mechanisms for protection against local and remote exploits (e.g., in regard to the presence of new or unpatchable exploits). Thereby, it shall be noted that the presence of unpatchable exploits, for example, in the boot-up sequence, may make this security control unenforceable.

2.4 SC-04 – Secure random number generator

2.4.1 Requirements

The UD shall implement a Random Number Generator (RNG) that provides unpredictable randomness to generate cryptographic keys and nonces (e.g., IVs or challenges), with each output of the RNG being only accessible to a single app.

2.4.2 Evaluation criteria

The following criteria shall be used to ensure the fulfilment of this security control:

1. For RNGs used under control of the WI, where available, APIs of the UD shall be used to determine that the provided security strength/level is sufficient for its use case (e.g., at least 128 bits). This can be done, for example, on Android by setting the minimum strength during RNG instantiation via the `SecureRandom` API.
2. Where no information about the RNGs' security strength is available at runtime but the RNG outputs are accessible to the WI, health tests of the random data shall be performed at runtime to detect severe entropy deficits (e.g., via statistical testing).
3. Where runtime checks are not available, where possible, cryptographic primitives and protocols shall be used that do not rely on strong randomness (e.g., nonce-reuse resistant block cipher modes or deterministic signature schemes).
4. Where runtime checks and the use of deterministic crypto schemes is not possible, due diligence shall be done to determine the security strength of the RNGs used on the UD, e.g., via review of platform documentation and available security certifications. Note that this may be the only option in case an RNG is only accessed indirectly by the WI, e.g., for keys generated within a SE/TEE (cf. SC-06 – Isolated environment for keys) or as part of the UD's network protocols (cf. SC-11 – Secure networking).

2.5 SC-05 – Secure cryptographic primitives

2.5.1 Requirements

The UD shall provide standard-compliant and vulnerability-free implementations of all cryptographic primitives not implemented by the wallet app itself.

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

2.5.2 Evaluation criteria

The following criteria shall be used to ensure the fulfilment of this security control:

1. Known-answer testing shall be performed for all cryptographic primitives provided by the UD and used by the WI to ensure their standard compliance (e.g., by using a precomputed plaintext-ciphertext pair with a fixed key).

2.6 SC-06 – Isolated environment for keys

2.6.1 Requirements

The UD shall provide an isolated environment for cryptographic keys that

1. Provides means for key generation, key storage, key deletion/zeroization, and use of these keys for cryptographic operations,
2. Does not allow keys inside the environment to be exported / accessed from outside,
3. Provides key attestations as evidence that keys are actually stored in such an environment, and
4. Is purposely designed to mitigate physical and logical tamper and side-channel attacks against the above properties.

The environment shall be provided by a dedicated SE or a hardware-backed TEE. As TEEs share their hardware with other processes on the UD, thus greatly increasing their attack surface, the following additional requirements shall apply for such implementations:

5. Only trusted code shall run inside the TEE supported by a secure boot mechanism (cf. SC-03 – Trusted firmware),
6. The memory of TEE shall be strongly isolated from accesses originating outside the isolated environment, and
7. The TEE shall still keep all its security properties in case the normal, rich OS is compromised.

2.6.2 Evaluation criteria

The following criteria shall be used to ensure the fulfilment of this security control:

1. Runtime checks shall ensure that critical key material is stored in a SE or TEE of the UD. This can be done, for example, on iOS using the `CryptoKit` and on Android using the `KeyInfo` API.
2. To further ensure the correct storage of critical keys, key attestations shall be obtained at runtime and validated in a trusted environment (i.e., never in the UD itself) based on the latest platform guidance. See SC-08 – Integrity attestations for details on how this can be done. It is of particular importance that SC-01 – Vulnerability management is enforced for the hardware identified in the attestation.
3. If considered necessary for additional assurance, due diligence via an off-device analysis of whether a SE/TEE included in a particular UD meets the security level required by the considered attacker model shall be done (see below).

As the security level of an isolated environment for keys provided by a UD cannot be determined at runtime or hardly at all for closed-source systems without assistance by the vendor, the following list provides guidance on how the necessary assurance may be obtained:

1. Presence of a valid, i.e., not expired or withdrawn, certificate proving the required security properties that traceable to the UD in question (e.g., regarding model and

version). Reputable, independent security evaluations or certification schemes include EUCC, FIPS 140-3, SESIP, and GlobalPlatform. The certificate should provide assurance in the fulfilment of the concrete requirements of this security control and not just generic security properties. For example, the required non-exportability of keys is usually not covered by certifications of hardware platform but is rather a property of composite certifications that also consider the concrete applications running on the certified hardware.

Examples of certification schemes for TEEs and SEs that cover (parts of) this security control include:

a. For TEEs, certifications based on the GlobalPlatform TEE Protection Profile [1] or via tested compliance of the TEE implementation with a standardized functional specification.
b. For SEs, a composite certificate that covers both the hardware platform and a concrete application running on it is ideal. For the highest assurance, the hardware itself should have a valid EUCC certificate against one of the following Protection Profiles (PP): PP0084 or PP0117. With EUCC, this also ensures the presence of a vulnerability management process. For Android 17, the CDD requires any StrongBox KeyMint, which corresponds to a SE, to be evaluated/certified according to these PPs or a similar scheme [2, 9.11.2].

Note that security certificates do not necessarily have to be made public and can vary in scope and the considered attacker model. It can also be difficult to determine whether a concrete instance of a SE/TEE underwent a security certification, as identification and versioning information may not always be available. Also note, that in general independent security certifications provide assurance in the fulfilment of security requirements, but do not necessarily result in a more secure product overall.

2. When no certificate from an independent evaluation/certification body is available, assurance in the fulfilment of the required properties can be obtained, for example, via independent functional testing and penetration testing (e.g., similar to what is usually done for full EUCC certifications) as well as from other information publicly available (e.g., technical documentation and research papers).
3. For TEEs, the following properties, among others, may be used as additional indicators for determining the security level provided by a TEE in a UD:

a. Isolation of processing and storage of the TEE by hardware-based mechanisms.
b. Isolation mechanisms between TAs running within the TEE.
c. Secure integration of the TEE into the secure boot process and root of trust.
d. Secure integration of the TEE into Android.
e. Implementation of security mechanisms against software and physical attacks.
f. Small, maintainable, and auditable code base.
g. Up-to-date and correctly implemented suite of cryptographic schemes.
h. Availability of design information and source code for review.
i. Still actively supported by the vendor with security updates (cf. SC-01 – Vulnerability management).
j. History of vulnerabilities and corresponding security patches.

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

2.7 SC-07 – App sandbox

2.7.1 Requirements

The UD shall be able to or provide means to isolate app processes, memory, cryptographic keys, and data from unauthorized accesses by other apps or system services. This shall include direct programmatic accesses but also indirect means, like side channels, drawing menus on top of other running apps, or by abusing accessibility services, as well as unwanted use by the UD itself (e.g., for backups or personalization).

2.7.2 Evaluation criteria

The following criteria shall be used to ensure the fulfilment of this security control:

1. On UDs where multiple apps can draw onto the screen at the same time, runtime measures shall be implemented to ensure that no other process draws on top of the user interface of the wallet app. This can be done, for example, on Android using the `HIDE_OVERLAY_WINDOWS` and/or `filterTouchesWhenObscured` APIs.
2. On UDs where screen content is accessible to other apps, e.g., malicious accessibility services, runtime measures shall be implemented to ensure that these other apps cannot access sensitive wallet data. This can be done, for example, on Android using the `isAccessibilityDataSensitive` API.
3. Due diligence shall be done to ensure that the UD implements a non-bypassable app sandbox, e.g., via review of platform documentation or independent testing.

2.8 SC-08 – Integrity attestations

2.8.1 Requirements

The UD shall implement a hardware-supported attestation mechanism to prove the integrity of the wallet app at runtime (i.e., that it has not been tampered with). Where OS modifications are likely, the UD shall also provide support in obtaining attestations about the integrity of the OS.

2.8.2 Evaluation criteria

The following criteria shall be used to ensure the fulfilment of this security control:

1. The WI shall obtain integrity attestations for the app and possibly OS at runtime, with their validation being performed in a trusted environment (i.e., never in the UD itself) based on the guidance provided for the respective API. This can be done, for example, on iOS using Apple's App Attest API and on Android via the verification of hardware-backed key pairs or alternatively the Google Play Integrity API (basic or strong integrity).

2.9 SC-09 – Secure lock screen

2.9.1 Requirements

The UD shall implement a hardware-backed lock screen requiring authentication before granting full access to the device that is resistant to bypass.

2.9.2 Evaluation criteria

The following criteria shall be used to ensure the fulfilment of this security control:

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

1. Runtime checks shall be used to enforce that a secure lock screen is activated on the UD. This can be done, for example, on iOS via the `LocalAuthentication` API and on Android via the `KeyguardManager` API.

2.10 SC-10 – Stored data confidentially

2.10.1 Requirements

The UD shall protect app data stored at rest (i.e., in non-volatile memory) from extraction in plaintext by use of a secure memory encryption algorithm.

2.10.2 Evaluation criteria

The following criteria shall be used to ensure the fulfilment of this security control:

1. On UDs with an OS that does not always enforce the use of memory encryption, runtime checks shall be used to determine its activation status on every storage media where WI data is to be stored persistently. On Android, for example, the `DevicePolicyManager` API can be used to determine whether a particular storage media is encrypted.
2. Where information about the security strength of the memory encryption algorithm on a UD is not available at runtime (e.g., use of at least XTS-AES with at least 128-bit keys), due diligence shall be done instead to determine its general security level, e.g., via review of platform documentation.

2.11 SC-11 – Secure networking

2.11.1 Requirements

The UD shall provide the following means for secure networking:

1. Mechanisms to prevent all unencrypted/plaintext networking, and
2. Secure implementations of current network protocols (e.g., TLS 1.3) used for transactions, when not implemented by the wallet app itself.

2.11.2 Evaluation criteria

The following criteria shall be used to ensure the fulfilment of this security control:

1. Runtime checks shall be performed to ensure that the UD supports and enforces the use of the required, secure networking protocols. This can be done, for example, on iOS using the `Network.framework` API and on Android using the `NetworkSecurityPolicy` API.
2. The opposite connection end of the WI shall perform server-side checks to ensure that the correct protocols are used in a standard-compliant way (e.g., by checking that handshake messages are received in the correct order).

2.12 SC-12 – User Device privacy

2.12.1 Requirements

The UD shall preserve the user's privacy by not monitoring apps and not providing means to track app users in a way that would disclose sensitive information about the wallet to the UD's vendor or third parties (e.g., using analytics or device-unique identifiers).

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

2.12.2 Evaluation criteria

The following criteria shall be used to ensure the fulfilment of this security control:

1. Due diligence shall be done to ensure that the UD provides a privacy level sufficient for running the WI, for example, via review of platform documentation detailing the scope of data disclosed to the UD's vendor or third parties.

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

3 Abbreviations

Abbreviation	Term
API	Application Programming Interface
CDD	Compatibility Definition Document
EUCC	European Union Common Criteria
FIPS	Federal Information Processing Standards
IOT	Internet of Things
OS	Operating System
RNG	Random Number Generator
SC	Security Control
SE	Secure Element
SESIP	Security Evaluation Standard for IOT Platforms
TEE	Trusted Execution Environment
TLS	Transport Layer Security
UD	User Device
WI	Wallet Instance

Table 1: Abbreviations used in this document

4 References

[1] *Android 17 Compatibility Definition*, Google, https://source.android.com/docs/compatibility/17/android-17-cdd.

[2] *TEE Protection Profile v1.3* | *GPD_SPE_021*, GlobalPlatform, v1.3, 2020-09, https://globalplatform.org/specs-library/tee-protection-profile-v1-3/.

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme