PAXPORT: Analysis of the Spanish Age Verification System

David Balbás, Diego Castejón

11th November 2025

IMDEA Software Institute

Outline

- Digital identity systems
- PAXPORT: the Spanish age verification system
  - Design
  - Problems
  - Solutions
- The EUDI Wallet and Next Steps

Digital Identity Systems

The Problem

- **Digital identity systems** (EUDI wallet) will be deployed soon. Advantages are prominent:

The Problem

- **Digital identity systems** (EUDI wallet) will be deployed soon. Advantages are prominent:
  - Scam and fraud prevention.
  - Digital agility and interoperability within EU countries.
  - Increased privacy and security.

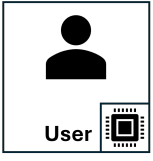
The Problem

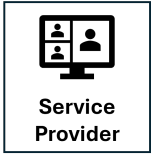
- **Digital identity systems** (EUDI wallet) will be deployed soon. Advantages are prominent:
  - Scam and fraud prevention.
  - Digital agility and interoperability within EU countries.
  - Increased privacy and security.
- *Privacy and security* are central to system design:
  - Selective disclosure / data minimization.
  - Unlinkability across service providers.
  - Non-transferability / unforgeability.
  - Anonymity / pseudonymity.
  - ...

The Setting

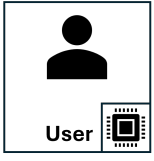

Registration
Authority

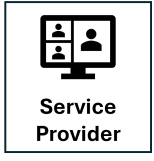

Credential
Issuer


User


Service
Provider

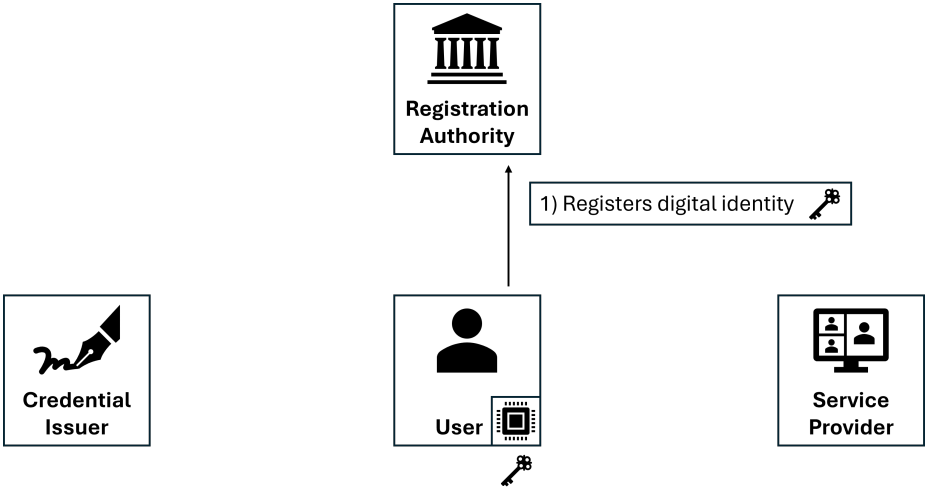
The Setting



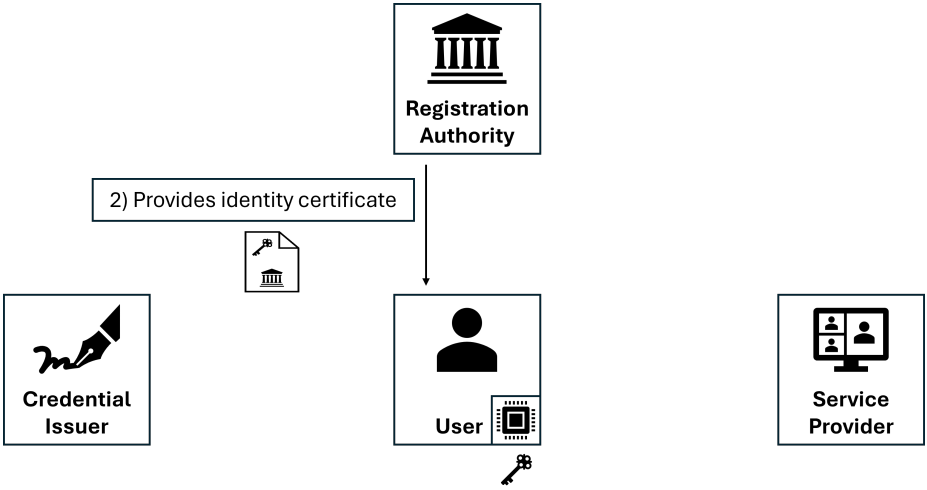


0) Generates digital identity

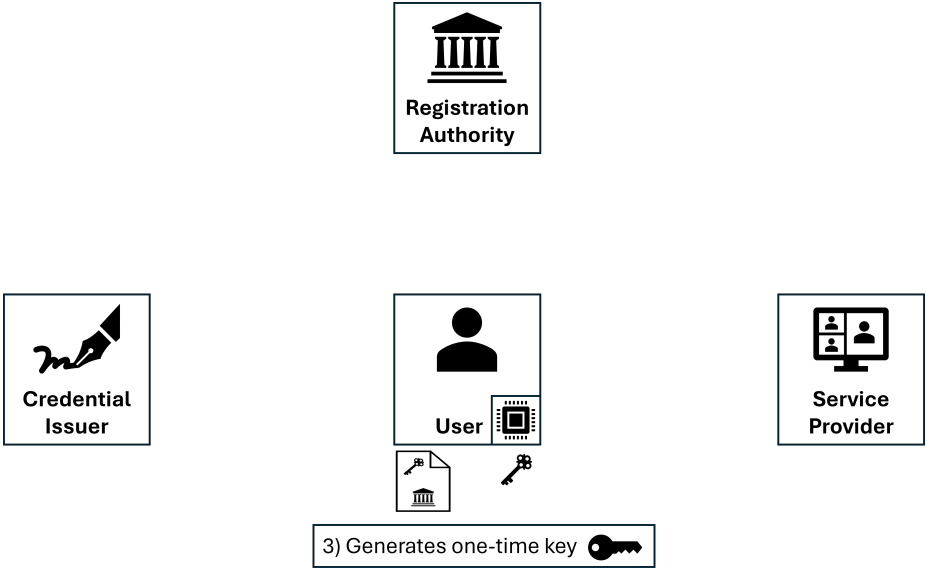
The Setting


Registration Authority
1) Registers digital identity
Credential Issuer
User
Service Provider

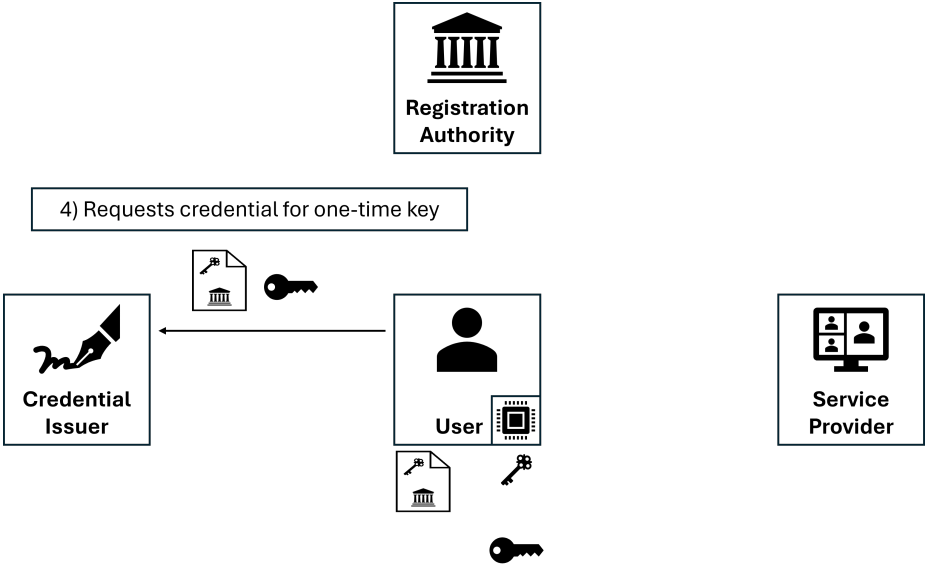
The Setting


Registration Authority
2) Provides identity certificate
Credential Issuer
User
Service Provider

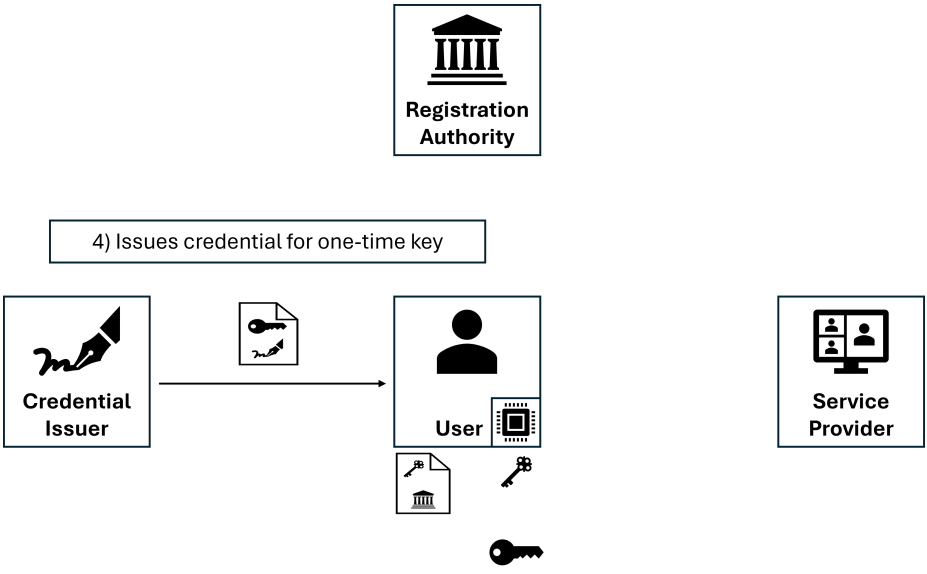
The Setting


Registration Authority
Credential Issuer
User
Service Provider
3) Generates one-time key

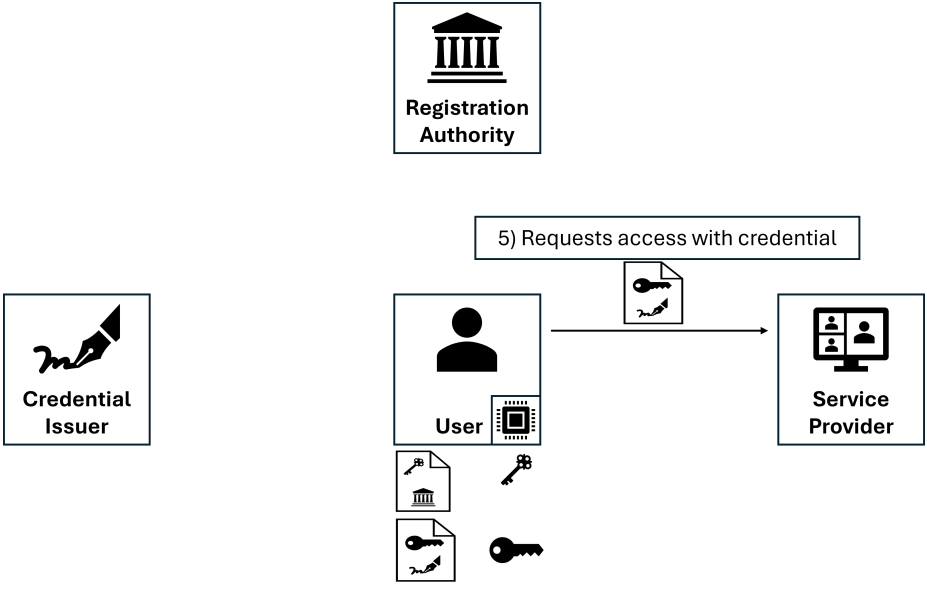
The Setting


Registration Authority
4) Requests credential for one-time key
Credential Issuer
User
Service Provider

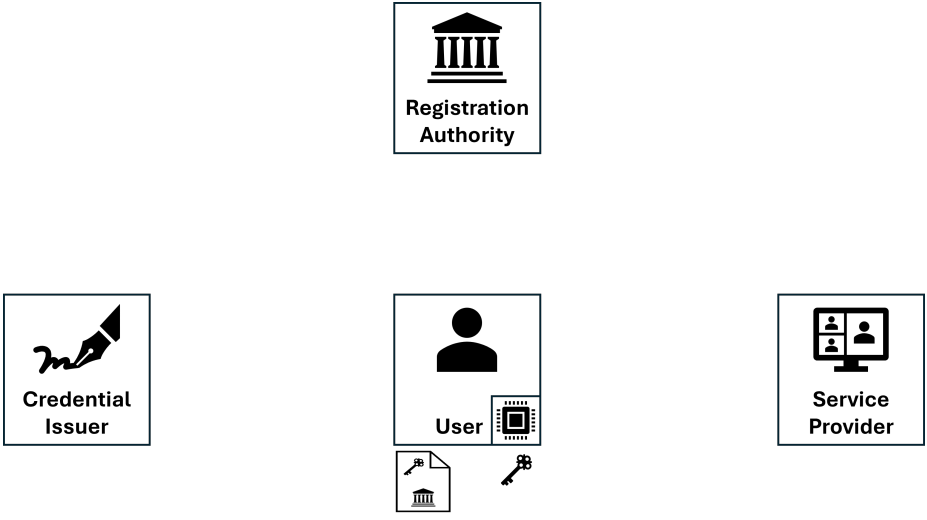
The Setting


Registration Authority
4) Issues credential for one-time key
Credential Issuer
User
Service Provider

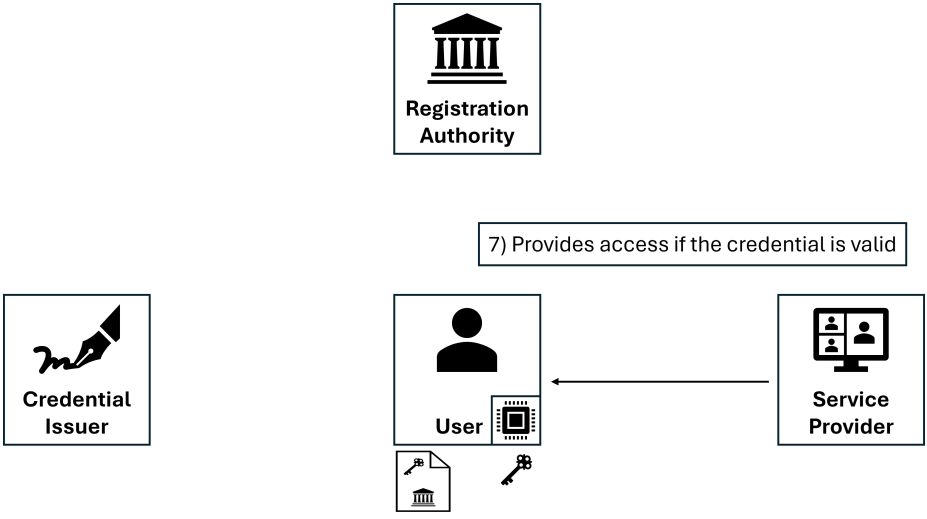
The Setting


Registration Authority
5) Requests access with credential
Credential Issuer
User
Service Provider

The Setting


Registration Authority
Credential Issuer
User
Service Provider
6) User deletes one-time key

The Setting


Registration Authority
7) Provides access if the credential is valid
Credential Issuer
User
Service Provider

Security Properties

- **Unforgeability:**
  - On registration
  - On credential issuing

Security Properties

- **Unforgeability:**
  - On registration
  - On credential issuing
- **Presentation soundness:** Only users with validly issued credentials can make a service provider accept.

Security Properties

- **Unforgeability:**
  - On registration
  - On credential issuing
- **Presentation soundness:** Only users with validly issued credentials can make a service provider accept.
- **Selective disclosure:** Only the information required by the service provider is shared.

Security Properties

- **Unforgeability:**
  - On registration
  - On credential issuing
- **Presentation soundness:** Only users with validly issued credentials can make a service provider accept.
- **Selective disclosure:** Only the information required by the service provider is shared.
- **Device binding:** Credentials should not leave the device to prevent users from sharing or selling them.

The need for Privacy

One main property: **unlinkability**.

The need for Privacy

One main property: **unlinkability**.

- Service providers shouldn't be able to *correlate different presentations*.

The need for Privacy

One main property: **unlinkability**.

- Service providers shouldn't be able to *correlate different presentations*.
- Same for issuers.

The need for Privacy

One main property: **unlinkability**.

- Service providers shouldn't be able to *correlate different presentations*.
- Same for issuers.
- *Potential scenario:* if an issuer leaks, hackers get a database of timestamps, URLs, identities… of citizens accessing adult content webpages – or worse.

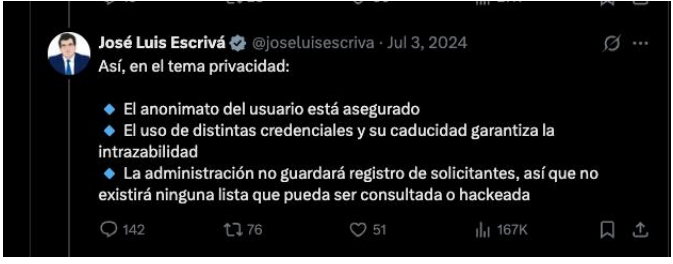
The need for Privacy

One main property: **unlinkability**.

- Service providers shouldn't be able to *correlate different presentations*.
- Same for issuers.
- *Potential scenario:* if an issuer leaks, hackers get a database of timestamps, URLs, identities… of citizens accessing adult content webpages – or worse.
- *Data minimization* is central.

The PaXport

PAXPORT Security Goals



PAXPORT Credential Issuing

Credentials are only for one attribute: $K =$ *"I am over 18"*.

PAXPORT Credential Issuing

Credentials are only for one attribute: $K =$ *"I am over 18"*.

- User requests N credentials with a short expiration date $\mathrm{exp}^{\mathrm{age}}$ (e.g. 30 days).

PAXPORT Credential Issuing

Credentials are only for one attribute: $K =$ *"I am over 18"*.

- User requests N credentials with a short expiration date $\mathsf{exp}^{\mathsf{age}}$ (e.g. 30 days).
- For this, users **generate** N key pairs $\{(\mathsf{sk}_i, \mathsf{pk}_i)\}_{i \in [N]}$.

PAXPORT Credential Issuing

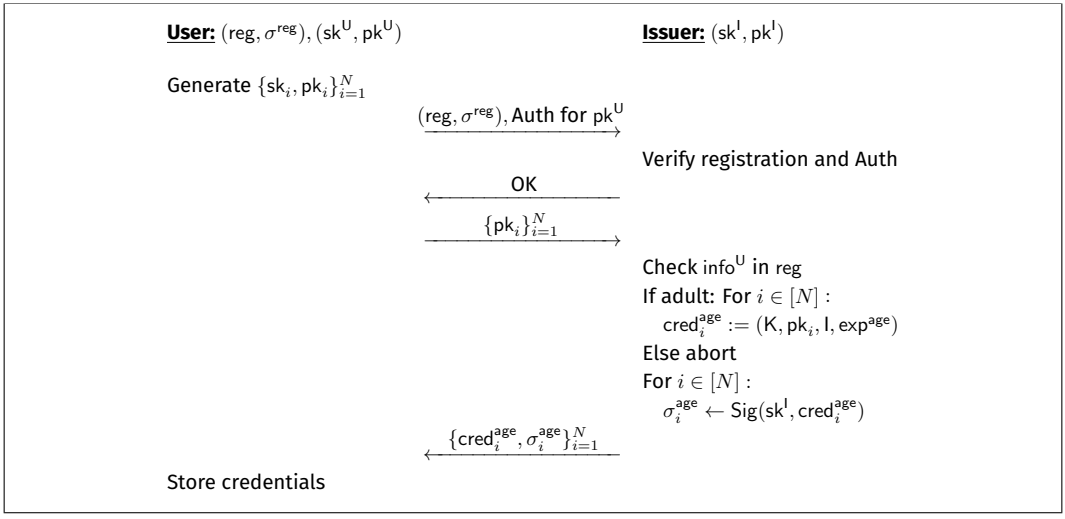
Credentials are only for one attribute: $K =$ *"I am over 18"*.

- User requests N credentials with a short expiration date $\mathsf{exp}^{\mathsf{age}}$ (e.g. 30 days).
- For this, users **generate** N key pairs $\{(\mathsf{sk}_i, \mathsf{pk}_i)\}_{i \in [N]}$.
- The issuer **signs** the credential:

$$\mathsf{cred}_i^{\mathsf{age}} = (K, \mathsf{pk}_i, I, \mathsf{exp}^{\mathsf{age}})$$

Where I is the issuer.

PAXPORT Credential Issuing



PAXPORT Credential Presentation

User wants to present $\mathsf{cred}_i^{\mathsf{age}}, \sigma_i^{\mathsf{age}}$ associated to pk_i.

PAXPORT Credential Presentation

User wants to present $\mathsf{cred}_i^{\mathsf{age}}, \sigma_i^{\mathsf{age}}$ associated to pk_i.

- User gets a *challenge* c from the service provider.

PAXPORT Credential Presentation

User wants to present $\mathsf{cred}_i^{\mathsf{age}}, \sigma_i^{\mathsf{age}}$ associated to pk_i.

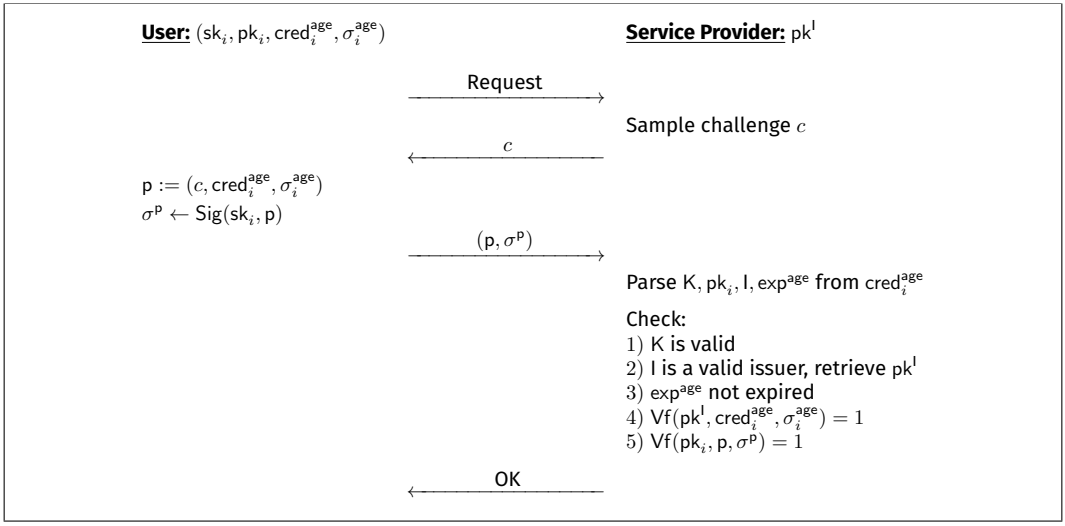
- User gets a *challenge* c from the service provider.
- User **signs** c using sk_i and sends the signed challenge and the issuer-signed credential.

PAXPORT Credential Presentation

User wants to present $\mathsf{cred}_i^{\mathsf{age}}, \sigma_i^{\mathsf{age}}$ associated to pk_i.

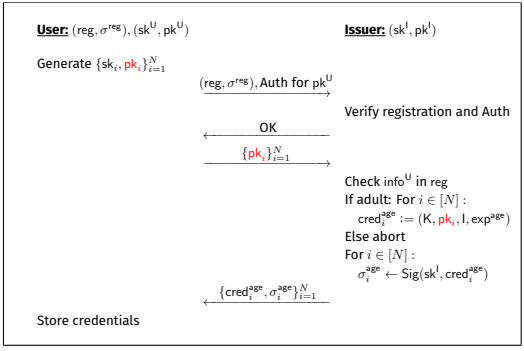
- User gets a *challenge* c from the service provider.
- User **signs** c using sk_i and sends the signed challenge and the issuer-signed credential.
- Service provider **checks**:
  - the *signature* on c,
  - the credential *validity* (using the issuer's public key),
  - the *expiration* date

PAXPORT Credential Presentation

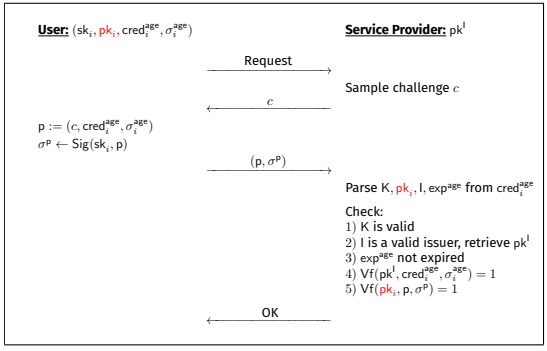


PAXPORT in a Nutshell

Issuance:



Presentation:



Properties of PAXPORT

- **Unforgeability:**
  - Registration $\rightarrow \checkmark$
  - Credential issuing $\rightarrow \checkmark$

Properties of PaXport

- **Unforgeability:**
  - Registration $\rightarrow$ ✓
  - Credential issuing $\rightarrow$ ✓
- **Presentation soundness:** ✓

Properties of PAXPORT

- **Unforgeability:**
  - Registration $\rightarrow$ ✓
  - Credential issuing $\rightarrow$ ✓
- **Presentation soundness:** ✓
- **Selective disclosure:** ✓

Properties of PAXPORT

- **Unforgeability:**
  - Registration $\rightarrow$ ✓
  - Credential issuing $\rightarrow$ ✓
- **Presentation soundness:** ✓
- **Selective disclosure:** ✓
- **Device binding: ?**

Properties of PAXPORT

- **Unforgeability:**
  - Registration $\rightarrow$ ✓
  - Credential issuing $\rightarrow$ ✓
- **Presentation soundness:** ✓
- **Selective disclosure:** ✓
- **Device binding: ?**
- **Unlinkability:** $\times$
  Only holds under the assumption that the issuer is uncorruptible.

The Unlinkability Issue

Credential presentations can be tracked:

1. During issuance, issuer receives $\mathsf{info}^{\mathsf{U}}$ and $\{\mathsf{pk}_i\}_{i=1}^{N}$ from user U.

The Unlinkability Issue

Credential presentations can be tracked:

1. During issuance, issuer receives $\mathsf{info}^{\mathsf{U}}$ and $\{\mathsf{pk}_i\}_{i=1}^{N}$ from user U.
2. Service provider receives $(\mathsf{p}, \sigma^{\mathsf{p}})$, where p includes pk_i.

The Unlinkability Issue

Credential presentations can be tracked:

1. During issuance, issuer receives $\mathsf{info}^{\mathsf{U}}$ and $\{\mathsf{pk}_i\}_{i=1}^{N}$ from user U.
2. Service provider receives $(\mathsf{p}, \sigma^{\mathsf{p}})$, where p includes pk_i.
3. pk_i can be sent to the issuer, who checks to which $\mathsf{info}^{\mathsf{U}}$ it corresponds.

The Unlinkability Issue

Credential presentations can be tracked:

1. During issuance, issuer receives $\mathsf{info}^{\mathsf{U}}$ and $\{\mathsf{pk}_i\}_{i=1}^{N}$ from user U.
2. Service provider receives $(\mathsf{p}, \sigma^{\mathsf{p}})$, where p includes pk_i.
3. pk_i can be sent to the issuer, who checks to which $\mathsf{info}^{\mathsf{U}}$ it corresponds.

A *security breach* in the issuer *exposes* this information to attackers.

The problem is that this data *exists at all* beyond the user's device.

Other Limitations

- **Storage of cryptographic keys.** Secure enclaves have bounded storage.

Other Limitations

- **Storage of cryptographic keys.** Secure enclaves have bounded storage.
- **Denial of credentials.** The design relies on a central party as both registration authority and issuer.

Other Limitations

- **Storage of cryptographic keys.** Secure enclaves have bounded storage.
- **Denial of credentials.** The design relies on a central party as both registration authority and issuer.
- **Server overhead.** Short expiration time requires lots of interaction.

Other Limitations

- **Storage of cryptographic keys.** Secure enclaves have bounded storage.
- **Denial of credentials.** The design relies on a central party as both registration authority and issuer.
- **Server overhead.** Short expiration time requires lots of interaction.
- **Issuer disclosure.** Hiding the issuer's identity is not possible.

Other Limitations

- **Storage of cryptographic keys.** Secure enclaves have bounded storage.
- **Denial of credentials.** The design relies on a central party as both registration authority and issuer.
- **Server overhead.** Short expiration time requires lots of interaction.
- **Issuer disclosure.** Hiding the issuer's identity is not possible.
- **Attestation of complex statements.** For credentials with multiple attributes, proving complex statements is impossible.

Other Limitations

- **Storage of cryptographic keys.** Secure enclaves have bounded storage.
- **Denial of credentials.** The design relies on a central party as both registration authority and issuer.
- **Server overhead.** Short expiration time requires lots of interaction.
- **Issuer disclosure.** Hiding the issuer's identity is not possible.
- **Attestation of complex statements.** For credentials with multiple attributes, proving complex statements is impossible.
- **Possible deanonymization due to secure enclaves' key signing.**

Solutions

How to Solve the Unlinkability Issue

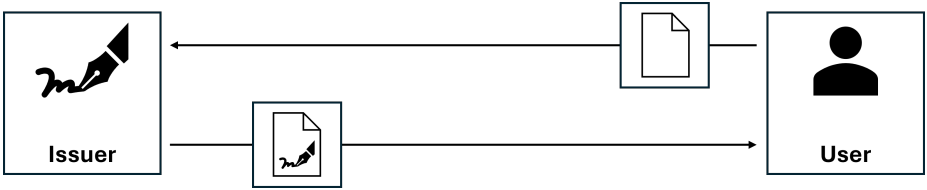
1. **A simple, quick patch:** fixing the current system with enhanced privacy.
   - We suggest a simple patch using partially blind signatures.
   - These are standardized and readily available.

How to Solve the Unlinkability Issue

1. **A simple, quick patch:** fixing the current system with enhanced privacy.
   - We suggest a simple patch using partially blind signatures.
   - These are standardized and readily available.
2. **A necessary redesign:** Work towards an overall better design of the system based on *anonymous credentials*.

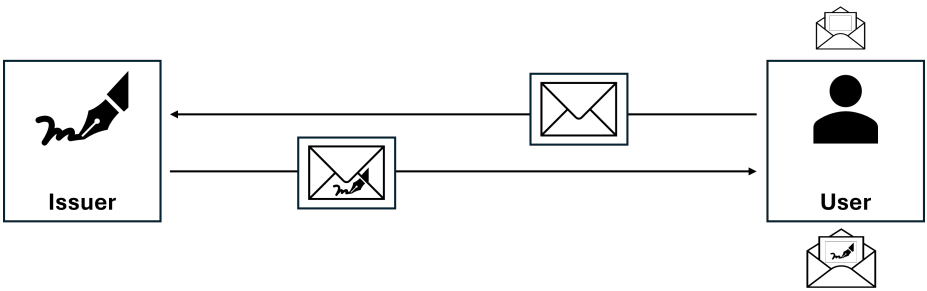
A Quick Patch: Partially Blind Signatures

Credentials are now digitally signed by the issuer.



A Quick Patch: Partially Blind Signatures

The user can hide pk_i in the signing process.

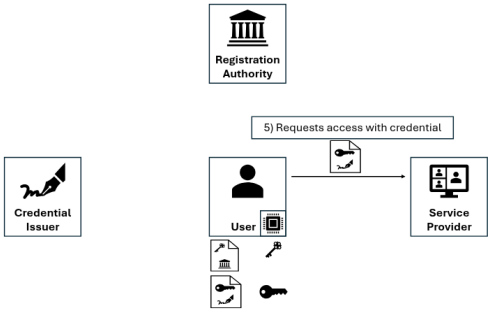


A Redesign: Anonymous Credentials

Anonymous credential systems enable users to:

- *Re-randomize* the credential signature at every presentation.
- *Prove* (in zero-knowledge) statements about the credential without disclosing the attributes fully.

There are many systems available.

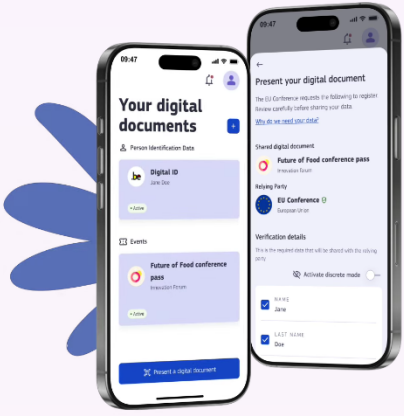


The EUDI Wallet and Next Steps

Current Designs: EUDI Wallet

A digital ID and personal digital wallet for EU citizens, residents and businesses

EU Digital Identity Wallets will provide a safe, reliable, and private means of digital identification for everyone in Europe. Every Member State will provide at least one wallet to all its citizens, residents, and businesses allowing them to prove who they are, and safely store, share and sign important digital documents.



Current Designs: EUDI Wallet

The European Digital Identity (EUDI) Wallet describes a system with good privacy properties that must be implemented by all EU countries.[1]

> The EUDI Wallet **shall** enable the user to share only the information they intend to share. The Wallet **shall** ensure an appropriate level of privacy, implementing policies about non-traceability and unlinkability of user's activities for third parties as appropriate considering:

[1] https://digital-strategy.ec.europa.eu/en/library/european-digital-identity-architecture-and-reference-framework-outline

[2] Anja Lehmann: EU Digital Identity and Anonymous Credentials - A Happy End?. RWC 2025.

Current Designs: EUDI Wallet

The European Digital Identity (EUDI) Wallet describes a system with good privacy properties that must be implemented by all EU countries.[1]

> The EUDI Wallet **shall** enable the user to share only the information they intend to share. The Wallet **shall** ensure an appropriate level of privacy, implementing policies about non-traceability and unlinkability of user's activities for third parties as appropriate considering:

Implementation: batch issuance of one-time credentials that support *selective disclosure*. This is insufficient.[2]

Properties	Classic Signatures	„Patched" Signatures
Unobservability	✓	✓
Selective Disclosure	✗	Salted hashes ✓
RP ↔ RP Unlinkability	✗	Batch issuance ✓
IdP ↔ RP Unlinkability	✗	Impossible ✗

[1] https://digital-strategy.ec.europa.eu/en/library/european-digital-identity-architecture-and-reference-framework-outline

[2] Anja Lehmann: EU Digital Identity and Anonymous Credentials - A Happy End?. RWC 2025.

EUDI Wallet and ZKPs

The good news: EUDI Wallet is actually considering ZKP implementation in future versions.[3]

Specification for ZKP Implementation in EUDI Wallet

Abstract

The present document specifies the technical specification and requirements for Zero-Knowledge Proof (ZKP) Implementation in EUDI Wallet.

GitHub discussion

Versioning

Version	Date	Description
0.1	03.04.2025	Initial version for discussion
0.2	23.04.2025	Updates based on first group meeting
0.3	02.05.2025	Updates based on second group meeting
1.0	21.05.2025	Final version

[3] https://github.com/eu-digital-identity-wallet/eudi-doc-standards-and-technical-specifications/blob/main/docs/technical-specifications/ts4-zkp.md

Why ZKP-based systems?

“Academic” AC systems (e.g. BBS/PS signatures) are not easy to deploy at a large scale in practice:

Why ZKP-based systems?

"Academic" AC systems (e.g. BBS/PS signatures) are not easy to deploy at a large scale in practice:

- Desired use of standardized cryptography (e.g. ECDSA, RSA).

Why ZKP-based systems?

"Academic" AC systems (e.g. BBS/PS signatures) are not easy to deploy at a large scale in practice:

- Desired use of standardized cryptography (e.g. ECDSA, RSA).
- Hardware modules for cryptographic operations.

Why ZKP-based systems?

"Academic" AC systems (e.g. BBS/PS signatures) are not easy to deploy at a large scale in practice:

- Desired use of standardized cryptography (e.g. ECDSA, RSA).
- Hardware modules for cryptographic operations.
- Post-quantum readiness.

Why ZKP-based systems?

"Academic" AC systems (e.g. BBS/PS signatures) are not easy to deploy at a large scale in practice:

- Desired use of standardized cryptography (e.g. ECDSA, RSA).
- Hardware modules for cryptographic operations.
- Post-quantum readiness.
- Compatibility with existing mechanisms: ID cards, passports, drivers license...

Why ZKP-based systems?

"Academic" AC systems (e.g. BBS/PS signatures) are not easy to deploy at a large scale in practice:

- Desired use of standardized cryptography (e.g. ECDSA, RSA).
- Hardware modules for cryptographic operations.
- Post-quantum readiness.
- Compatibility with existing mechanisms: ID cards, passports, drivers license...
- Multiple issuer support.

Why ZKP-based systems?

"Academic" AC systems (e.g. BBS/PS signatures) are not easy to deploy at a large scale in practice:

- Desired use of standardized cryptography (e.g. ECDSA, RSA).
- Hardware modules for cryptographic operations.
- Post-quantum readiness.
- Compatibility with existing mechanisms: ID cards, passports, drivers license...
- Multiple issuer support.
- Handling revocation, updates, storage of cryptographic material...

Why ZKP-based systems?

"Academic" AC systems (e.g. BBS/PS signatures) are not easy to deploy at a large scale in practice:

- Desired use of standardized cryptography (e.g. ECDSA, RSA).
- Hardware modules for cryptographic operations.
- Post-quantum readiness.
- Compatibility with existing mechanisms: ID cards, passports, drivers license...
- Multiple issuer support.
- Handling revocation, updates, storage of cryptographic material...

Overall, *zero-knowledge proofs* generated by the holder are a good solution.

How do they work?

ZKPs for Anonymous Credentials

Issuing:

- User holds a long-term key pair $(\mathsf{sk}, \mathsf{pk})$.

ZKPs for Anonymous Credentials

Issuing:

- User holds a long-term key pair $(\mathsf{sk}, \mathsf{pk})$.
- Issuer signs $\mathsf{cred} = (\mathsf{pk}, a_1, \ldots, a_k)$ for user U, obtaining σ_{cred}.

ZKPs for Anonymous Credentials

Issuing:

- User holds a long-term key pair $(\mathsf{sk}, \mathsf{pk})$.
- Issuer signs $\mathsf{cred} = (\mathsf{pk}, a_1, \ldots, a_k)$ for user U, obtaining σ_{cred}.

Presentation:

ZKPs for Anonymous Credentials

Issuing:

- User holds a long-term key pair $(\mathsf{sk}, \mathsf{pk})$.
- Issuer signs $\mathsf{cred} = (\mathsf{pk}, a_1, \ldots, a_k)$ for user U, obtaining σ_{cred}.

Presentation:

- Service provider sends *challenge* c.

ZKPs for Anonymous Credentials

Issuing:

- User holds a long-term key pair $(\mathsf{sk}, \mathsf{pk})$.
- Issuer signs $\mathsf{cred} = (\mathsf{pk}, a_1, \ldots, a_k)$ for user U, obtaining σ_{cred}.

Presentation:

- Service provider sends *challenge* c.
- User *signs* $\sigma \leftarrow \mathsf{Sig}(\mathsf{sk}, c)$ on the secure enclave.

ZKPs for Anonymous Credentials

Issuing:

- User holds a long-term key pair $(\mathsf{sk}, \mathsf{pk})$.
- Issuer signs $\mathsf{cred} = (\mathsf{pk}, a_1, \ldots, a_k)$ for user U, obtaining σ_{cred}.

Presentation:

- Service provider sends *challenge* c.
- User *signs* $\sigma \leftarrow \mathsf{Sig}(\mathsf{sk}, c)$ on the secure enclave.
- User *proves* that it knows $(\mathsf{pk}, \sigma_{\mathsf{cred}}, \mathsf{cred})$ such that:

ZKPs for Anonymous Credentials

Issuing:

- User holds a long-term key pair $(\mathsf{sk}, \mathsf{pk})$.
- Issuer signs $\mathsf{cred} = (\mathsf{pk}, a_1, \ldots, a_k)$ for user U, obtaining σ_{cred}.

Presentation:

- Service provider sends *challenge* c.
- User *signs* $\sigma \leftarrow \mathsf{Sig}(\mathsf{sk}, c)$ on the secure enclave.
- User *proves* that it knows $(\mathsf{pk}, \sigma_{\mathsf{cred}}, \mathsf{cred})$ such that:
  - Challenge signature verifies: $\mathsf{Vf}(\sigma, \mathsf{pk}, c) = 1$.
  - Credential (including pk) is signed: $\mathsf{Vf}(\mathsf{cred}, \mathsf{pk}^{\mathsf{I}}, \sigma_{\mathsf{cred}}) = 1$.
  - $P(a_1, \ldots, a_k) = 1$ for some predicate P.

  User never reveals pk. Also, sk *is not used in the proof*.

Deploying Zero-Knowledge Proofs

Deploying ZK proofs in digital ID systems brings *many technical challenges*.

Deploying Zero-Knowledge Proofs

Deploying ZK proofs in digital ID systems brings *many technical challenges*.

- **Security:**
  - Careful design and specification to avoid forgeries (e.g. malleability, Fiat-Shamir attacks).
  - Proper, unambiguous parsing of attributes for all supported data formats.
  - Integration within a larger protocol, realistic security models.
  - Trusted setups.

Deploying Zero-Knowledge Proofs

Deploying ZK proofs in digital ID systems brings *many technical challenges*.

- **Security:**
  - Careful design and specification to avoid forgeries (e.g. malleability, Fiat-Shamir attacks).
  - Proper, unambiguous parsing of attributes for all supported data formats.
  - Integration within a larger protocol, realistic security models.
  - Trusted setups.
- **Efficiency:**
  - Proving and verification time trade-offs.
  - Support for proof-unfriendly cryptographic primitives (e.g. ECDSA).
  - Credential-specific optimizations.

Bonus: The Google ZKP Library

- Anonymous Credentials for ECDSA.[4]

[4] https://blog.google/technology/safety-security/opening-up-zero-knowledge-proof-technology-to-promote-privacy-in-age-assurance/

Bonus: The Google ZKP Library

- Anonymous Credentials for ECDSA.[4]
- *Cryptographically sound*, but some things left open:
  - Current proposal cannot hide issuers.
  - Only specifies two specific data formats for US drivers licenses.
  - Extending ZK proofs to complex data formats is not easy.
  - Heavily optimized for ECDSA – extending to other algorithms is open.

[4] https://blog.google/technology/safety-security/opening-up-zero-knowledge-proof-technology-to-promote-privacy-in-age-assurance/