FUTURE OF CRYPTOGRAPHY 2026 · ESTONIAN ACADEMY OF SCIENCES

Theoretically Sound, Practically Achievable

Zero-Knowledge Proofs from Multi-Message Signatures for the EUDI Wallet

SPEAKER

Søren Eller Thomsen

Senior Cryptographic Engineer, Partisia, Co-editor of TS14

Today's talk

Part 1

Requirements

- The regulation
- The high-level requirements that operationalise it
- The LoA High constraint
- The two technical paths: TS13 and TS14

~5 min

Part 2

A modular construction

- Multi-message signatures
- Committed disclosure and sub-proofs
- Device binding
- Other sub-proofs

~15 min

Part 3

Revocation

- Three viable mechanisms and their trade-offs
- The TS14 recommendation

~5 min

PART 1 OF 3

Requirements

What is required from the ZKP functionality of the wallet.

Regulation (EU) 2024/1183 / Recital 14

> Member States should integrate different privacy-preserving technologies, such as zero knowledge proof, into the European Digital Identity Wallet. Those cryptographic methods should allow a relying party to validate whether a given statement based on the person's identification data and attestation of attributes is true, without revealing any data on which that statement is based, thereby preserving the privacy of the user.

Rather vague. Details left to the ARF to pin down.

Source: Regulation (EU) 2024/1183, Recital 14

High-level requirements

Mandatory (SHALL):

ZKP_01

Selective disclosure: hide every attribute that is not asked about.

(i) prove an attribute value is in an attestation

(ii) prove the attestation has not expired

(iii) prove the attestation has not been revoked

(iv) prove that a device-bound key sits in the WSCA/WSCD or wallet keystore

ZKP_02

Prove possession of an attestation of a given type (e.g. “this is a PID”).

ZKP_03

Combined presentation: link two attestations without revealing the shared attribute.

ZKP_05

Usable in remote and proximity flows; ZKP delays must not undermine the relying party’s purpose.

ZKP_07

No additional communication or information that could be used to track or link the user.

ZKP_08

Only algorithms from the ENISA ECCG Agreed Cryptographic Mechanisms v2.0.

ZKP_09

The ZKP scheme must not prevent the wallet from authenticating at **eIDAS LoA High.**

Source: ARF / Annex II / Topic 53

CIR (EU) 2015/1502 / ANNEX

LoA High requirements

§2.2.1: Characteristics and design (LoA High)

> *Level substantial, plus:*
>
> 1. *The electronic identification means protects against duplication and tampering as well as against attackers with high attack potential*

⟹ Wallets' keys must be stored in a certified **Wallet Secure Cryptographic Device**.

§2.4.6: Technical controls (LoA High)

> *Sensitive cryptographic material, if used for issuing electronic identification means and authentication is protected from tampering.*

⟹ The issuer's keys must be protected in certified **hardware**.

Source: Commission Implementing Regulation (EU) 2015/1502 / Annex

EUDI TECHNICAL-SPECIFICATIONS REPOSITORY

Two technical specifications

TS13: ZK-SNARKs over arithmetic circuits

Encode the existing credential (mdoc or SD-JWT VC) as a circuit witness. Prove possession + selective disclosure by running a SNARK over that circuit.

- **Compatible with mdoc and SD-JWT VC:** no issuance change, no new credential format.
- Reference implementation by Google: Longfellow.
- Each circuit is specialised per signature scheme, parsing, hash function, and predicate ⟹ large audit surface and new standardization path.

TS14: ZKPs from multi-message signatures

Replace the credential format. The issuer signs a vector of indexed messages with a signature scheme designed for re-randomisation. The proof exploits the scheme's algebraic structure.

- **New credential format:** mdoc and SD-JWT VC do not fit .
- Reference: BBS over BLS12-381, IRTF `draft-irtf-cfrg-bbs-signatures`.
- Predicate proofs compose over a fixed commitment scheme - small and composable audit surface.

Both specifications target the same HLRs. Part 2 of this talk will explain the TS14 path.

PART 2 OF 3

The construction

ZKPs form multi-message signatures for the EUDIW.

DEFINITION

Multi-message signatures

Issuance

An issuer with key pair $(\mathsf{isk}, \mathsf{ipk})$ signs an indexed vector of n attributes: $\sigma = \mathsf{Sign}(\mathsf{isk}, (m_1, \ldots, m_n))$.

The credential is the pair $(\sigma, (m_1, \ldots, m_n))$; one signature attesting to all n messages together.

Presentation

The holder picks a disclosure set D of indices and runs the presentation algorithm. The output Π reveals the attributes m_i for $i \in D$ and **keeps the rest hidden from the verifier**.

Π is a zero-knowledge proof that exploits σ's **algebraic structure** i.e, based on re-randomization, not a SNARK circuit over byte-level hashing and parsing.

Example

A relying party asks for a credential. The holder discloses `vct`; every other attribute stays hidden:

M_1	M_2	M_3	M_4	M_5
vct	*hidden*	*hidden*	*hidden*	*hidden*

THREE STATES PER MESSAGE

A Modular Architecture

At presentation, each signed attribute is in one of three states. The holder picks the partition per relying-party request.

Disclosed

The value is sent to the verifier in cleartext alongside the proof.

Committed

A fresh commitment stands in for the value. Sub-proofs can operate on it without revealing what is inside.

Hidden

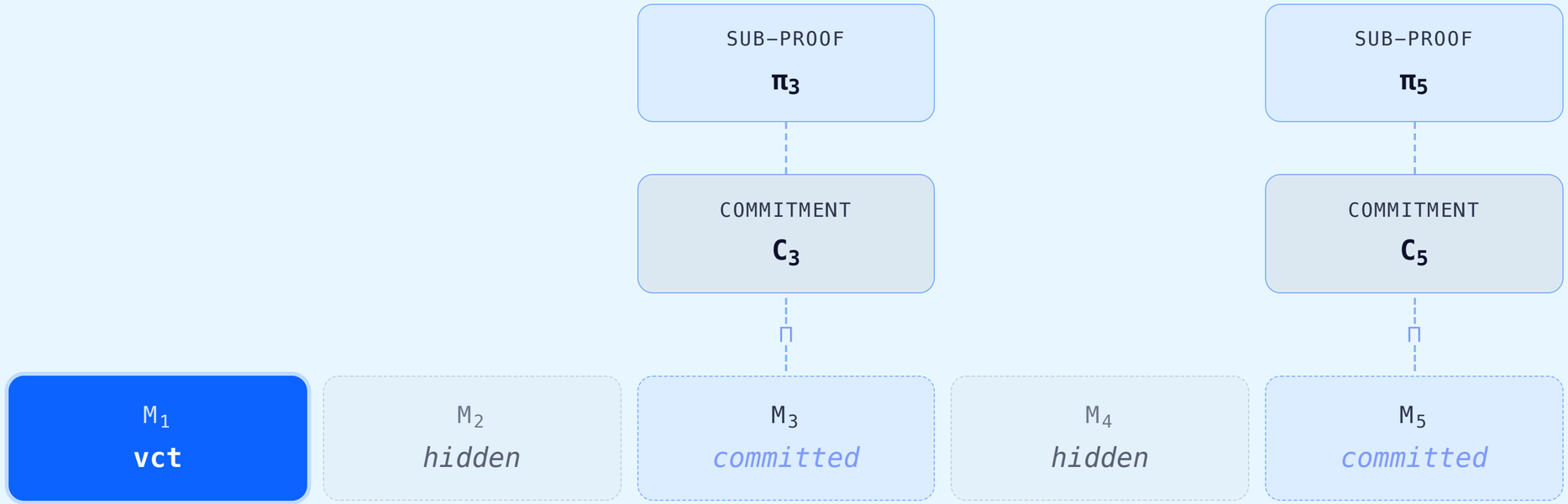
Not sent at all. The verifier learns nothing about this attribute.

Article: Lehmann, Sidorenko, Zacharakis. Vision: A Modular Framework for Anonymous Credential Systems [LSZ25]

PREDICATE PROOFS HANG OFF EACH COMMITMENT

Sub-proofs over commitments

Each commitment C_i carries an additional sub-proof π_i: a zero-knowledge proof that the opening of C_i satisfies some predicate P_i.



THE MMS SCHEME USED IN THE REST OF THIS PART

BBS

Signature

$$\sigma = (A, e)$$

One group element and one scalar. Size is constant, independent of the number of signed messages.

Verification

Two pairings. That is the entire verifier computation.

Presentation

The holder re-randomises σ into a fresh σ' and proves knowledge of it.

Introduced by Boneh, Boyen, and Shacham (2004). Standardised as `draft-irtf-cfrg-bbs-signatures`. TS14 picks BBS for its short signature, mature implementation ecosystem, and compatibility with the committed-disclosure framework from the previous slides.

THE COMMITMENT SCHEME USED FOR COMMITTED DISCLOSURE

Pedersen commitments

Commitment

$$C = g^m \cdot h^r$$

Two group generators g, h with unknown discrete-log relation. The committer draws fresh randomness r and publishes C. Open by revealing (m, r).

Hiding

Perfect. C is uniform over the group regardless of m; no adversary not knowing the opening can learn anything about the committed value.

Binding

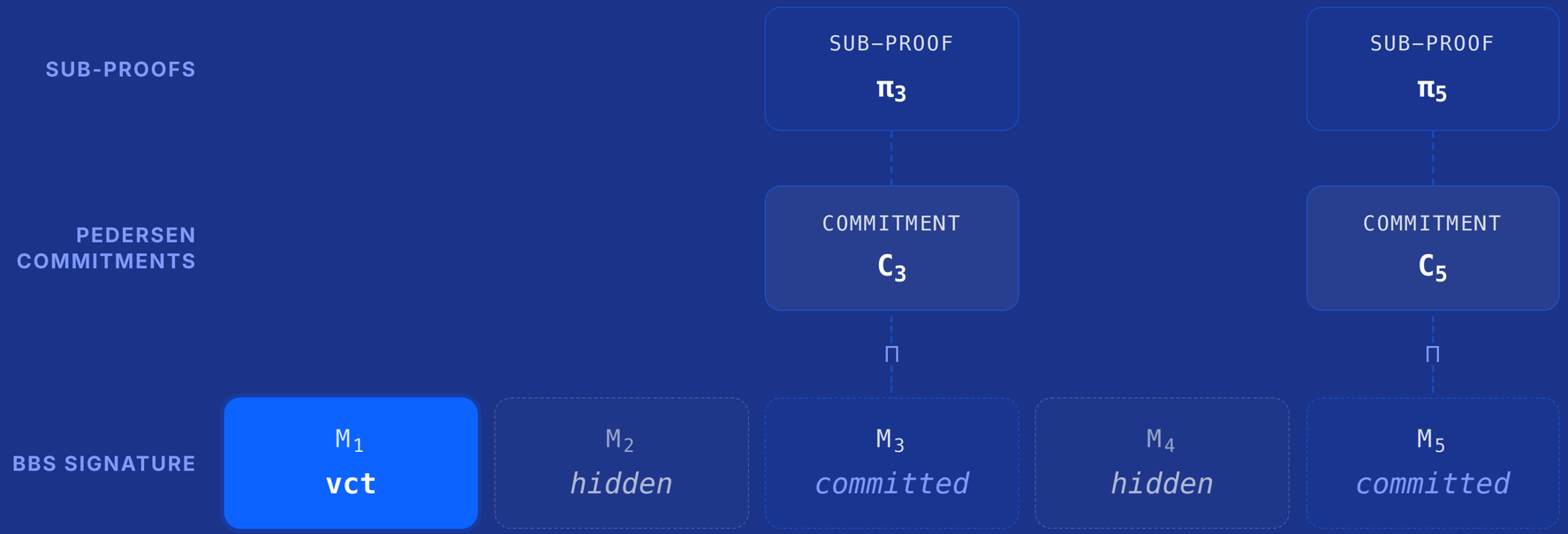
Computational. Opening to a different (m', r') reduces to finding the discrete log of h in base g.

Pedersen, T.P. Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing, CRYPTO 1991

WHERE BBS AND PEDERSEN EACH LIVE IN THE CONSTRUCTION

The modular construction

BBS does the issuer signature on the indexed attribute vector; Pedersen commitments decouple the signature scheme from the sub proofs.



FOUR PREDICATE PROOFS THAT COMPOSE OVER PEDERSEN COMMITMENTS

Sub-proof types

Range proofs

Prove $\mathrm{age} \geq 18$

Bulletproofs over the same Pedersen commitments from the previous slide. Transparent setup, ~ 1 KByte proof, milliseconds to prove and verify.

Enables ZKP_01 (ii, iii)
(expiry and revocation checks)

Equality

Link two attestations

Prove two committed attributes match without revealing them. **Schnorr** equality on Pedersen commitments.

Enables ZKP_03
(attestation bound to PID)

Set membership

Prove $m_i \in S$

“Citizen of an EU member state” without saying which. **OR-of-equality** via the CDS94 compiler over Schnorr proofs.

Non-mandatory in TS14
but the compiler is free

Device binding

Prove control a WSCD key

The device public key is signed as one of the issuer’s messages.

Required for ZKP_01 (iv)
(LoA High constraint)

NO CERTIFIED HARDWARE SPEAK BBS TODAY

The hardware gap

Device hardware (WSCD)

Phones speak ECDSA and RSA.

No native BBS.

Issuer hardware (HSM)

HSMs sign ECDSA and RSA.

No native BBS.

Why not just use ECDSA or RSA?

Both are linkable: the same signature is presented every time. That violates **ZKP_07** (no tracking across presentations).

Chicken and egg

No hardware vendor adds BBS without a standard demanding it. No standard can demand BBS until vendors ship it.

TS14 SHIPS IN TWO PHASES

Bridging the gap

TS14 defines two phases so the spec ships on today’s hardware and migrates to native BBS as vendors catch up.

Phase 1

Ship now, on today’s hardware

No certified hardware speaks BBS yet. Phase 1 works around this using additional zero-knowledge proofs so the wallet and issuer can keep using ECDSA while still meeting the HLRs.

Phase 2

When hardware catches up

Once HSMs and WSCDs support BBS natively, the ECDSA scaffolding drops. The construction simplifies and the additional ZKPs are no longer needed.

BIND THE WALLET'S CERTIFIED KEY INTO THE PROOF

Device binding: Phase 1

The requirement (ZKP_01 iv)

The wallet authenticates with a key inside its WSCD. The presentation must prove the holder controls that key without revealing it, and without letting two presentations be linked through it.

1: At issuance

An ECDSA device public key is one of the messages signed by the issuer’s MMS.

2: At presentation

The WSCD signs the verifier’s challenge with its ECDSA key. This is the same operation phones already do today.

3: In the proof

A Schnorr-style sub-proof proofs knowledge of a valid ECDSA signature on the challenge that verifies under the committed device key.

Inconvenience: The bridge between ECDSA P-256 (held in the WSCD) and the MMS curve uses an auxiliary curve T-256 which must be standardized.

ECDSA-ON-TOP WRAPS THE BBS PAYLOAD

Issuance: Phase 1

Outer layer: ECDSA(σ)

Issuer's HSM-backed P-256 signature over the BBS payload

Inner layer

BBS signature σ

over $(m_1, \ldots, m_n)$

Holder proves

Knows valid σ + valid ECDSA(σ)

in zero knowledge

Security reduction

Forging the combined object

$\Longrightarrow$ forging ECDSA

The issuer's pairing-friendly key never needs to touch an HSM in Phase 1. The cost: proving knowledge of the ECDSA signature requires a **circuit-based ZKP**, adding complexity that Phase 2 eliminates.

Article: Ringers, S. Augmenting BBS with Conventional Signatures

WHAT CHANGES WHEN HARDWARE CATCHES UP

Phase 1 vs Phase 2

	Phase 1 ships now	Phase 2 when HSMs and WSCDs speaks BBS natively
DEVICE BINDING	Schnorr ZK of ECDSA P-256 auxiliary curve bridge	DAA-style native binding single WSCD call, no bridge
ISSUANCE	BBS in software + ECDSA from the HSM unforgeability reduces to ECDSA	Native BBS from the HSM ECDSA scaffolding drops

SIX MANDATORY HLRS, BOTH PHASES

Requirements satisfied

	Phase 1	Phase 2
ZKP_01 (i–iv)	MMS + Pedersen commitments + Schnorr sub-proofs for device binding, circuit proof of the hybrid issuer signature.	Same, with native BBS replacing the ECDSA hybrid.
ZKP_02	Credential type carried as a signed message (e.g. `vct`). Can be disclosed or proved via equality sub-proof.	Same.
ZKP_07	Re-randomised σ, fresh commitments per presentation, no stable identifier. **Revocation is the remaining challenge. See Part 3.**	Same.
ZKP_08	Requires inclusion of BBS and sub proofs in ENISA's list of agreed crypto primitives.	Same.
ZKP_09	Issuer signs via HSM-bound ECDSA; wallet key sits in WSCD, circuits used to prove knowledge of signature. LoA High preserved.	Issuer signs MMS natively in an HSM that supports it; LoA High preserved.

PART 3 OF 3

Revocation

The hardest privacy problem in the wallet.

WHAT ZKP_07 FORCES US TO REDESIGN

Revocation

The constraint

The Token Status List is what mdoc and SD-JWT VC use today. It works by handing each attestation a **stable, verifier-disclosed index**. That index is, by definition, a tracking handle. It breaks ZKP_07.

Trade-off 1

Unlinkability

Does the revocation check leak a stable identifier or other metadata to the verifier or the issuer?

Trade-off 2

Wallet cost

How much extra data does the wallet fetch, store, and prove against on each presentation?

Trade-off 3

Latency

How long after revocation can a revoked credential still be used?

No mechanism wins on all three axes at once.

WHERE EACH LANDS ON THE TRADE-OFF CURVE

Four mechanisms

Mechanism	Unlinkable	Wallet cost	Latency
Token Status List (today, for mdoc / SD-JWT VC)	No	Trivial: fetch and check an index.	Low.
Short-lived attestations re-issue every N hours	**Yes***	Refresh load on the issuer; wallet fetches a new credential per epoch.	Bounded by the epoch length.
Signed pairs of indices prove “my id ∈ a non-revoked gap”	**Yes**	One extra range proof per presentation.	As fast as the issuer publishes pairs.
Dynamic accumulators constant-size membership proof via bilinear accumulators	**Yes**	Wallet pulls witness updates per epoch.	Epoch length, not standardised yet.

The trade-offs between unlinkability, wallet cost, and latency are similar both for TS13 and TS14.

*Presentations are unlinkable, but the wallet’s reissuance traffic reveals communication patterns to the issuer.

R9, BY PHASE

The TS14 recommendation

Phase 1

Signed pairs of indices + short-lived attestations

Issuer signs pairs around revoked indices. Wallet receives a short-lived non-revocation attestation each epoch and proves it has one. Reuses the same Pedersen commitments and Schnorr proofs from Part 2.

Phase 2: research direction

Dynamic accumulators

Constant-size proofs, no per-epoch reissuance. The open questions are how to protect the accumulator trapdoor in an HSM, how to distribute witness updates at scale, and which construction to standardise on.

The Token Status List remains useful as a transitional fallback for credentials where the holder is already identifiable to the verifier (cases where ZKP_07 does not apply).

Theoretically sound.
Practically achievable.

Phase 1 ships now on the HSMs we have. Phase 2 ships when HSMs and WSCDs speaks BBS natively. **Revocation** remains the a hard problem to do privately.

Questions

Thank you.

Speaker

Søren Eller Thomsen

Senior Cryptographic Engineer, Partisia

soren.eller.thomsen@partisia.com

Resources

TS14 spec: eu-digital-identity-wallet/eudi-doc-standards-and-technical-specifications