The Prson Trust Architecture: Verifiable Personhood, Unlinkable Disclosure, and Capture-Time Media Provenance in a Non-Custodial Identity Wallet

Technical white paper

Bruno Silva

Yumankind OÜ — Prson Network

Tallinn, Estonia · hello@yumankind.com · https://app.prson.net

Version 2.1 · June 2026

Abstract. Generative models have collapsed the marginal cost of fabricating people, documents, and audiovisual evidence [12, 13, 14], while conventional identity verification responds by demanding ever more disclosure — concentrating exactly the data whose leakage it should prevent. We present the architecture of Prson, a deployed, non-custodial identity wallet and proof network that decomposes "trust" into independently verifiable, minimally disclosing primitives: (i) a hierarchical key model in which no private key material exists server-side; (ii) a graded verification ladder culminating in cryptographic validation of government eMRTD chips per ICAO Doc 9303 [25]; (iii) a Sybil-resistance mechanism [9] that enforces *one document, one identity* through a salted-hash alias registry while releasing only audience-scoped (pairwise) identifiers, so that uniqueness is provable per relying party yet computationally unlinkable across relying parties; (iv) selective disclosure via IETF SD-JWT-VC [31] with per-audience holder keys derived by HKDF [21] and signed boolean predicates over undisclosed values; (v) capture-time media provenance combining platform attestation, on-device C2PA-compatible signing [15], and notarial countersignature, yielding photographs and videos that remain verifiable offline and after the issuer's disappearance; and (vi) a PAdES-conformant [37] document-signing PKI whose leaf certificates bind a signer's notarised legal name to keys that never leave the device. We position the design against three decades of anonymous- credential research [2, 3, 5, 6, 7], the W3C/IETF verifiable- credential stack [29, 30, 31], proof-of-personhood systems [10, 11], and the European Digital Identity framework [39]. We state the threat model, give the constructions with security arguments, and analyse the residual trust assumptions honestly — including what a full compromise of our own servers could and could not achieve.

Keywords: self-sovereign identity · proof of personhood · Sybil resistance · pairwise pseudonymity · anonymous credentials · selective disclosure · SD-JWT-VC · OpenID4VP · media provenance · C2PA · PAdES · eIDAS 2.0 · NIST SP 800-63

Contents

1 Introduction

1.1 Two coupled failures

Authenticity collapse. The forensic position — detect fabrication by analysing artefacts after the fact — is losing an arms race it cannot win. Surveys of media forensics conclude that detector generalisation degrades sharply outside the manipulation families seen in training [13], and benchmark studies show detection accuracy falling toward chance as generators improve and as content is re-compressed in distribution [14]. Chesney and Citron [12] identify the deeper harm as the *liar's dividend*: once fabrication is cheap, genuine evidence becomes deniable. Any system whose trustworthiness rests on "looking real" must therefore be presumed broken; authenticity has to be established at *capture time* and carried as verifiable provenance, a position now institutionalised by the C2PA effort [15] — which, however, deliberately leaves open *who* captured and whether the capturing environment was genuine. Sections 10–11 address exactly that gap.

Disclosure inflation. The dominant industrial response to online fraud — know-your-customer pipelines — answers every doubt with more collection. A user proves a single bit (*over 18*) by exposing a full identity document, to every service that asks, each verification event creating another replica of exactly the data whose theft enables the impersonation the check was meant to prevent. The re-identification literature shows that such accumulations compound: sparse attribute sets that look innocuous individually identify people uniquely in combination [17, 18]. The cryptographic community solved minimal disclosure in principle decades ago — Chaum's pseudonym systems [2], Brands' selective-disclosure certificates [3], and the Camenisch–Lysyanskaya line of anonymous credentials [5, 6] underlying Idemix, U-Prove [7] and modern BBS-based schemes [4, 36] — yet deployment remained marginal, largely for ecosystem rather than cryptographic reasons: no issuers, no verifier tooling, no recovery story, no business model [28].

1.2 The personhood dilemma

Between these failures sits the problem this work centres on: *proof of personhood* — convincing a service that an account is controlled by a real, unique human [10, 11]. Douceur's foundational result [9] shows that without a trusted external anchor, one entity can present arbitrarily many identities; every practical system therefore picks an anchor, and the choice determines the privacy outcome. Biometric anchors (e.g. iris-code registries [19]) achieve strong uniqueness at the cost of a global biometric identifier and a custodial database of exactly the most sensitive data class. Social-graph and ceremony anchors [10, 11] avoid biometrics but scale poorly and resist Sybils only statistically. State-issued identity anchors inherit mature issuance and revocation infrastructure (ICAO eMRTDs carry issuing-state signatures over their contents [25, 26]) but, used naively, expose a lifelong government identifier to every verifier — the correlation handle par excellence.

Prson's thesis is that the dilemma dissolves under one design move: *anchor uniqueness privately; expose it only through audience-scoped derivations.* A government document anchors at most one

identity in the network (§7), but nothing derived from that anchor ever leaves the system except keyed-PRF images scoped to a single relying party (§8). Each verifier gets a stable, Sybil-resistant account key; no coalition of verifiers can compute the join. Uniqueness becomes a per-audience theorem rather than a global identifier.

1.3 Contributions

1. A deployed non-custodial key hierarchy with deterministic recovery and hardware-bound device factors (§4).
2. A graded identity-assurance ladder grounded in ICAO 9303 chip validation and on-device biometric correspondence (§5).
3. A notarial countersignature layer with a rotating, publicly enumerable keyring enabling offline, issuer-independent verification of every artefact the network produces (§6).
4. A Sybil-resistance design reconciling one-document-one-identity with cross-verifier unlinkability, including a per-domain unique-personhood credential (§7–8), with explicit security arguments and an explicit statement of the operator-coalition boundary (§14).
5. Practical selective disclosure: SD-JWT-VC with per-audience holder binding, and signed predicates over values that never persist server-side (§9).
6. An end-to-end encrypted, phishing-resistant login protocol with live-session semantics, plus app-to-app and sessionless transports under one delivery-pinning invariant (§10).
7. A capture-time provenance pipeline for photographs and video with challenge-based freshness and multi-party fulfilment (§11), and a PAdES signing PKI with legal-name binding (§12).
8. An interoperability layer implementing OpenID4VCI and OpenID4VP 1.0 Final, including verification of signed request objects under the `x509_san_dns` and `x509_hash` schemes with bundled trust anchors, exercised end-to-end against the EUDI reference verifier and the OpenID Foundation conformance suite (§13).

We follow Kerckhoffs's principle throughout: nothing here is secret, and every verification described is reproducible against key material published at `https://app.prson.net/.well-known/`.

2 Related work

2.1 Anonymous credentials and selective disclosure

Chaum [2] introduced transaction pseudonyms and the principle that organisations should see different, unlinkable identifiers for the same person. Brands [3] gave practically efficient certificates with selective disclosure of attributes; Camenisch and Lysyanskaya [5, 6] constructed multi-show unlinkable credentials (the Idemix lineage), and BBS/BBS+ signatures [4, 36] underpin current zero-knowledge credential efforts. Prson adopts the *goals* of this line but a different deployment point: credentials are IETF SD-JWT-VC [31] — salted-digest selective disclosure over standard JOSE signatures — combined with *per-audience credential issuance* and per-audience holder keys. This trades the elegant multi-show unlinkability of CL/BBS for primitives that today's verifier ecosystems, audit regimes, and HSMs already understand; §16 charts the migration path to BBS-class schemes as verifier support matures. The pairwise-identifier idea itself has standards precedent in OpenID Connect's pairwise subject identifiers [27] and SAML pairwise name-identifiers; Prson extends it from

login identifiers to *every* identifier and predicate that leaves the network, and derives holder keys per audience so even key material is pairwise.

2.2 Self-sovereign identity

The SSI movement [28, 29, 30] established the wallet/issuer/verifier triangle, data models for verifiable credentials [29], and decentralised identifiers [30]. Surveys [28] note recurring open problems: recovery, Sybil resistance, and issuer trust. Prson is an opinionated instantiation: recovery via deterministic derivation from a user-held mnemonic (no social or custodial recovery of keys); Sybil resistance via state-document anchoring (§7); issuer trust via a single transparent notary whose keys are public and whose assertions are deliberately limited to process provenance (§6).

2.3 Proof of personhood

Borge et al. [10] coined proof-of-personhood via physical ceremonies; Ford [11] surveys the design space and its tension with privacy. Iris-biometric registries [19] demonstrate planetary-scale uniqueness with a custodial biometric database. Prson takes the state-anchored corner of the design space, with two differentiators: the anchor registry stores salted hashes rather than identities or biometrics (§7), and uniqueness is only ever exposed pairwise (§8.3) — to our knowledge the first deployed system offering per-relying-party Sybil resistance with cross-relying-party unlinkability from government-document anchors.

2.4 Authentication and phishing resistance

Bonneau et al. [22] frame the password-replacement design space; FIDO2/WebAuthn [23] achieves phishing resistance by binding authentication to the RP origin and generating per-RP key pairs — properties Prson's login mirrors (origin pinning at session creation, per-domain holder keys) while additionally carrying verifiable attributes and live-session semantics (§10). The QR-initiated cross-device pattern inherits known risks of remote-initiation (consent phishing), addressed with server-stamped time, canonical request echo, and homograph rejection (§10.2).

2.5 Media provenance

Detection-based forensics [13, 14] and provenance-based approaches (C2PA [15]) are complements: the former examines pixels, the latter attaches signed history. C2PA explicitly scopes out claim-signer trustworthiness and capture-environment integrity. Prson composes C2PA-compatible on-device signing with platform attestation (environment integrity), verified-human gating (accountable capturer), notarial countersignature (process provenance and timestamping), and challenge-bound freshness (§11) — and documents the residual analog hole honestly (§14.3).

2.6 Electronic identity documents

ICAO Doc 9303 [25] defines eMRTD security: passive authentication binds data groups (including the chip portrait) to issuing-state signatures. Early analyses [26] catalogued skimming and traceability risks of contactless chips, motivating on-device, user-initiated reads — the model Prson follows: chips are read locally in the user's own device, validated locally and re-validated server-side via the signed transcript; no chip image or raw data group is retained server-side (§5).

3 Threat model and design goals

3.1 Adversaries

Adversary	Capabilities
A1 Malicious relying party (RP)	Crafts arbitrary requests; replays or transplants presentations; attempts over-collection; correlates its own sessions.
A2 Colluding RPs	Pool received identifiers, holder keys, predicates, timing and network metadata to link users across services.
A3 Network attacker	Observes/manipulates traffic; phishes via lookalike domains, tampered QR codes, malicious deep links; relays requests across origins.
A4 Media forger	Synthesises or re-captures media; injects frames via virtual cameras or repackaged apps; replays previously certified artefacts.
A5 Sybil attacker	Registers multiple verified identities from one document; fabricates documents; races or harasses legitimate holders via claim disputes.
A6 Device thief	Physical possession of a device, locked or unlocked; attempts key use or exfiltration; attempts wallet takeover via recovery flows.
A7 Compromised operator	Full read access to Prson's server-side storage and secrets, possibly sustained. Analysed explicitly in §14.2.

3.2 Design goals

- **G1 — No custody.** The operator must be architecturally unable to sign, present, or decrypt as a user.
- **G2 — Minimal server knowledge.** Server-side state must not suffice to reconstruct identity attributes, biometrics, or documents.
- **G3 — Uniqueness with unlinkability.** An RP can require one-unique-human-per-account; colluding RPs gain no computational advantage in deciding whether they share a user (formalised in §8.4).
- **G4 — Audience binding.** Every artefact released to an RP must be cryptographically useless at any other RP.
- **G5 — Offline, issuer-independent verification.** Verifying any artefact must require neither contacting Prson nor Prson's continued existence.
- **G6 — Capture-time authenticity.** Media provenance must be established at the moment of capture, not inferred afterwards.
- **G7 — Recoverability without escrow.** Device loss must not mean identity loss; recovery must not introduce a custodian.

Goals G1/G2/G5 are absolute (they hold even under A7); G3/G4 are computational, with the precise adversary coalition stated in §8.4 and §14.2; G6 is conditional on platform attestation, with the dependency stated in §14.3.

4 Key hierarchy and wallet model

4.1 Problem

Custodial identity providers concentrate signing authority: whoever holds the keys can impersonate every user, and subpoena, breach, or insider risk follows the concentration. Conversely, fully local keys without recovery convert device loss into identity death — the recovery problem the SSI literature repeatedly flags as unsolved in practice [28]. A third requirement is binding identity use to *enrolled hardware*, so that mnemonic exposure alone does not yield account takeover.

4.2 Construction

A wallet is created from a BIP-39 mnemonic [1] held only by the user. All long-lived user keys derive deterministically from it:

```
mnemonic ──► wallet root
             ├─► identity key  I_k   (Ed25519 [20], one per identity,
             │                        per-identity derivation path)
             └─► document-signing key D_k  (ECDSA P-256, dedicated
                                      sub-path; used only in §12)
```

Ed25519 [20] is used for identity and holder signatures; P-256 for the PAdES PKI where reader-ecosystem compatibility dictates the curve. Determinism yields G7: importing the mnemonic on new hardware re-derives every identity and document-signing key, so certificates and credentials bound to those keys survive device loss without any escrow. No server-side key material exists (G1).

Independently of the mnemonic, each installation generates a *device key* inside the platform's hardware-backed keystore (Apple Secure Enclave; Android StrongBox/TEE). The key is non-extractable — signing occurs inside the secure element — and is *not* derivable from the mnemonic. Sensitive operations require both factors: the identity key answers *who*, the device key answers *from which enrolled hardware*, and the (identity, device) binding is registered and signed at identity creation. This is the same architectural intuition that gives FIDO authenticators their strength [23], applied to wallet operations.

4.3 Recovery, rebinding, and takeover damping

Re-deriving a wallet on new hardware re-registers identities under a new device key; the old device's bindings cease to operate. Because recovery is powerful, it is deliberately slowed: wallet rebinding is subject to a multi-day transfer hold, during which the incumbent device is notified — a thief racing a victim cannot silently assume a wallet before the victim can react (A6). Paid entitlements transfer only after server-side validation of platform-store receipts, never on client assertion. Certified artefacts (credentials, PAdES certificates) survive migration because they bind to mnemonic-derived keys, while device-bound state (e.g. ephemeral per-site keys, §8.2) is intentionally lost — each class is assigned to the factor whose lifecycle matches its semantics.

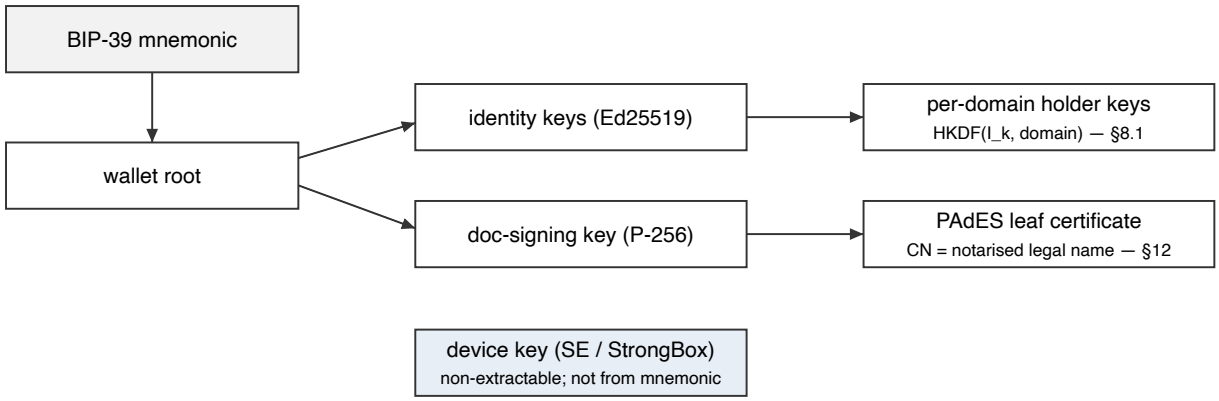

Figure 1. Key hierarchy. Everything left of the device key is deterministically recoverable from the mnemonic; the device key is a per-hardware second factor that never leaves the secure element.

5 The verification ladder

5.1 Problem

Binary "verified" badges force every use case to pay for (and every user to disclose toward) the maximum assurance any use case needs. NIST SP 800-63A [38] separates identity assurance into evidence classes precisely to avoid this; Prson encodes the separation in the identity object itself, so relying-party policy can demand exactly the assurance required and the wallet can refuse over-asking by default.

5.2 Construction

Level	Evidence	Credential semantics
0	None	Pseudonymous; credentials honestly carry `verified_human:false`; short validity; issued only when the RP explicitly opts in.
1–3	Verified contact points and liveness signals (email and phone via one-time codes; live selfie capture)	Weak personhood signals; suitable for anti-abuse, not assurance.
4	Government identity document captured and checked optically (MRZ consistency, security-feature heuristics)	Document-backed claims become available, marked as optical-tier.
5–6	eMRTD chip read over NFC: *passive authentication* per ICAO 9303 [25] — the Document Security Object's hashes over the data groups are validated against issuing-state document-signer certificates, binding contents (including the chip portrait, DG2) to the issuing authority; on-device biometric correspondence between chip portrait and live holder	"Verified human": state-anchored personhood; unlocks document-field credentials (§9.1), the unique-personhood credential (§8.3), and PAdES certificate issuance (§12).

5.3 Properties and rationale

- **Forgery cost at the top tier is state-PKI forgery.** Passive authentication makes document contents cryptographically attributable to the issuing state [25]; defeating it requires compromising issuing-state signing keys, not producing a better print. Known limitations of optical-only checks [26] are why levels 4 and 5–6 are distinguished rather than merged.
- **Biometrics never leave the device.** The chip-portrait/live correspondence runs on the phone. No template, embedding, or raw selfie is stored server-side (G2); compromise of the operator yields no biometric corpus (§14.2).
- **Attributes are stored as commitments.** Each verified property (name fields, birthdate, phone, postal address, website domain, bank-derived facts) is stored server-side as a *salted hash plus notary signature*, with per-type expiry; the raw value persists only on the device, disclosed per-event by the user. Re-presentation re-hashes the disclosed value against the commitment, so the server can verify without retaining.
- **Out-of-band evidence uses native challenge–response.** Phone numbers are verified by SMS codes, postal addresses by physical mail with a code, website domains by a DNS-01-style TXT challenge under `_prson-challenge.<domain>` [34] (validity capped by the domain's registration term where discoverable via RDAP), and bank facts via regulated open-banking access where the figure itself remains a device-side disclosed value (§9.2).

6 The notary: countersignatures and a public key registry

6.1 Problem

Verifier-side trust needs an anchor: who says this property was verified, this capture observed, this credential issued? Centralised issuers usually answer with an API — "ask us" — which couples every verification to the issuer's availability, business continuity, and observation (the issuer learns who verifies what, when — a privacy failure mode the W3C VC model calls issuer phone-home). The goal is an issuer whose assertions are verifiable forever, by anyone, without telling the issuer anything (G5).

6.2 Construction

- **Countersignatures.** A server-side *notary keyring* signs verification outcomes: property attestations, capture manifests, credential issuances, proof-package verdicts. Signing is over a canonical JSON serialisation (lexicographically sorted keys), reproducible byte-for-byte by independent implementations; open verifier libraries exist in JavaScript and Dart.
- **Key transparency.** The keyring is multi-key. Every key is identified by its RFC 7638 JWK thumbprint [33], recorded as `kid` in every signature. Rotation adds keys without invalidating history; the active and historical public keys are published at `/.well-known/jwks.json` and as a `did:web:app.prson.net` DID document [30] at `/.well-known/did.json`. Anyone may archive the key set; artefacts remain verifiable against archived keys after the operator's disappearance.
- **No phone-home.** Verification is a local signature check; there is no Prson API in any verify path, hence no issuer-side ledger of who verified what.
- **Bounded semantics.** A notary signature asserts provenance of *process* — "Prson's pipeline observed evidence E at time t" — and nothing more. It cannot fabricate a user's signature, because

the notary holds no user keys (G1); user-signed and notary-signed layers are verified independently.

7 Sybil resistance without a global identifier

7.1 Problem

Per Douceur [9], uniqueness requires an external anchor. The candidate anchors all carry costs (§1.2); having selected state documents, the remaining problem is to enforce *one document, one identity* without building the very correlation database the design exists to avoid: the naive implementation — index identities by document number — is a national-identifier registry with extra steps. Additionally, contested anchors are inevitable (family members sharing devices, stolen documents, re-verification after wallet loss), so the uniqueness mechanism needs adjudication semantics, not just an insert-if-absent rule.

7.2 Construction: the alias registry

When a document is verified, the wallet computes salted hashes of its stable identifiers. The server maintains the *alias registry*, keyed by these hashes, recording only which identity currently anchors the document. Contents are hash-valued under a private salt: the registry answers "is this document already anchored?" during a verification flow, but does not enumerate document numbers, names, or any attribute (G2). It is an internal *uniqueness oracle*; nothing derived from it is released except through the pairwise constructions of §8.

Contested claims enter an adjudication flow:

- **Evidence precedence.** A chip-validated (level 5–6) claim outranks an optical-only (level 4) claim; equal-tier claims favour the incumbent.
- **Notice and dispute window.** The current holder is notified and a fixed window applies before transfer settles.
- **Cooldown.** A rejected claimant faces a re-claim cooldown, damping oscillation and harassment-by-claim (A5).
- **Suspension during dispute.** While contested, derived properties are suspended for presentation — contested evidence cannot be double-used.

7.3 Analysis

Against A5: registering two identities from one document fails at the registry (one anchor per document); registering one identity from a fabricated document fails at level 5–6 on issuing-state signatures [25]; downgrading to optical tiers is visible in the credential's level and strength fields, so RPs requiring chip-tier assurance are unaffected by optical-tier forgeries. The takeover race (steal a document, claim it) is damped by precedence, notice, and cooldown, and ultimately resolves to the party able to repeatedly pass chip + biometric correspondence — i.e., the document's living subject.

8 Pairwise unlinkability constructions

8.1 Per-audience holder keys

The wallet never uses an identity key directly with a relying party. It derives a per-domain Ed25519 holder key via HKDF-SHA-256 [21, 35]:

> seed_d = HKDF(ikm = I_k-seed, salt = "prson-login-rekey/v1", info = UTF8(d), L = 32); HK_d = Ed25519.KeyGen(seed_d)

where *d* is the normalised RP host. The same user at the same domain always presents the same public key — a stable per-service account identifier, deterministic in the mnemonic and therefore migration-safe — while keys at distinct domains are independent PRF outputs. This is WebAuthn's per-RP key discipline [23] extended to credential holder binding.

8.2 Ephemeral mode

An RP may opt into *ephemeral identity*: the wallet generates a random holder key per enrolment, stored only device-locally and deliberately outside the mnemonic derivation. No stable identifier then exists even under later mnemonic compromise — retrospective de-anonymisation is impossible — traded against recoverability (loss of device is loss of that account), a trade-off surfaced to the user at consent time.

8.3 Pairwise attribute identifiers and the unique-personhood credential

Identifiers accompanying disclosed attributes (e.g. the value naming a verified property inside a signed predicate) are derived server-side with a keyed PRF — HMAC-SHA-256 [24, 32] under a server-held key *pepper* — over (scope, anchor, audience):

> $\text{pid}(a, d) = \text{HMAC}_{\text{pepper}}(\text{"scope"} \Vert \text{anchor}_a \Vert d)$ $\quad$ $\text{uid}(u, d) = \text{HMAC}_{\text{pepper}}(\text{alias}_u \Vert d)$

where $anchor_a$ is the internal value (property hash, document alias) and $alias_u$ the registry anchor of §7.2. The *unique-personhood credential* packages uid(u,d) in a notary-signed envelope naming the audience and validity window. Signatures over released artefacts include the audience in the signed payload, so an artefact minted for one host fails verification at any other (G4).

8.4 Security argument

> **Claim 1 (per-domain uniqueness).** For a fixed domain d, each verified document yields exactly one uid(·,d). *Argument*. The alias registry admits one identity per document (§7.2) and uid is a deterministic function of (alias, d). ∎

> **Claim 2 (cross-domain unlinkability).** No coalition of relying parties lacking the pepper and the mnemonic can distinguish whether uid(u,d_1) and uid(v,d_2), $d_1 \neq d_2$, belong to the same user, beyond negligible advantage. *Argument sketch.* HMAC-SHA-256 is a PRF under standard assumptions [32]; outputs on distinct inputs ($d_1 \neq d_2$ forces distinct inputs regardless of anchor equality) are computationally independent of the anchor bit by PRF security [8]. The same argument covers pid values, and HKDF's extract-then-expand PRF security [21, 35] covers holder keys HK_d. ∎

> **Claim 3 (trust boundary).** Claim 2's coalition excludes the pepper holder: the operator (or its compromiser, A7) can link pairwise identifiers *that it is subsequently shown*. This is the designed boundary — stated, not hidden. Mitigations and consequences are analysed in §14.2; note the pepper forges nothing (authenticity rests on notary signatures), so its compromise degrades unlinkability against an operator-inclusive coalition only.

Timing and network metadata remain a side channel for A2 coalitions (two RPs comparing login timestamps and source addresses); §14.3 discusses this honestly — cryptography removes the identifier join, not traffic analysis [17].

9 Selective disclosure and predicates

9.1 SD-JWT-VC credentials

Credentials follow IETF SD-JWT [31]: the notary signs a JWT in which each selectively disclosable claim is replaced by the digest of a salted disclosure; the holder releases, per presentation, only chosen disclosures, and the verifier checks digest membership — a hash-commitment selective-disclosure scheme over standard JOSE processing. Two credential types are issued: `verified-human` (personhood tier: level, strength, timestamps; issued from level 0 upward with honest semantics) and `verified-identity` (document-backed attribute claims, level ≥ 5; each disclosed field is re-verified against the notarised document transcript at issuance, so a credential cannot assert fields the document did not contain). Holder binding sets `cnf.jwk` to HK_d (§8.1); presentations append a Key-Binding JWT signed by HK_d over the audience, a verifier-chosen nonce, and a digest of the exact released presentation — defeating replay (A1) and audience transplantation (G4).

9.2 Predicates over undisclosed values

A public fragment catalog defines boolean facts — *age ≥ 18*, *document currently valid*, *nationality ∈ EU*, *bank balance ≥ X* — signable without disclosing the underlying value. The protocol for *time-variant* and *disclosed-value* facts:

1. the raw value (birthdate; bank figure) exists only on the device, with a salted commitment notarised at verification time (§5.3);
2. at signing time the wallet submits the value transiently; the server re-hashes against the commitment — a mismatched submission is rejected, so a holder cannot have a predicate signed over a value they never verified;
3. the server evaluates the predicate, signs the boolean with audience binding and pairwise identifiers (§8.3), and discards the value; nothing valued persists.

Relative to zero-knowledge predicates over CL/BBS credentials [5, 6, 36], this trades a transient server exposure for deployability with today's verifier ecosystems; §16 treats the ZK migration as the designated successor, removing step 2's exposure and adding multi-show unlinkability.

10 The login protocol

10.1 Problem

Federated login concentrates observation at the identity provider (every RP visit is logged at the IdP) and remains phishable where the binding between session and origin is human-checked [22]. The design targets: the operator must not see presentations (extending G2 to transport); the RP origin must be machine-checked, not user-checked (A3); credentials must be short-lived with live revocation; and the protocol must degrade gracefully across same-device, cross-device, native-app, and standards transports.

10.2 Construction

1. **Draft.** The RP page generates an ephemeral X25519 keypair [16] and registers a request blob (domain, nonce, required claims/ predicates, options) with a stateless edge relay. The relay rejects drafts whose claimed domain differs from the page's browser-enforced `Origin` header — page script cannot forge Origin, so a malicious page cannot run a login in another site's name (A3).
2. **Invocation.** The request renders as a QR code (cross-device) or universal link (same-device), carrying the blob and a *server-issued* timestamp. The wallet re-parses defensively: hosts containing invisible or bidirectional control codepoints are rejected outright (homograph hardening), and remaining non-ASCII letters are rendered with per-character classification so the user can spot confusables.
3. **Claim and anti-tamper.** The wallet signs a claim with the identity key over the request identifier, domain, and the server-issued timestamp (the wallet never trusts its own clock for protocol time); the relay returns the canonical stored request, which the wallet compares byte-for-byte against the scanned blob — a tampered QR (A3) is detected before any consent UI.
4. **Consent.** The wallet renders the verified domain, requested disclosures, assurance floor, and any sponsorship or uniqueness requirements; first-contact domains require explicit confirmation before biometric unlock. Identities below the floor are not offered (over-asking refused by default).
5. **Presentation.** The wallet mints a fresh SD-JWT-VC bound to HK_d, filters disclosures, signs the KB-JWT (audience = RP origin, nonce from the request), and delivers over a WebRTC data channel established peer-to-peer with the page, encrypted (NaCl box [16]) to the page's ephemeral key. The relay forwards opaque signaling only; presentations are end-to-end encrypted past the operator.
6. **Session.** Credentials live minutes; the wallet pushes signed renewals over the channel and can revoke instantly ("sign out everywhere"). Mid-session, the RP may request additional facts (predicate checks, document presence, form input), each gated by a fresh consent drawer.

RP-side verification is a local library: issuer signature against the published JWKS, disclosure digests, KB-JWT audience/nonce/digest binding, expiry, acceptance-tier policy. No Prson API stands in the login hot path (G5). Acceptance tiers span fully anonymous holders (per-site key, no claims), pseudonymous level-0, and verified tiers up to chip-validated personhood with the unique-personhood credential.

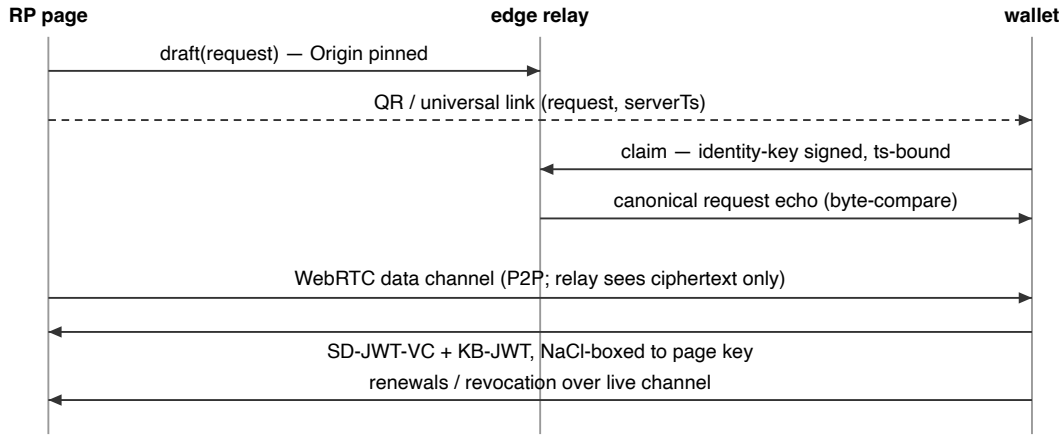

Figure 2. Login flow. The relay pins the claimed domain to the browser-enforced Origin and stores the canonical request for tamper detection; presentations are end-to-end encrypted between wallet and page.

10.3 Additional transports, one invariant

The trust core is reachable beyond the browser flow: (i) *app-to-app* — a native RP app opens its own login page in a system browser surface; after delivery the wallet returns the user via an HTTPS app link whose host must equal the login domain exactly, so the OS's app-link verification (domain-published association files) decides which app receives the return — package names are never audiences, keeping app and web identifiers identical; (ii) *sessionless wallet-action links* — one-shot requests (sign a document, prove a fact set) where, because no Origin-verified session exists, safety derives from *delivery pinning*: result endpoints must be HTTPS on the displayed domain, so a forged link can only deliver data to the domain the user saw and approved; and (iii) *OpenID4VP* (§13). Across all transports one invariant holds: *results are delivered only to the domain displayed to and approved by the user*.

11 Capture-time media provenance

11.1 Problem

Post-hoc detection loses the generative arms race [13, 14], and provenance standards deliberately scope out the two questions that decide evidentiary weight: was the capturing environment genuine, and is the capturer accountable [15]? A complete design must bind *content*, *time*, *environment integrity*, and *capturer* — and must additionally provide *freshness*, since a genuine artefact captured last year can be replayed as today's.

11.2 Construction

1. **Environment attestation.** Certified capture occurs in-app on hardware whose application and OS integrity is attested by the platform (Google Play Integrity; Apple App Attest). In production, failed attestation blocks certification: emulators, rooted devices, repackaged builds, and virtual-camera vectors that depend on such environments are excluded at the gate. The claim shifts from "pixels look authentic" to "a known build on intact hardware reported this capture path".
2. **Artefact binding.** Each photo/video is hashed (SHA-256) at capture; a C2PA-compatible manifest [15] is signed *on device* with device-held keys chaining to the Prson root (§12), gated on the

capturer's verified-human status — content is bound to a capture event and an accountable (pseudonymous, per §8) capturer.

3. **Notarial countersignature.** The notary countersigns a manifest binding content hash, capture time, optional user-consented location, and the verification verdict — adding process provenance and the public-keyring verifiability of §6.
4. **Freshness via challenges.** A verifier issues a *challenge*: typed slots (capture a photo of X; present property Y; sign predicate Z) bound to a challenge identifier. Responses are captured live against it — the cryptographic analogue of "hold up today's newspaper" — defeating replay of older genuine media (A4). Slots may be fulfilled by different identities and devices with per-responder attribution; partially fulfilled, multi-party evidence is a first-class object.

Verification recomputes the hash and checks both signature layers against published keys — in a browser verifier or any independent implementation, offline, indefinitely (G5). The screen-re-capture ("analog hole") residual is analysed in §14.3.

12 Document signing PKI (PAdES)

12.1 Problem

Document signing demands a different artefact class: signatures that validate in unmodified mainstream readers, attribute a *legal name* rather than a pseudonym, and survive certificate expiry. The constraints conflict with §8's pseudonymity by design — signing a contract is precisely the act where the user *wants* attribution — so the PKI is a separate, opt-in surface with its own key (D_k, §4.2), leaving login pseudonymity untouched.

12.2 Construction

- **Three-tier PKI.** An *offline* root CA; an online issuing CA; per-identity ECDSA P-256 leaf certificates. Root certificate at `/.well-known/identity-ca.cer`; revocation via CRL at `/.well-known/identity-ca.crl`.
- **Legal-name binding.** A leaf is issued only after the requested name fields are re-verified against the cryptographic transcript of the signer's chip-read document (§5): the certificate's CN is provably the notarised legal name, not free text.
- **Split signing, local keys.** The service prepares the PDF (signature field, byte range, CMS [40] signed attributes) and returns the exact bytes to sign; the device signs with the mnemonic-derived P-256 key; the service assembles the final CMS container with the chain and an RFC 3161 [41] timestamp token — PAdES B-T [37] — proving the signature predates later expiry or revocation. The private key never leaves the device (G1); documents are processed transiently and never stored.
- **Requestable remotely.** Third parties can request a signature via the sessionless links of §10.3 (document URL in, signed document delivered to the pinned endpoint), with the consent screen making the attribution consequence explicit.

13 Standards interoperability

13.1 Issuance: OpenID4VCI

The pre-authorised code flow [42] is implemented end-to-end: the wallet mints a single-use offer; any conformant third-party wallet redeems it at the token endpoint and presents a proof-of-possession JWT at the credential endpoint, receiving the same SD-JWT-VC the native pipeline issues, holder-bound to the presented key. Issuer and authorisation-server metadata are served under `/.well-known/`. Codes and tokens are single-use, short- lived, and stored only as hashes.

Issuer-key resolution is dual-stack from a single notary key: credentials can name the issuer as `did:web` [30] for DID-aware verifiers, or as an HTTPS issuer (`iss: https://app.prson.net`) resolved via the live `/.well-known/jwt-vc-issuer` metadata, with the signer's X.509 certificate chain embedded as an `x5c` header — so x5c-only verifier stacks in the eIDAS ecosystem validate the same credential without DID resolution. The signer leaf is issued under the Prson Identity Root CA (§12) with its SAN bound to the issuer host.

13.2 Presentation: OpenID4VP, with signed-request verification

The wallet accepts OpenID4VP 1.0 Final [43] authorisation requests — invoked via `openid4vp://` or the `haip-vp://` scheme used by EUDI/HAIP verifiers [44], from deep links or the in-app scanner — with DCQL queries, and responds via `direct_post` (`request_uri_method=post` is supported). Two signed-request client-identifier schemes are implemented. Under `x509_san_dns` the wallet verifies the request-object JWT signature (ES256/RS256) against the `x5c` leaf and requires the leaf's SAN dNSName to equal the client identifier; under `x509_hash` the client identifier must equal the base64url SHA-256 digest of the leaf certificate's DER encoding. In both schemes chain signatures and validity windows are validated, and the chain must terminate at a trust anchor bundled with the wallet per environment — today the Prson Identity Root CA (§13.4); eIDAS relying-party registers (§15) attach at the same point as they become operational. The verifier's domain is rendered on the consent screen with the same monospace, never-truncated, homoglyph-highlighting display used by the login protocol (§10).

Three request-level refusals harden the response channel: `redirect_uri` is rejected when the response mode is `direct_post`; requests carrying unsupported `transaction_data` types are rejected rather than silently ignored; and the response endpoint's host is bound to the verified leaf certificate (the SAN dNSName, generalising the delivery-pinning invariant of §10.3 to `x509_hash` requests) — so even a validly signed request cannot route a presentation away from the domain its certificate names.

13.3 Data model mapping

Credentials are SD-JWT-VC [31] with published type identifiers; keys are addressable via `did:web` [30] and JWKS; predicates exist both in a compact native encoding and a JWT-shaped encoding for uniform JOSE processing; the conceptual model maps onto the W3C Verifiable Credentials data model [29] for ecosystems that require it.

13.4 Verifier PKI and independent validation

Signed-request trust scales by configuration, not code: a *Prson verifier CA* issues per-relying-party request-signing certificates (EC P-256, SAN-bound to the verifier's domain) under the per-environment Issuing CA of the existing Identity Root (§12), emitting the request-object `x5c` chain and both client-

identifier forms. Any verifier holding such a certificate is accepted by every wallet with no per-RP wallet change, because the chain terminates at the bundled trust anchor.

The presentation stack has been validated against independent third parties, not only our own infrastructure. A self-hosted deployment of the **EUDI reference verifier** completed a full end-to-end run on a production device: `haip-vp` invocation, `x509_hash` signed-request verification with trust-anchor pinning, consent, fresh-credential minting under the HTTPS issuer with the `x5c` chain, `direct_post` delivery, and verifier-side acceptance of the SD-JWT-VC and its KB-JWT bound to the correct audience and nonce. The **OpenID Foundation conformance suite** wallet plan for OID4VP 1.0 Final (SD-JWT-VC, `direct_post`, signed requests, `x509_hash`) passed its happy-path and data-minimisation tests, and the wallet correctly refused the suite's adversarial requests — invalid request-object signatures, mismatched client identifiers, missing nonces, invalid client-identifier prefixes; the three additional findings the suite surfaced (the refusals enumerated in §13.2) were remediated and unit-tested. Notably, the suite's request-signing certificate was issued from our own Issuing CA, so the wallet validated a third party's signed requests against its bundled trust anchor with no special configuration — the verifier-CA model proving itself externally.

14 Security and privacy analysis

14.1 Threat coverage

Threat	Primary mitigations
A1 replay / transplant	KB-JWT audience + nonce + presentation-digest binding (§9.1); minutes-scale TTL; audience inside every signed fragment (§8.3).
A2 cross-RP correlation	Per-domain holder keys (§8.1); pairwise attribute ids and uids (§8.3, Claim 2); fresh credential per audience; no analytics SDKs in the wallet; no global identifier in any released envelope.
A3 phishing / tamper / relay	Origin pinning at draft; canonical-request byte comparison; server-issued protocol time; homograph rejection + per-character domain rendering; first-contact confirmation; delivery pinning for sessionless and app-to-app transports (§10.3).
A4 synthetic / replayed media	Attestation-gated capture; on-device C2PA signing; notarial manifest; challenge-bound freshness (§11).
A5 Sybil	State-PKI document validation; alias registry with precedence, notice, dispute window, cooldown, suspension (§7).
A6 device theft / takeover	Hardware-bound device factor; OS biometric gating; two-factor signing; rebinding transfer hold with incumbent notice (§4.3).

14.2 The compromised operator (A7)

We analyse the adversary most white papers omit. Full read access to Prson's server-side state yields: salted attribute hashes and signatures; the hash-valued alias registry; the pairwise pepper; notary private keys; operational metadata. With this, the adversary *can*:

- issue fraudulent notary attestations going forward — bounded by the transparency of the keyring model: keys are public, enumerable, and archivable, and every key introduction / activation / retirement / revocation is recorded in an append-only, hash-chained, signed *keyring transparency*

log (`/.well-known/keyring-log.json`, committed to git and optionally anchored in the public Sigstore Rekor log), so a silent key swap or backdating breaks the hash chain, the git history, and the Rekor anchor simultaneously; revocation and incident disclosure are therefore auditable events, and artefacts signed before compromise remain distinguishable by timestamp tokens;
- link pairwise identifiers *that it is subsequently shown* across audiences (Claim 3) — a degradation of unlinkability against the operator-inclusive coalition only.

The adversary *cannot*:

- recover raw attribute values, biometrics, document images, or bank figures — they are not stored (G2, §5.3);
- sign, present, or decrypt as any user — no user private keys exist server-side (G1);
- retroactively and undetectably alter signed history — artefacts are client-held, timestamped, and verify against archived public keyrings (G5);
- read past login presentations — they were end-to-end encrypted to ephemeral RP keys (§10.2) and never stored.

14.3 Honest dependencies and residual risks

- **Platform attestation.** Capture-environment claims (§11) inherit the security of Play Integrity / App Attest; their compromise degrades §11 to its signature layers. This is a vendor trust we state rather than obscure.
- **The analog hole.** A photograph *of a screen showing a fake* is a genuine capture of a fake scene. Challenge prompts (pose, context, multiple angles), capture-time metadata, and human review for high-stakes use narrow but do not eliminate this; no capture-time system eliminates it, and claims otherwise should be distrusted. The proof asserts capture circumstances, not scene semantics.
- **Issuing-state PKI.** Levels 5–6 inherit state trust [25] — a deliberate inheritance, with optical tiers explicitly labelled weaker.
- **Traffic analysis.** Pairwise identifiers remove the cryptographic join; correlated timing and source addresses across colluding RPs remain a statistical channel [17], partially mitigated by relay architecture and batching, not eliminated.
- **Mnemonic hygiene.** Everything recoverable is bounded by the user's custody of the mnemonic; ephemeral mode (§8.2) exists to cap the blast radius where users prefer it.
- **Secure elements.** Device-key non-extractability rests on SE/TEE guarantees; their compromise collapses the second factor to software strength.

15 Regulatory alignment

NIST SP 800-63 [38]. The ladder's evidence tiers are structured against 800-63A identity-evidence categories: chip-validated documents with on-device biometric binding align with IAL2 evidence requirements; hardware-bound, origin-pinned authentication aligns with AAL2 authenticator properties and the phishing-resistance discussion of 800-63B. We state alignment, not certification.

eIDAS 2.0 [39]. Formats and protocols track the European Digital Identity architecture: SD-JWT-VC, OpenID4VCI/VP including high-assurance signed requests, published issuer trust infrastructure

(`did:web`, JWKS, and x5c chains under a published root), and a designated extension point for relying-party trust lists as member-state registers become operational. Interoperability with the EUDI stack is demonstrated, not asserted: the full presentation flow has been validated end-to-end against the EUDI reference verifier (§13.4). Wallet recognition under Art. 5a and conformity assessment are roadmap items (§16).

GDPR [45]. Data minimisation (Art. 5(1)(c)) is implemented structurally: hash-only attribute storage, on-device biometrics, pairwise identifiers, per-event disclosure, and absence of behavioural analytics in the wallet (crash diagnostics are user-disablable). Yumankind OÜ is an Estonian (EU) controller; processing runs in EU regions; the subprocessor list is published in the privacy policy, and conformance documentation (data inventory, cryptographic inventory, DPIA support) is available to partners.

16 Limitations and future work

- **Zero-knowledge predicates and multi-show unlinkability.** Migrating predicates and credentials to BBS-class schemes [4, 36] would remove the transient disclosed-value submission (§9.2) and the fresh-issuance linkage surface at the issuer, at the cost of verifier-ecosystem maturity; this is the designated successor design.
- **Verifier trust lists.** Signed-request verification (§13.2) pins certificate identity and chain-to-bundled-anchor but not register membership; eIDAS relying-party registers and external access-certificate authorities will be added alongside the bundled anchors as they become operational, adding entitlement enforcement ("registered to ask for age, nothing more").
- **Formal verification.** The login protocol's core has been machine-checked in the symbolic (Dolev–Yao) model with ProVerif [47]: under an attacker controlling the QR and peer-to-peer transport, with the relay's TLS channels authentic, we prove presentation secrecy, domain-binding authentication (the anti-phishing property), injective agreement (anti-replay of a session), and injective end-to-end replay resistance of the presentation; a negative control without Origin-pinning correctly exhibits the attacks. The model and outputs are published (`login-sdk/formal/`). Claims 1–3 (§8.4) remain informal reductions; a computational-model proof and a Tamarin cross-check of the uniqueness and dispute protocols are future work, and independent analysis is invited.
- **Revocation transparency.** The notary keyring now publishes an append-only, hash-chained, signed transparency log of all key lifecycle events (§14.2); a status-list mechanism for issued credentials and an OCSP responder for the PAdES PKI are in progress, as is surfacing keyring-log verification in the browser proof verifier and routine Rekor anchoring.
- **Post-quantum readiness.** Ed25519, P-256, and X25519 are not post-quantum secure; NIST's selected schemes [46] (ML-KEM, ML-DSA, SLH-DSA) define the migration targets. The keyring and `kid` discipline (§6) were designed so signature-suite rotation is an additive, auditable event.
- **Format breadth.** ISO mdoc/mDL presentation and encrypted OpenID4VP responses are not yet implemented.
- **Certification.** The OpenID Foundation conformance suite wallet plan has been run with all genuine findings remediated, and the EUDI reference verifier accepts the wallet's presentations end-to-end (§13.4); the formal OpenID Certified listing and EUDI wallet conformity assessment remain planned milestones — §13 and §15 describe validated implementation status, not yet third-party certification.

17 Conclusion

Prson demonstrates that the apparent trilemma between Sybil resistance, privacy, and practicality dissolves under a single design move: anchor uniqueness privately, and expose identity, attributes, and personhood only through audience-scoped, independently verifiable derivations. Keys stay with people; raw data stays on devices; every assertion is a signature anyone can check against public keys, offline, indefinitely. The operator is reduced to two auditable roles — a notary whose keys are public and whose claims are bounded to process provenance, and a uniqueness oracle whose outputs are pairwise by construction — and the paper states precisely what compromising that operator would and would not yield. We publish the architecture in full because systems that ask for trust should make their guarantees, dependencies, and failure modes checkable. Everything described here can be exercised today at `https://app.prson.net`.

References

[1] M. Palatinus, P. Rusnak, A. Voisine, S. Bowe. *BIP-39: Mnemonic code for generating deterministic keys*. Bitcoin Improvement Proposals, 2013.

[2] D. Chaum. *Security without identification: transaction systems to make Big Brother obsolete*. Communications of the ACM 28(10):1030–1044, 1985.

[3] S. Brands. *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy*. MIT Press, 2000.

[4] D. Boneh, X. Boyen, H. Shacham. *Short group signatures*. CRYPTO 2004, LNCS 3152, Springer.

[5] J. Camenisch, A. Lysyanskaya. *An efficient system for non-transferable anonymous credentials with optional anonymity revocation*. EUROCRYPT 2001, LNCS 2045, Springer.

[6] J. Camenisch, A. Lysyanskaya. *Signature schemes and anonymous credentials from bilinear maps*. CRYPTO 2004, LNCS 3152, Springer.

[7] C. Paquin, G. Zaverucha. *U-Prove cryptographic specification V1.1*. Microsoft Research, 2013.

[8] O. Goldreich, S. Goldwasser, S. Micali. *How to construct random functions*. Journal of the ACM 33(4):792–807, 1986.

[9] J. R. Douceur. *The Sybil attack*. IPTPS 2002, LNCS 2429, Springer.

[10] M. Borge, E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, B. Ford. *Proof-of-personhood: redemocratizing permissionless cryptocurrencies*. IEEE EuroS&P Workshops, 2017.

[11] B. Ford. *Identity and personhood in digital democracy: evaluating inclusion, equality, security, and privacy in pseudonym parties and other proofs of personhood*. arXiv:2011.02412, 2020.

[12] R. Chesney, D. K. Citron. *Deep fakes: a looming challenge for privacy, democracy, and national security*. California Law Review 107:1753–1820, 2019.

[13] L. Verdoliva. *Media forensics and deepfakes: an overview*. IEEE Journal of Selected Topics in Signal Processing 14(5):910–932, 2020.

[14] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, M. Nießner. *FaceForensics++: learning to detect manipulated facial images*. ICCV 2019.

[15] Coalition for Content Provenance and Authenticity. *C2PA technical specification*, v2.x, 2024–2026.

[16] D. J. Bernstein, T. Lange, P. Schwabe. *The security impact of a new cryptographic library*. LATINCRYPT 2012, LNCS 7533, Springer.

[17] A. Narayanan, V. Shmatikov. *Robust de-anonymization of large sparse datasets*. IEEE Symposium on Security and Privacy, 2008.

[18] L. Sweeney. *k-anonymity: a model for protecting privacy*. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10(5):557–570, 2002.

[19] Worldcoin Foundation. *The Worldcoin protocol*. White paper, 2023.

[20] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, B.-Y. Yang. *High-speed high-security signatures*. Journal of Cryptographic Engineering 2(2):77–89, 2012. See also RFC 8032.

[21] H. Krawczyk, P. Eronen. *HMAC-based extract-and-expand key derivation function (HKDF)*. RFC 5869, IETF, 2010.

[22] J. Bonneau, C. Herley, P. C. van Oorschot, F. Stajano. *The quest to replace passwords: a framework for comparative evaluation of web authentication schemes*. IEEE Symposium on Security and Privacy, 2012.

[23] W3C. *Web Authentication: an API for accessing public key credentials (WebAuthn)*, Level 2, W3C Recommendation, 2021.

[24] M. Bellare, R. Canetti, H. Krawczyk. *Keying hash functions for message authentication*. CRYPTO 1996, LNCS 1109, Springer.

[25] ICAO. *Doc 9303: Machine Readable Travel Documents*, 8th edition, Part 11: Security mechanisms for MRTDs, 2021.

[26] A. Juels, D. Molnar, D. Wagner. *Security and privacy issues in e-passports*. SecureComm 2005, IEEE.

[27] N. Sakimura, J. Bradley, M. Jones, B. de Medeiros, C. Mortimore. *OpenID Connect Core 1.0* (pairwise subject identifiers, §8). OpenID Foundation, 2014.

[28] A. Mühle, A. Grüner, T. Gayvoronskaya, C. Meinel. *A survey on essential components of a self-sovereign identity*. Computer Science Review 30:80–86, 2018. See also C. Allen, *The path to self-sovereign identity*, 2016.

[29] M. Sporny, D. Longley, D. Chadwick, et al. *Verifiable credentials data model*, v2.0. W3C Recommendation, 2025.

[30] M. Sporny, A. Guy, M. Sabadello, D. Reed. *Decentralized identifiers (DIDs) v1.0*. W3C Recommendation, 2022.

[31] D. Fett, K. Yasuda, B. Campbell. *Selective disclosure for JWTs (SD-JWT)*; O. Terbu, D. Fett, B. Campbell. *SD-JWT-based verifiable credentials (SD-JWT VC)*. IETF OAuth Working Group, 2024–2025.

[32] M. Bellare. *New proofs for NMAC and HMAC: security without collision-resistance*. CRYPTO 2006, LNCS 4117, Springer.

[33] M. Jones, N. Sakimura. *JSON Web Key (JWK) thumbprint*. RFC 7638, IETF, 2015.

[34] R. Barnes, J. Hoffman-Andrews, D. McCarney, J. Kasten. *Automatic certificate management environment (ACME)*. RFC 8555, IETF, 2019.

[35] H. Krawczyk. *Cryptographic extraction and key derivation: the HKDF scheme*. CRYPTO 2010, LNCS 6223, Springer.

[36] T. Looker, V. Kalos, A. Whitehead, M. Lodder. *The BBS signature scheme*. IRTF CFRG Internet-Draft, 2023–2025.

[37] ETSI. *EN 319 142: electronic signatures and infrastructures — PAdES digital signatures*, 2016.

[38] NIST. *SP 800-63-3 digital identity guidelines* (63A: enrollment and identity proofing; 63B: authentication), 2017; revision 4, 2024–2025.

[39] European Union. *Regulation (EU) 2024/1183 amending Regulation (EU) No 910/2014 as regards establishing the European Digital Identity Framework*, 2024; and the EUDI Wallet Architecture Reference Framework, 2023–2026.

[40] R. Housley. *Cryptographic message syntax (CMS)*. RFC 5652, IETF, 2009.

[41] C. Adams, P. Cain, D. Pinkas, R. Zuccherato. *Internet X.509 PKI time-stamp protocol (TSP)*. RFC 3161, IETF, 2001.

[42] T. Lodderstedt, K. Yasuda, T. Looker. *OpenID for verifiable credential issuance (OpenID4VCI)*. OpenID Foundation, 2024–2025.

[43] O. Terbu, T. Lodderstedt, K. Yasuda, D. Fett. *OpenID for verifiable presentations (OpenID4VP) 1.0*. OpenID Foundation, 2025.

[44] OpenID Foundation. *High assurance interoperability profile (HAIP)*. Digital Credentials Protocols WG, 2025.

[45] European Union. *Regulation (EU) 2016/679 (General Data Protection Regulation)*, 2016.

[46] NIST. *FIPS 203 (ML-KEM), FIPS 204 (ML-DSA), FIPS 205 (SLH-DSA): post-quantum cryptography standards*, 2024.

[47] B. Blanchet. *Modeling and verifying security protocols with the applied pi calculus and ProVerif*. Foundations and Trends in Privacy and Security 1(1–2):1–135, 2016.

© 2026 Yumankind OÜ. This document may be redistributed verbatim. Verification endpoints referenced herein: `app.prson.net/.well-known/{jwks.json, did.json, identity-ca.cer, identity-ca.crl,`

`security.txt}` and the public verifier at `app.prson.net/verify`. Questions and challenges to any claim: hello@yumankind.com.