THREAT MODEL

version 10.0

INTERNET IDENTITY CARD ™

Security analysis, trust assumptions & residual risks

Issuer	Https Card — Internet Identity Card Ltd
Company Registration	UK Companies House No. 09168431
Trademark	UK IPO UK00003166480 (2016) · INPI FR 4071419
US Copyright	TX 8-508-373 (2014/2017)
Office	124 City Road, London EC1V 2NX, United Kingdom
Disclosures	TDCommons #10079 · #10167 · #10394 · #10795 · #10796 · #11121 (CC BY 4.0)

This document supersedes the v9.0 edition. IIC v10.0 is a corrective release: it introduces no new capability and repairs six defects found in the v9.0 implementation. Two of those defects fell outside the scope of the v9.0 threat model, which described the exported card at rest but did not name the issuer's local vault as an asset. This edition adds that asset, the adversary that reaches it, and the residual risks that remain after the correction.

A threat model is not a security guarantee. It states what the system is designed to do under stated assumptions; the conditions under which those protections do not apply are documented in Section 9.

1. Scope & objectives

This document describes the threat model of the Internet Identity Card (IIC) v10.0: the assets it protects, the adversaries it defends against, and the residual risks it does not. It supersedes the v9.0 threat model.

The v9.0 edition documented the exported card. Its Section 4 listed identity data at rest as protected by AES-256-GCM under an Argon2id-derived key, which was and remains accurate for the exported card. It did not describe the issuer's local vault — the browser-side state holding the issuer's identity and private signing key — either as an asset or as a surface. In v9.0 that vault was encrypted under a key written in unencrypted beside the ciphertext it protected, so any party able to read the browser profile could decrypt it. The memorable word controlled what the interface displayed; it derived no cryptographic material.

The precise cost of that recovery is worth stating, because a reader may otherwise assume either more or less than was the case. The identity data itself was never in the clear: it was encrypted with AES-256-GCM. What was in the clear was the key, held as thirty-two raw bytes in base64 in the same browser storage, under an adjacent name. Recovering the identity and the private signing key therefore required no passphrase, no secret and no cryptographic attack — a base64 decode and a standard decryption with the key already in hand. Opening the profile directory did not display the data; a few lines of code did.

Naming that omission is the substance of this edition. A model that silently gains a section is less informative than one that states what it previously failed to cover.

2. System overview

The IIC is a single self-contained HTML file. Identity data in an exported card is encrypted at rest with AES-256-GCM under an Argon2id-derived key. The file verifies its own integrity offline via an embedded SHA-256 of its own serialisation. Where authorship must be provable, the issuer signs the card identity and any signed documents with the hybrid suite. Blockchain anchoring (Bitcoin and Ethereum) provides authority-free proof of existence at a date. No server, PKI, or certificate authority is involved at any point; verification runs in a vanilla browser, offline.

New in v10.0: the issuer's local vault is described as a distinct component. It holds the issuer identity, the private signing key and the signature chain, and is encrypted under a state key that is never stored. That key is wrapped in two independent envelopes — one derived from the memorable word, one from a recovery phrase generated at setup — and exists in memory only for the duration of a session.

3. Trust assumptions

The security properties in Sections 7 and 8 hold only while these assumptions are true:

- The device and browser executing the file are not compromised.
- The passphrase, the memorable word and the recovery phrase are chosen well, kept secret, and — in the case of the recovery phrase — kept where the holder can still reach it.
- Public keys used to verify a card are obtained from a trusted channel; the file proves what was signed, not who the signer is.
- For post-deprecation verification, the verifier can consult a blockchain header chain to confirm an anchor timestamp.
- New in v10.0: the browser's storage isolation holds while the vault is locked. Once unlocked, the state key exists in page memory and is exposed to anything with script access to that page.

4. Assets under protection

Asset	Protection
Identity data at rest in an exported card	AES-256-GCM, Argon2id-derived key (96 MiB, t=4, p=4)
Issuer vault at rest (new in v10.0)	AES-256-GCM under a state key that is never stored. The key is wrapped in two envelopes, derived with Argon2id (96 MiB, t=4, p=4) from the memorable word and from the recovery phrase. Authentication is the GCM tag; no password hash is retained.
Issuer private signing key	Held inside the vault, and therefore inaccessible without one of the two secrets
File integrity	Embedded SHA-256 self-check, fail-closed lockdown, offline

Authorship	Hybrid ECDSA P-256 + ML-DSA-65 signature over the signed core
Local signature chain	Hash-linked entries with a monotonic counter and a truncation anchor, so a retained chain still verifies to a known point
Proof of existence at a date	Bitcoin + Ethereum timestamp anchors

5. Adversaries & capabilities

- **A1 — Network adversary.** Sees or alters data in transit. Defeated by offline operation and AES-256-GCM.
- **A2 — Storage adversary.** Obtains an exported card at rest. Faces AES-256-GCM + Argon2id.
- **A3 — Forger.** Attempts to alter a card or forge a signature. Faces SHA-256 self-integrity and the hybrid signature, both components required.
- **A4 — Quantum adversary (future).** A large fault-tolerant quantum computer able to run Shor's algorithm against ECDSA P-256. Addressed by the hybrid construction.
- **A5 — Local profile adversary (new in v10.0).** Reads the browser profile of the issuer's device: malware running as the user, a shared or unattended machine, a disk image, a synchronised profile backup. This adversary defeated v9.0 outright, because the vault key sat beside the ciphertext. In v10.0 it faces Argon2id-wrapped envelopes and obtains nothing usable while the vault is locked. It is not defeated while the vault is unlocked; see Section 9.

6. STRIDE analysis

Mapping the six STRIDE categories onto the IIC v10.0 architecture. Dispositions marked **corrected in v10.0** were stated more favourably in the v9.0 edition than the implementation supported.

Category	Applied to IIC v10.0	Disposition
Spoofing	Hybrid ECDSA P-256 + ML-DSA-65 signing binds a card to an issuer key. A valid signature requires both private keys.	MITIGATED
Tampering	SHA-256 page self-integrity with fail-closed lockdown detects any byte change outside the hash zone, verifiable fully offline.	PROTECTED
Repudiation	No central log; blockchain timestamping provides external proof of existence at a date. The local chain now carries a monotonic counter and a truncation anchor, so a truncated history is no longer silently unverifiable.	MITIGATED *corrected in v10.0*
Information disclosure	Exported cards: AES-256-GCM with Argon2id-derived keys. Issuer vault: same construction, under a key that is no longer stored anywhere. In v9.0 the vault key was written in unencrypted beside the ciphertext, so this category was not in fact protected for the issuer's own data.	PROTECTED *corrected in v10.0*
Denial of service	A single offline file has no service to deny. Loss of access is now governed by two secrets rather than one; loss of both remains final.	PROTECTED
Elevation of privilege	Strict CSP and Permissions-Policy disable camera, microphone, geolocation, payment and USB in the exported card.	PROTECTED

7. Concrete attack scenarios

- **Byte-level tamper.** Any change outside the hash zone breaks the SHA-256 self-check and triggers fail-closed lockdown. Detected offline.
- **Signature forgery (classical).** Forging ECDSA alone fails: the ML-DSA-65 component must also verify.
- **Profile copy (new).** An adversary copies the issuer's browser profile. Under v9.0 this yielded the vault key in unencrypted and, with it, the private signing key: the adversary could sign as the issuer. Under v10.0 the copy yields two Argon2id-wrapped envelopes and no key. The adversary is reduced to guessing the memorable word or the recovery phrase, at 96 MiB per attempt.
- **Silent post-quantum degradation (new).** Under v9.0, any failure in ML-DSA-65 signing was swallowed and a classical receipt was emitted that was indistinguishable from a legitimate one: the holder believed a hybrid signature had been made and it had not. Under v10.0 a configured hybrid signature that cannot be produced causes signing to

be refused and the cause reported. No misleading artefact is created.

- **Harvest-now, decrypt-later.** An adversary storing an encrypted card today cannot decrypt it with a future quantum computer: AES-256 retains ~128-bit security under Grover's algorithm.
- **Harvest-now, forge-later.** An adversary storing a signed card today cannot forge authorship: ML-DSA-65 remains secure, and the blockchain anchor proves the card existed before ECDSA deprecation.
- **Post-deprecation forgery.** A forged card carrying only a quantum-broken ECDSA signature fails ML-DSA-65 verification and lacks a valid pre-deprecation anchor.

8. Recovery and loss model

The v9.0 edition stated that loss of the passphrase was unrecoverable by design. That is no longer accurate for the issuer vault and the change should be read carefully, because it alters what the holder must protect.

The vault is opened by either of two secrets, each wrapping the same state key. The memorable word is the everyday secret. The recovery phrase — twenty-four characters drawn from a 62-character alphabet by rejection sampling, approximately 142 bits — is generated at setup, displayed once, offered as a file, confirmed by transcription, and stored nowhere in any form.

Two consequences follow. Forgetting the memorable word is now recoverable, provided the phrase was kept. Losing both remains final, with no party able to intervene: there is no server-side copy, no escrow and no support path, by construction. A vault created without a memorable word has the recovery envelope only, so the phrase is then the sole way in; the interface states this at the point where the phrase is shown.

9. Residual risks & out of scope

A credible threat model names what it does not defend against. The following remain out of scope for IIC v10.0:

- **Compromised endpoint.** A malicious browser or OS can subvert any in-page cryptography. This is unchanged and is the dominant residual risk.
- **Unlocked vault.** Once the vault is open, the state key is in page memory and the private signing key is reachable by anything with script access to that page. The correction in v10.0 protects the vault at rest; it does not protect an open session. JavaScript cannot guarantee erasure of key material, since the engine may have relocated it during heap compaction.
- **Storage residue after migration.** Removing an entry from browser storage does not guarantee that the bytes leave the disk; the underlying store may retain the previous unencrypted key until compaction. Migration protects a vault against future profile access. It cannot retroactively protect one whose profile has already been copied. A holder who believes the profile was compromised should create a new identity rather than migrate the existing one.
- **Key custody.** The system proves what was signed, not that the signer's key belongs to a particular real-world identity; that binding is external.
- **Offline revocation.** A frozen offline file cannot receive a revocation signal without an information channel; short-lived credentials or pre-downloaded status lists remain the realistic approaches.
- **Receipt provenance.** A receipt states what was signed and by which key. The card identifier, fingerprint and verification key it carries are all supplied by the signer, so a receipt alone does not establish that the signer is the party they claim to be.
- **Verification of pqStatus by exported cards.** v10.0 receipts carry a status field distinguishing a classical receipt from a hybrid one. The verification engine embedded in cards exported before this release does not read it, and would report a malformed receipt as classical. This is a known open item.

Quantum adversaries. AES-256 retains ~128-bit strength under Grover. ECDSA P-256 is not the sole authorship mechanism: ML-DSA-65 (FIPS 204) is present in a hybrid construction, following ANSSI's stated preference for hybrid constructions during the transition period. A quantum computer able to break ECDSA does not break the card's authorship. Hybrid PQC-readiness in code is distinct from a formal ANSSI qualification, which requires evaluation by an approved CESTI laboratory.

10. Operational recommendations

- Set a memorable word. A vault without one can be opened only by the recovery phrase. Note that in v10.0 the word can be set once and not subsequently changed, and a recovery phrase cannot be reissued; both operations are implemented but unreachable from the interface. Loss or exposure of either secret is therefore a reason to create a new identity.
- Store the recovery phrase offline, away from the device holding the vault. A phrase kept in the same profile as the vault it protects defends against nothing.
- Enable hybrid signing when authorship needs to be provable, and treat a refusal to sign as a fault to investigate rather than an obstacle to work around.
- Anchor the card or its package to Bitcoin and Ethereum to obtain proof of existence at a date.
- Retain a blockchain header chain, or access to one, for long-term post-deprecation verification.
- Treat a suspected profile compromise as a reason to create a new identity, not to migrate the existing one.

11. Verification status of this edition

Stated so that the reader can weigh what follows. The v10.0 build has been exercised on a single machine, in a single browser, against test vaults. Six defects were identified by code review, of which three were confirmed independently by two separate automated reviewers; four further defects were found only by executing the build, in paths that review had passed over. The release has not been examined by a third party, has not been anchored, and carries no formal evaluation.

12. Disclaimer

This threat model is provided for informational purposes only. It describes intended security properties under the trust assumptions of Section 3. Where those assumptions do not hold, the protections described may not apply. “PQC-ready” denotes conformance to the ML-DSA (FIPS 204) standard in code; it does not constitute a FIPS 140 CMVP validation or an ANSSI qualification, both of which require formal third-party evaluation.