Zero-Knowledge Selective Disclosure for eIDAS 2.0: A ZK-Native Architecture for the EU Digital Identity Wallet

AffixIO Research June 2026 WP-017 Download PDF

The EUDI Wallet rollout needs selective disclosure that scales across 27 member states. SD-JWT works, but linkability bites at scale. We outline a ZK-native wallet architecture with predicate proofs for age thresholds and attribute claims.

CONTENTS

1. eIDAS 2.0 and the Digital Identity Mandate

eIDAS 2.0, formally Regulation (EU) 2024/1183, entered into force on 20 May 2024 and represents the most significant reform of EU digital identity infrastructure since the original eIDAS Regulation of 2014. The core obligation is unambiguous: every EU member state must provide citizens, residents, and businesses with a certified European Digital Identity Wallet by late 2026. The wallet must be free to obtain, cross-border interoperable, and capable of storing government-verified credentials, professional qualifications, payment instruments, and other attestations.

The regulatory timeline makes this one of the most immediate identity technology programmes in European history. Specified private sector services, including very large online platforms, qualified trust service providers, and providers of services in banking, finance, and transport, must accept EUDI Wallets for strong authentication by late 2027. This creates a 12-to-18 month window in which both wallet infrastructure and acceptance infrastructure must be built and certified simultaneously.

The privacy implications of this programme are significant. The EUDI Wallet will store credentials that include sensitive personal attributes: dates of birth, addresses, professional qualifications, medical information, criminal records, and financial data. The architecture must ensure that presenting credentials from the wallet for one purpose does not inadvertently disclose attributes intended for another purpose, and that different relying parties cannot collude to correlate a user's wallet usage across contexts.

---

2. Selective Disclosure: The Privacy Core

Selective disclosure is the technical capability that enables a holder to present only the specific attributes required by a verifier, without revealing other attributes in the same credential. It is a foundational privacy principle in the EUDI Wallet architecture, mandated by the General Data Protection Regulation's data minimisation principle (Article 5(1)(c)), by the specific data minimisation obligations in the eIDAS 2.0

implementing acts, and by the broader principle of privacy by design embedded in the architecture.

In practice, selective disclosure addresses a problem that has plagued digital identity systems for decades: the mismatch between what a credential contains and what a verifier needs. A national identity card contains a full name, date of birth, address, nationality, and document number. If a verifier needs only to confirm that the holder is over 18, presenting the full identity card discloses five attributes that are not needed. Each unnecessary disclosure is a privacy risk, a data minimisation violation, and a potential GDPR liability.

Two Approaches to Selective Disclosure

There are two fundamentally different approaches to selective disclosure in digital credentials. The first is selective attribute revelation: the credential contains multiple attributes, and the holder selects which attributes to reveal to the verifier. The verifier receives the revealed attributes in cleartext and a proof that they were disclosed from a valid credential. The second approach is predicate proofs: the holder proves a statement about an attribute, such as "my age is above 18" or "my income is above 30,000 EUR per year", without revealing the underlying attribute value. The verifier learns only the truth of the predicate, not the attribute itself.

SD-JWT implements the first approach. ZK proofs implement the second. The difference matters significantly for high-privacy use cases.

---

3. SD-JWT: What the ARF Specifies

The EUDI Wallet Architecture and Reference Framework (ARF) specifies Selective Disclosure for JSON Web Tokens (SD-JWT) as the primary mechanism for selective disclosure in the wallet. SD-JWT is a straightforward extension of standard JWT: credential attributes are individually hashed and included in the JWT; the holder presents only the hashes for attributes they wish to disclose, together with the salt

values needed to verify those hashes. The verifier reconstructs and verifies only the disclosed attributes.

SD-JWT has meaningful practical advantages that justify its selection as the ARF baseline. It is relatively simple to implement using existing JWT libraries. It is compatible with existing PKI infrastructure. It supports incremental adoption: issuers that already issue JWTs can migrate to SD-JWT with modest engineering effort. The W3C specification for SD-JWT Verifiable Credentials provides a standard credential format that wallet implementations can target.

The ARF explicitly acknowledges that more advanced mechanisms, including BBS+ signatures, selective disclosure predicates, and ZK-SNARKs, may provide stronger privacy guarantees in specific use cases. This acknowledgement creates space for ZK-native implementations that go beyond the SD-JWT baseline while remaining within the ARF's architectural intent.

---

4. SD-JWT Limitations for High-Privacy Use Cases

SD-JWT's selective attribute revelation approach has four limitations that become significant in high-privacy use cases.

Linkability Across Sessions

When a holder presents an SD-JWT credential to a verifier, the verifier receives the JWT's signature, which is constant across all presentations of the same credential. Two relying parties that collude can determine that they verified the same credential, enabling cross-context user tracking. This is the linkability problem. SD-JWT does not prevent it; it merely limits the attributes disclosed in each presentation.

Attribute-Level Disclosure

SD-JWT reveals attribute values to verifiers. If a user presents their date of birth to prove they are over 18, the verifier learns their exact date of birth, not merely that the age threshold is met. In data minimisation terms, the disclosed date of birth is more

information than the verifier needs, and therefore its disclosure is a GDPR data minimisation violation relative to what a predicate proof would require.

Issuance Unlinkability

The SD-JWT credential is signed by the issuer at issuance. When the holder presents the credential to a verifier, the issuer's signature is included. If the issuer and verifier collude, or if the issuer logs issued credentials, the issuer can determine which verifier received a presentation from a specific holder. ZK-native approaches using blind issuance protocols prevent this.

Predicate Support

SD-JWT does not natively support predicate proofs. Proving "age above 18" requires revealing the date of birth attribute; there is no SD-JWT mechanism for proving a comparison predicate without disclosing the comparand. BBS+ signatures with predicate extensions provide some predicate capability, but the implementation is significantly more complex than the base SD-JWT specification.

---

5. ZK-Native Selective Disclosure: The Alternative

ZK-native selective disclosure replaces the selective attribute revelation model with a predicate proof model. The credential attributes are committed to a cryptographic commitment scheme rather than stored as hashed attributes. The holder then generates zero-knowledge proofs about the committed attributes, proving statements of the form "the committed value satisfies predicate P" without revealing the committed value.

ZK vs SD-JWT: The Core Difference

SD-JWT: holder reveals attribute_value; verifier learns attribute_value. ZK proof: holder proves predicate(attribute_value) == true; verifier learns only the truth of the

predicate. For age verification: SD-JWT reveals date_of_birth; ZK proof reveals only (current_date - date_of_birth) > threshold.

Unlinkable Presentations

ZK proofs of credential attributes can be made unlinkable using blind signature schemes or commitment rerandomisation. Each presentation of the same credential generates a cryptographically independent proof; colluding verifiers cannot determine that two presentations came from the same holder. This provides a fundamentally stronger privacy guarantee than SD-JWT presentation linkability allows.

Issuer-Holder Separation

With blind issuance protocols, the issuer signs a commitment to the holder's attributes without learning which holder the credential is being issued to. The holder can then generate proofs using this blinded credential without the issuer being able to correlate presentations with issuances. This protection against issuer-verifier collusion is not available in the SD-JWT model.

---

6. Predicate Proofs for Identity Attributes

The most practically significant advantage of ZK-native selective disclosure is support for predicate proofs: proving statements about attributes without revealing the attributes themselves.

Age Threshold Proofs

The most common predicate in digital identity is the age threshold: the holder proves they are above a specified age without revealing their exact date of birth. AffixIO's constraint language age threshold circuit takes the committed date of birth and the current date as inputs and produces a proof that the difference exceeds the threshold. The verifier learns only the boolean result; the exact date of birth remains private.

Income and Financial Threshold Proofs

Financial services applications frequently require proof that an account holder's income or net worth exceeds a threshold required for a specific product. ZK predicate proofs enable this without disclosing exact financial figures to the service provider. The income is committed in a credential issued by a tax authority or payroll service; the holder proves the threshold is met without revealing the income amount.

Professional Qualification Proofs

Rather than revealing a specific professional qualification in full, a holder can prove that they hold a qualification in a specified category: "I hold a medical qualification registered in an EU member state" without revealing which qualification, which institution, or which member state. This is particularly relevant for cross-border professional recognition under the EU's single market framework.

Compound Predicate Proofs

Multiple predicates can be combined in a single ZK proof. A user can prove simultaneously that they are over 18, resident in an EU member state, and hold a valid payment credential, without revealing their date of birth, specific country, or payment credential details. Compound proofs reduce the number of presentation interactions required for complex verification scenarios, improving user experience while strengthening privacy.

---

7. AffixIO constraint language Identity Circuits for EUDI Wallet

AffixIO has developed a suite of policy circuits implementing ZK-native selective disclosure for EUDI Wallet credential types. These circuits are designed to be compatible with the EUDI Wallet ARF's credential schema, enabling ZK-native presentations alongside or in replacement of SD-JWT presentations.

The identity_age_threshold Circuit

This circuit proves that the holder of a committed national identity credential is above a specified age threshold. It takes the credential commitment, the issuing authority's public key, the current date (as a public input), and the threshold age (as a public input) and produces a proof without revealing the date of birth or any other identity attribute.

The identity_residency Circuit

This circuit proves EU residency without revealing the specific member state. It is designed for use cases where the verifier needs to confirm that the holder is an EU resident for regulatory purposes, but where revealing the specific member state would constitute unnecessary disclosure. The circuit verifies that the committed address attribute falls within the set of EU member state codes.

The professional_qualification Circuit

This circuit proves membership in a specified professional category, such as "licensed healthcare professional" or "registered legal practitioner", without revealing the specific qualification, the issuing institution, or the licence number. The professional category is derived from a classification commitment embedded in the credential at issuance.

Circuit Composition for EUDI Wallet Presentations

In a full EUDI Wallet presentation flow, multiple circuits are composed into a single presentation proof. The holder's wallet software runs the required circuits locally, generating individual proofs for each required attribute, and then produces a combined presentation proof that the verifier validates in a single verification step. This is computationally efficient and provides a user experience comparable to a standard credential presentation.

---

8. Credential Issuance and the ZK Wallet Flow

The ZK-native EUDI Wallet flow differs from the SD-JWT flow primarily in the issuance step, where the attribute commitment scheme is established.

Issuance: Commitment Generation

During credential issuance, the holder's wallet software generates attribute commitments using a Pedersen commitment scheme. The commitment for each attribute is computed as C = g^v * h^r, where v is the attribute value, r is a random blinding factor known only to the holder, and g, h are generator points on the BN254 elliptic curve. The issuer signs the set of attribute commitments rather than the attribute values themselves.

Presentation: Proof Generation

During credential presentation, the wallet software uses the attribute values and blinding factors as private witnesses in the policy circuit. The circuit verifies that the committed values satisfy the required predicates and produces a ZK proof. The proof, together with the issuer's signature on the commitment set, constitutes the credential presentation. The verifier checks the issuer signature and the ZK proof; neither step requires the verifier to learn any attribute value.

Revocation Without Linkability

Credential revocation in ZK-native systems uses accumulator-based revocation rather than revocation lists. The issuing authority maintains a cryptographic accumulator, a commitment to the set of non-revoked credentials. The holder generates a proof of accumulator membership, proving that their credential is included in the accumulator without revealing their credential's identifier. This prevents the linkability that would arise if revocation checks required revealing a credential identifier to the verifier.

---

9. Interoperability: ISO 18013-5, W3C DID, AnonCreds

The EUDI Wallet must interoperate with multiple existing credential standards, each of which has different privacy properties and ZK compatibility characteristics.

ISO 18013-5 Mobile Driving Licence

ISO 18013-5, the mobile driving licence standard, uses the Mobile Security Object format and is explicitly included in the EUDI Wallet ARF as a credential format. The standard supports selective disclosure at the attribute level but does not natively support predicate proofs. AffixIO's approach wraps ISO 18013-5 credentials in a ZK commitment at issuance, enabling predicate proofs over the mDL attributes while maintaining compatibility with verifiers that accept the native ISO 18013-5 format.

W3C Decentralised Identifiers

The W3C DID specification provides a framework for self-sovereign, decentralised identity. AffixIO's ZK circuits are compatible with W3C DID-based credential issuance: the issuer's public key is resolved from a DID document, and the credential commitments are anchored to the holder's DID. This enables ZK-native credential proofs within the W3C Verifiable Credentials data model.

AnonCreds

AnonCreds, originally developed for the Hyperledger Indy ecosystem, provides ZK-native credential proofs with unlinkability and predicate support using Camenisch-Lysyanskaya signatures. AffixIO's policy circuits implement compatible predicate logic, enabling interoperability between AnonCreds-issued credentials and AffixIO-verified proofs, which is particularly relevant for member states that have piloted AnonCreds-based identity schemes.

---

10. Privacy-by-Design and GDPR Alignment

ZK-native selective disclosure provides a stronger implementation of GDPR's privacy-by-design principle than SD-JWT for several specific requirements.

Data Minimisation (Article 5(1)(c))

GDPR's data minimisation principle requires that personal data be "adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed." SD-JWT selective disclosure reduces the number of attributes disclosed

per presentation but still discloses attribute values for the selected attributes. ZK predicate proofs further minimise disclosure by revealing only predicate truth values rather than attribute values. For age threshold verification, a ZK proof discloses strictly less information than an SD-JWT presentation of the date of birth attribute.

Purpose Limitation (Article 5(1)(b))

ZK predicate proofs are inherently purpose-limited: the predicate committed in the proof circuit is specific to a stated purpose. A proof generated for an age threshold of 18 for alcohol purchase cannot be repurposed to prove a different age threshold for a different context without generating a new proof. This technical purpose limitation complements the organisational purpose limitation required by GDPR.

The GDPR Article 25 Nexus

GDPR Article 25 requires data protection by design and by default: controllers must implement appropriate technical measures to integrate data protection principles into processing. ZK-native selective disclosure in the EUDI Wallet satisfies Article 25 at the cryptographic layer: data minimisation is enforced by the proof system, not merely claimed in a privacy policy. This is a stronger compliance posture than can be achieved with purely organisational measures.

---

11. Deployment Architecture and Member State Integration

Deploying ZK-native selective disclosure within the EUDI Wallet architecture requires changes at three layers: the wallet application, the credential issuance infrastructure, and the relying party verification stack.

Wallet Application Requirements

The wallet application must be capable of running policy circuits locally. Proof generation for simple predicate circuits takes between 200 and 800 milliseconds on a modern mobile device, which is acceptable for interactive credential presentation

flows. The wallet must store blinding factors securely in the device's secure enclave, alongside the attribute values themselves.

Issuer Infrastructure

Credential issuers, primarily government identity authorities and recognised trust service providers, must generate attribute commitments at the time of credential issuance rather than signing attribute values directly. This requires a modest change to the issuance API and credential format, but the change is backward compatible: issuers can issue both SD-JWT credentials and ZK-commitment credentials from the same infrastructure, allowing a phased transition.

Relying Party Verification

Relying parties must be capable of verifying policy circuit proofs. AffixIO provides a verification SDK compatible with the EUDI Wallet presentation protocol, implementing proof verification in JavaScript, Python, and Java to cover the principal relying party technology stacks. Verification time for a composite predicate proof is under 50 milliseconds in all tested environments.

---

12. Conclusions

eIDAS 2.0 creates the largest mandatory digital identity deployment in history. All 27 EU member states must issue EUDI Wallets by late 2026; hundreds of millions of European citizens will hold digital credentials that enable online and in-person identity verification. The privacy properties of this infrastructure matter at a societal scale.

SD-JWT, the ARF's baseline selective disclosure mechanism, provides a meaningful improvement over non-selective credential disclosure, but it reveals attribute values rather than predicate truths, it links presentations through constant credential signatures, and it cannot prevent issuer-verifier collusion. For the highest-privacy use cases, including age threshold verification, professional qualification confirmation, and financial eligibility proofs, ZK-native selective disclosure provides materially stronger privacy guarantees.

AffixIO's Noir-based identity circuits implement ZK-native selective disclosure within the EUDI Wallet architecture, providing unlinkable presentations, predicate proofs, and issuer-holder separation without abandoning interoperability with the ARF's credential format requirements. The circuits are available under an open-source MIT licence, with documentation targeting member state identity authorities and EUDI Wallet implementers.

Related reading

- WP-009: Privacy-Preserving Age Verification with Zero-Knowledge Proofs
- WP-006: PII-Free KYC by Design with Zero-Knowledge Identity Circuits
- WP-014: Double-Spend Prevention for Zero-Knowledge Proofs

Frequently asked questions

What is selective disclosure?

Presenting only the attributes a service needs, such as proving age over 18 without sharing your full date of birth.

How is ZK different from SD-JWT?

ZK proofs can show predicates and hide correlatable identifiers, reducing cross-service tracking when the same credential is reused.

Is this compatible with eIDAS 2.0?

The architecture maps to EUDI Wallet issuance and presentation flows while adding predicate circuits for common eligibility checks.

All White Papers

▶ About

▶ Solutions

▶ Legal

▶ Trust & Security

Contact

truth layer | yes | no | proof