Keyed-Verification Anonymous Credentials with Highly Efficient Partial Disclosure

Omid Mirzamohammadi[1] , Jan Bobolz[2], Mahdi Sedaghat[1] , Emad Heydari Beni[1,3], Aysajan Abidin[1], Dave Singelée[4], Bart Preneel[1]

[1] COSIC, KU Leuven, Leuven, Belgium
[2] University of Edinburgh, Edinburgh, UK
[3] Nokia Bell Labs
[4] DistriNet, KU Leuven, Leuven, Belgium

PrivCrypt, May 10, 2026

Outline

- Motivation
- KVAC Introduction
- Problem Statement
- Our First KVAC Construction
- Our Second KVAC Construction

- **Motivation**
- KVAC Introduction
- Problem Statement
- Our First KVAC Construction
- Our Second KVAC Construction

Why Anonymous Credentials (AC) with Partial Disclosure?

User's attributes

- Date of birth 1993/02/17
- Nationality BEL
- University KUL

User

Service Provider

What the service provider needs to know

Is the university KUL?

Why Anonymous Credentials (AC) with Partial Disclosure?

User's attributes

- Date of birth 1993/02/17
- Nationality BEL
- University KUL

User

Privacy-preserving Authentication

Not revealing unnecessary data

Service Provider

What the service provider needs to know

Is the university KUL?

Access

Why Anonymous Credentials (AC) with Partial Disclosure?

Issuer

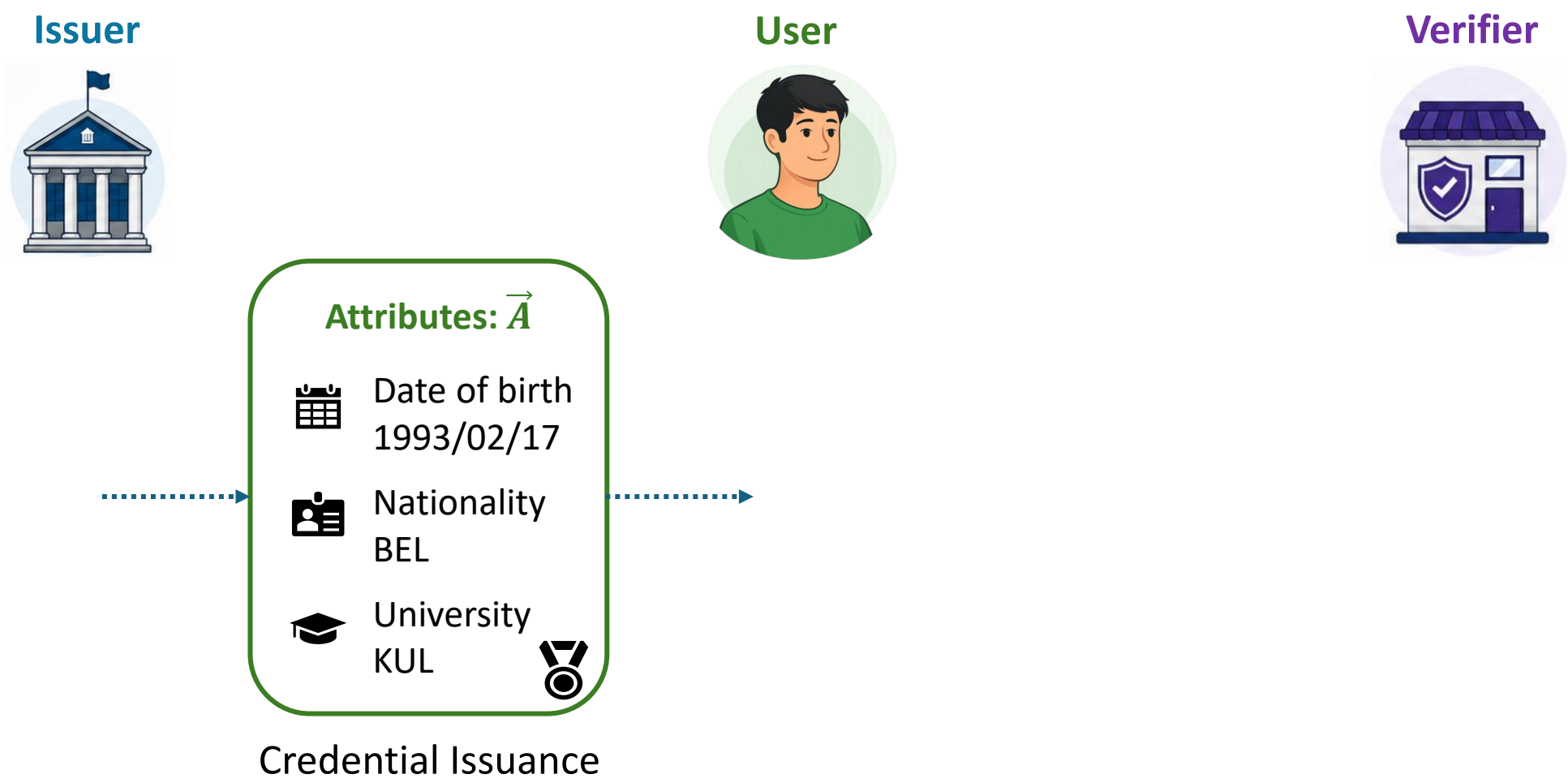
User

Attributes: $\vec{A}$

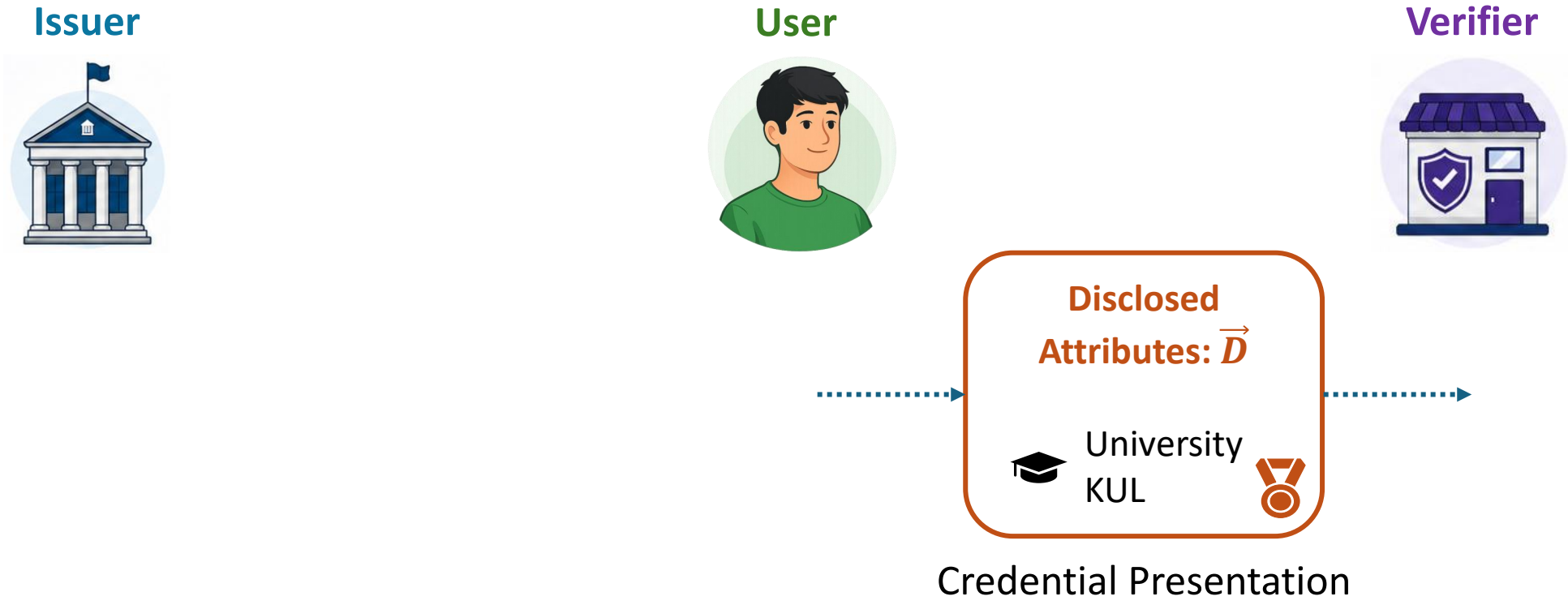
- Date of birth
  1993/02/17
- Nationality
  BEL
- University
  KUL

Verifier

Why Anonymous Credentials (AC) with Partial Disclosure?



Why Anonymous Credentials (AC) with Partial Disclosure?



✓ **Unforgeability and Multi-show Unlinkability:** The verifier learns that the user owns a valid credential for certain specific attributes and nothing beyond that.

- Motivation
- **KVAC Introduction**
- Problem Statement
- Our First KVAC Construction
- Our Second KVAC Construction

Publicly Verifiable Anonymous Credentials (PVAC)

Issuer

Issuer Secret Key (isk)

User

Verifier

Issuer Public Key (ipk)

Publicly Verifiable Anonymous Credentials (PVAC)

Issuer

Issuer Secret Key (isk)

User

Verifier

Issuer Public Key (ipk)

Signature using isk

$\vec{A}$: Set of attributes

$$\text{Cred} \leftarrow \text{Issue}(\vec{A}, \text{isk}, \text{ipk})$$

Publicly Verifiable Anonymous Credentials (PVAC)

Issuer

Issuer Secret Key (isk)

Signature using isk

User

$\vec{A}$: Set of attributes

Verifier

Issuer Public Key (ipk)

$\text{Cred} \leftarrow \text{Issue}(\vec{A}, \text{isk}, \text{ipk})$

$\vec{D}$: Set of disclosed attributes

$\text{Show} \leftarrow \text{ShowCred}(\text{cred}, \vec{A}, \vec{D}, \text{ipk})$

$0/1 \leftarrow \text{VerifCred}(\text{Show}, \vec{D}, \text{ipk})$

Publicly Verifiable Anonymous Credentials (PVAC)

Issuer | **User** | **Verifier**

Issuer Secret Key (isk)

Issuer Public Key (ipk)

Signature using isk

$\vec{A}$: Set of attributes

$$\text{Cred} \leftarrow \text{Issue}(\vec{A}, \text{isk}, \text{ipk})$$

$\vec{D}$: Set of disclosed attributes

$$\text{Show} \leftarrow \text{ShowCred}(\text{cred}, \vec{A}, \vec{D}, \text{ipk})$$

$$0/1 \leftarrow \text{VerifCred}(\text{Show}, \vec{D}, \text{ipk})$$

Most efficient schemes require pairing

Hinders Deployability

Keyed-Verification Anonymous Credentials (KVAC)

What if the verifier has isk?

Keyed-Verification Anonymous Credentials (KVAC)

What if the verifier has isk?

We can avoid pairing.

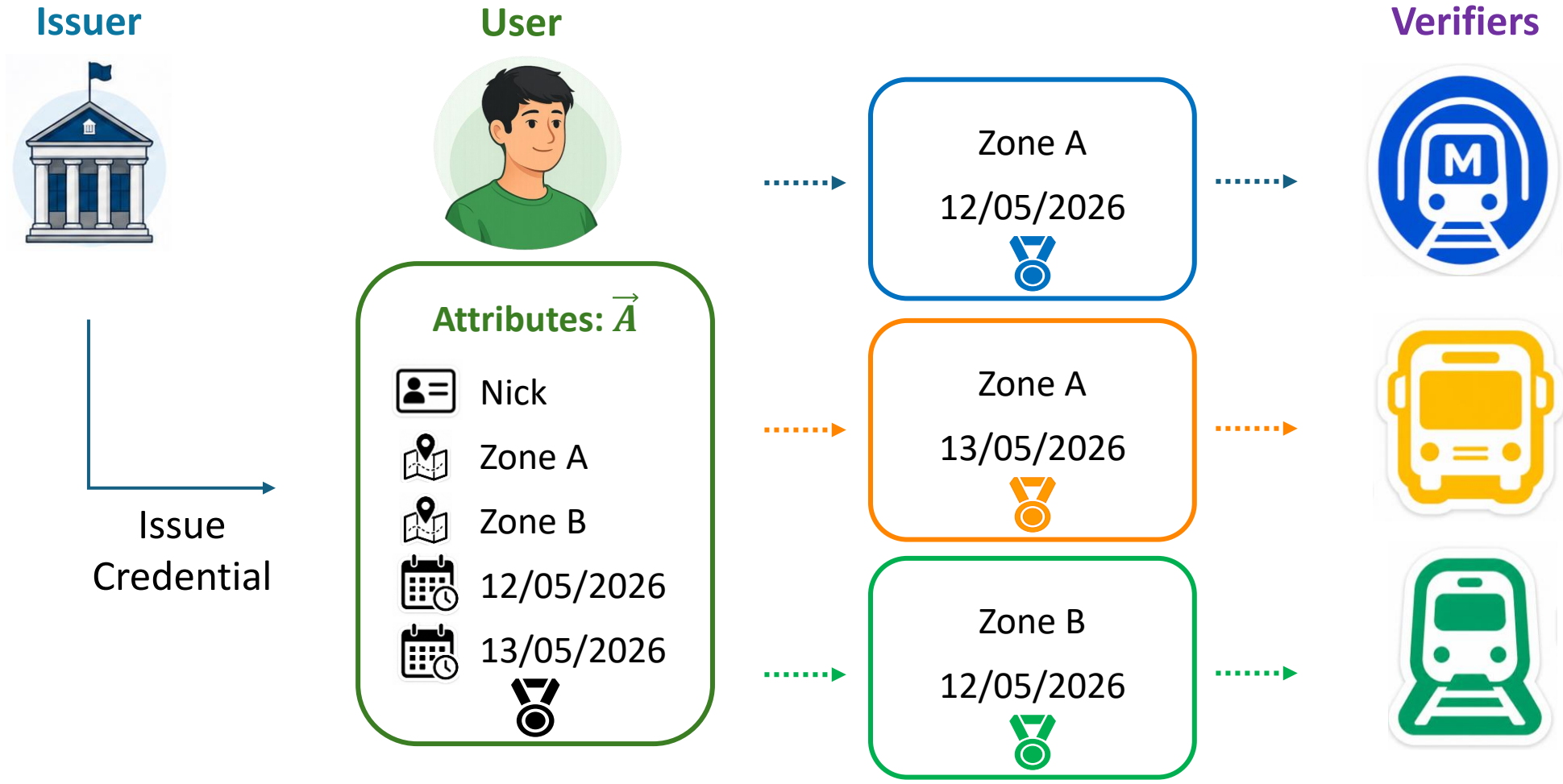
Keyed-Verification Anonymous Credentials (KVAC)

What if the verifier has isk?

We can avoid pairing.

When is it possible?

Keyed-Verification Anonymous Credentials (KVAC)



Keyed-Verification Anonymous Credentials (KVAC)

Issuer

Issuer Secret Key (isk)

MAC using isk

User

$\vec{A}$: Set of attributes

$\text{Cred} \leftarrow \text{Issue}(\vec{A}, \text{isk})$

$\vec{D}$: Set of disclosed attributes

$\text{Show} \leftarrow \text{ShowCred}(\text{cred}, \vec{A}, \vec{D})$

$0/1 \leftarrow \text{VerifCred}(\text{Show}, \vec{D}, \text{isk})$

Verifier

Issuer Secret Key (isk)

Keyed-Verification Anonymous Credentials (KVAC)

Issuer

Issuer Secret Key (isk)

MAC using isk

User

$\vec{A}$: Set of attributes

$\text{Cred} \leftarrow \text{Issue}(\vec{A}, \text{isk})$

$\vec{D}$: Set of disclosed attributes

$\text{Show} \leftarrow \text{ShowCred}(\text{cred}, \vec{A}, \vec{D})$

$0/1 \leftarrow \text{VerifCred}(\text{Show}, \vec{D}, \text{isk})$

Verifier

Issuer Secret Key (isk)

Different isk → ✘ Privacy ✘

Keyed-Verification Anonymous Credentials (KVAC)

Issuer

Issuer Secret Key (isk)

MAC using isk

User

Issuer Public Parameters (ipar)

Verifier

Issuer Secret Key (isk)

$$\text{Cred} \leftarrow \text{Issue}(\vec{A}, \text{isk}),$$

$$\pi \leftarrow \text{PoK}\{\text{isk}: \text{Cred}, \text{ipar}\}$$

Verify π

$$\text{Show} \leftarrow \text{ShowCred}(\text{cred}, \vec{A}, \vec{D})$$

$$0/1 \leftarrow \text{VerifCred}(\text{Show}, \vec{D}, \text{isk})$$

- Motivation
- KVAC Introduction
- **Problem Statement**
- Our First KVAC Construction
- Our Second KVAC Construction

KVAC with Partial Disclosure: Typical Approach

Issuer

Issuer Secret Key (isk)

Issuer Public Parameters (ipar)

User

Issuer Secret Key (isk)

Verifier

KVAC with Partial Disclosure: Typical Approach

Signer

Issuer Secret Key (isk)

User

Issuer Public Parameters (ipar)

Verifier

Issuer Secret Key (isk)

$$\tau \leftarrow \mathrm{MAC}(\vec{A}, \mathrm{isk})$$

$$\pi \leftarrow \mathrm{PoK}\{\mathrm{isk}: \tau, \mathrm{ipar}\} \xrightarrow{\tau, \pi}$$

KVAC with Partial Disclosure: Typical Approach

Issuer

Issuer Secret Key (isk)

User

Issuer Public Parameters (ipar)

Verifier

Issuer Secret Key (isk)

$\tau \leftarrow \mathrm{MAC}(\vec{A}, \mathrm{isk})$

$\pi \leftarrow \mathrm{PoK}\{\mathrm{isk}: \tau, \mathrm{ipar}\}$

τ, π →

Verify π

$(C_A, C_\tau) \leftarrow \mathrm{Commit}(\vec{A}, \tau; r_A, r_\tau)$

$\pi' \leftarrow \mathrm{PoK}\{\vec{A}, \tau, r_A, r_\tau : C_A, C_\tau\}$

$\mathrm{Show} := \mathrm{C_A}, \mathrm{C}_\tau, \pi'$ →

KVAC with Partial Disclosure: Typical Approach

Issuer

Issuer Secret Key (isk)

User

Issuer Public Parameters (ipar)

Verifier

Issuer Secret Key (isk)

$\tau \leftarrow \mathrm{MAC}(\vec{A}, \mathrm{isk})$

$\pi \leftarrow \mathrm{PoK}\{\mathrm{isk}: \tau, \mathrm{ipar}\}$

$\tau, \pi \longrightarrow$

Verify π

$(C_A, C_\tau) \leftarrow \mathrm{Commit}(\vec{A}, \tau; r_A, r_\tau)$

$\pi' \leftarrow \mathrm{PoK}\{\vec{A}, \tau, r_A, r_\tau : C_A, C_\tau\}$

$\mathrm{Show} := \mathrm{C_A}, \mathrm{C}_\tau, \pi' \longrightarrow$

$0/1 \leftarrow \mathrm{Verify}(C_A, C_\tau, \mathrm{isk})$

$\wedge \mathrm{Verify}(\pi')$

Problem with Typical Approach

Issuer | **User** | **Verifier**

Issuer Secret Key (isk)

Issuer Public Parameters (ipar)

Issuer Secret Key (isk)

$\tau \leftarrow \mathrm{MAC}(\vec{A}, \mathrm{isk})$

$\pi \leftarrow \mathrm{PoK}\{\mathrm{isk}: \tau, \mathrm{ipar}\}$

τ, π →

Verify π

$(C_A, C_\tau) \leftarrow \mathrm{Commit}(\vec{A}, \tau; r_A, r_\tau)$

$\pi' \leftarrow \mathrm{PoK}\{\vec{A}, \tau, r_A, r_\tau : C_A, C_\tau\}$

$\mathrm{Show} := \mathrm{C_A}, \mathrm{C}_\tau, \pi'$ →

$0/1 \leftarrow \mathrm{Verify}(C_A, C_\tau, \mathrm{isk})$

$\wedge \mathrm{Verify}(\pi')$

Our Research Question

Issuer | **User** | **Verifier**

Issuer Secret Key (isk)

Issuer Public Parameters (ipar)

Issuer Secret Key (isk)

$\tau \leftarrow \mathrm{MAC}(\vec{A}, \mathrm{isk})$

$\pi \leftarrow \mathrm{PoK}\{\mathrm{isk}: \tau, \mathrm{ipar}\}$

τ, π →

Verify π

$(C_A, C_\tau) \leftarrow \mathrm{Commit}(\vec{A}, \tau; r_A, r_\tau)$

$\pi' \leftarrow \mathrm{PoK}\{\vec{A}, \tau, r_A, r_\tau : C_A, C_\tau\}$

$\mathrm{Show} := \mathrm{C_A}, \mathrm{C}_\tau, \pi'$ →

$0/1 \leftarrow \mathrm{Verify}(C_A, C_\tau, \mathrm{isk})$

$\wedge \mathrm{Verify}(\pi')$

Can one design an efficient KVAC construction with selective disclosure where presentation does not rely on expensive zero-knowledge proofs?

Our KVAC Constructions

Scheme	Pairingless	Credential size	Presentation size	Security (Unforgeability, Anonymity)
$\mathrm{MAC}_{\mathrm{GGM}}$ + Schnorr [CMZ14]	✓	$2\lvert\mathbb{G}\rvert$	$(n+2)\lvert\mathbb{G}\rvert + (2n+2)\lvert\mathbb{Z}_p\rvert$	(GGM , DDH)
$\mathrm{MAC}_{\mathrm{BBS}}$ + Schnorr [BBDT16]	✓	$2\lvert\mathbb{G}\rvert + 2\lvert\mathbb{Z}_p\rvert$	$3\lvert\mathbb{G}\rvert + (n+7)\lvert\mathbb{Z}_p\rvert$	(q-SDH , Statistical)
$\mathrm{MAC}_{\mathrm{wBB}}$ + Optimized Schnorr [CDDH19]	✓	$(n+1)\lvert\mathbb{G}\rvert$	$\leq 2\lvert\mathbb{G}\rvert + (n+1)\lvert\mathbb{Z}_p\rvert$	(n-SCDHI, Statistical)
$\mathrm{MAC}_{\mathrm{GGM}}$ + O-DVNIZK [CR19]†	✓	$2n\lvert\mathbb{G}_{pq}\rvert$	$(n+2)\lvert\mathbb{G}_{pq}\rvert$	(GGM + IND-CPA + se-DL, Statistical)
μCMZ + Schnorr [Orr24]	✓	$2\lvert\mathbb{G}\rvert$	$(n+2)\lvert\mathbb{G}\rvert + (2n+2)\lvert\mathbb{Z}_p\rvert$	(AGM + 3-DL, Statistical)
μBBS + Schnorr [Orr24]	✓	$1\lvert\mathbb{G}\rvert + 1\lvert\mathbb{Z}_p\rvert$	$2\lvert\mathbb{G}\rvert + (n+4)\lvert\mathbb{Z}_p\rvert$	(AGM + q-DL, Statistical)
SP-MAC-EQ + DVSC ($\mathbf{KVAC_{MEQ}}$)	✗	$1\lvert\mathbb{G}_1\rvert + 1\lvert\mathbb{G}_2\rvert$	$4\lvert\mathbb{G}_1\rvert + 1\lvert\mathbb{G}_2\rvert$	(GGM, DDH)
Pairingless construction ($\mathbf{KVAC_{GGM}}$)	✓	$(n+2)\lvert\mathbb{G}\rvert$	$2\lvert\mathbb{G}\rvert$	(GGM, Statistical)

Our KVAC Constructions

Our First Construction

- ✓ Pairing-based
- ✓ Credential size: 2 group elements
- ✓ Presentation size: 5 group elements

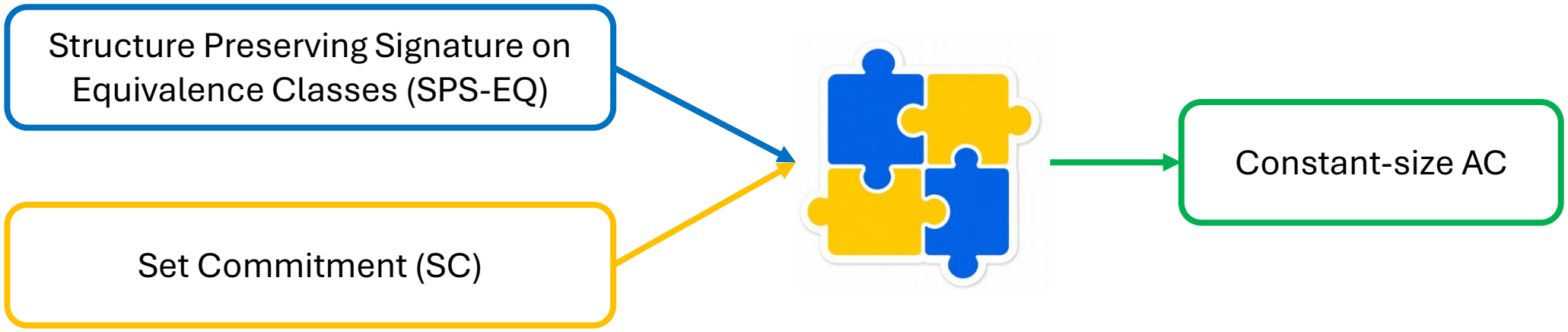
Our Second Construction

- ✓ Pairing-free
- ✓ Credential size: $n + 2$ group elements, where n is the number of attributes
- ✓ Presentation size: 2 group elements
- ✓ **Only requires one exponentiation for the verification**

- Motivation
- KVAC Introduction
- Problem Statement
- **Our First KVAC Construction**
- Our Second KVAC Construction

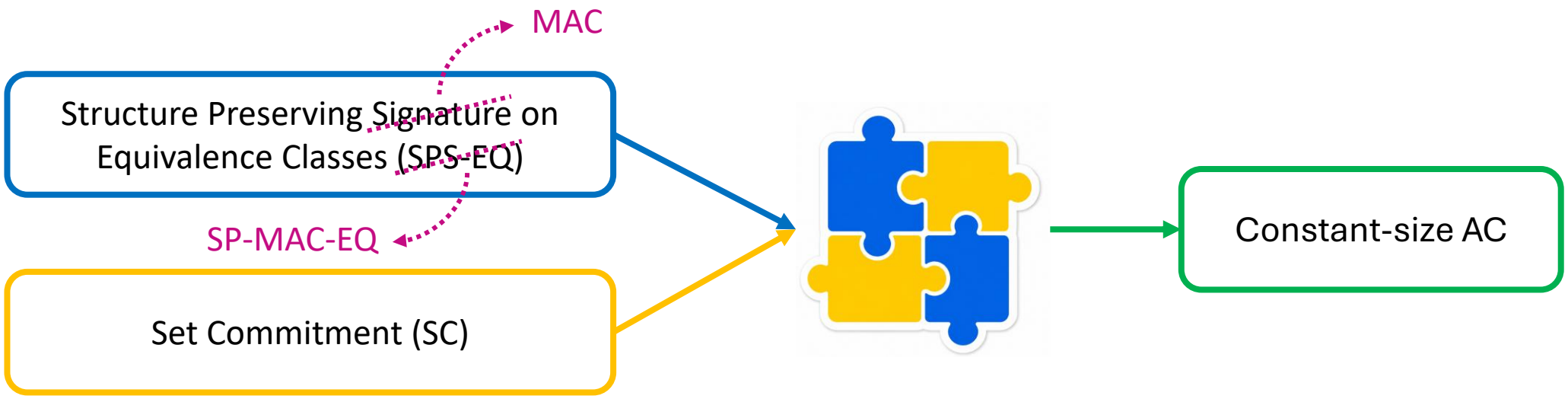
SPS-EQ and Anonymous Credentials

Inspired from: "Structure-Preserving Signatures on Equivalence Classes and Constant-Size Anonymous Credentials," Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig, *JoC 2019.*



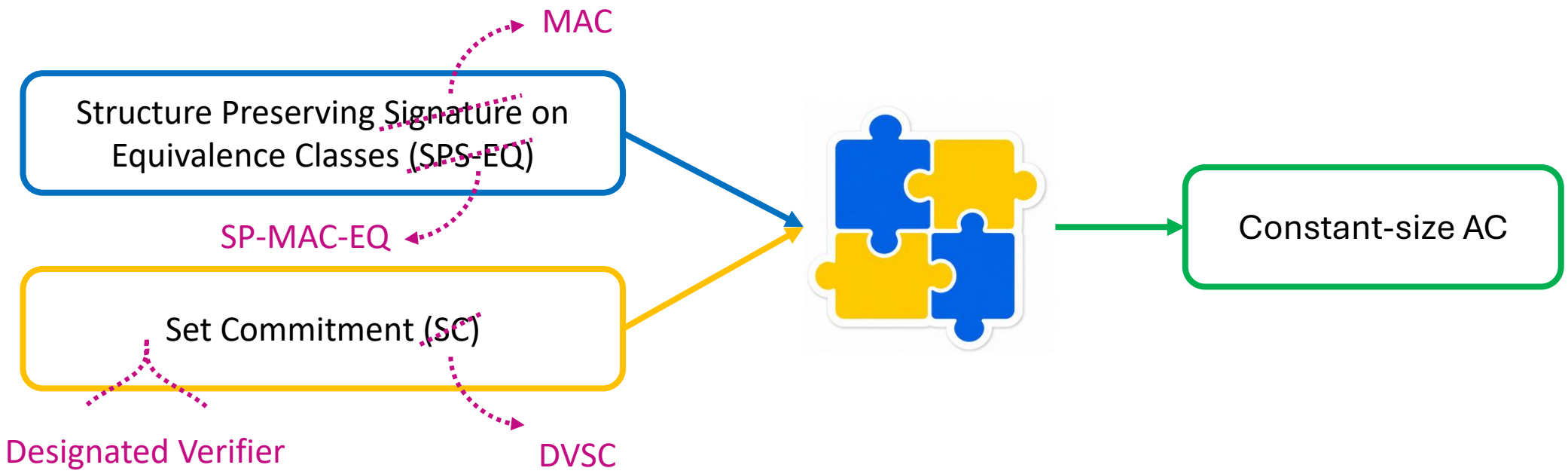
SPS-EQ and Anonymous Credentials

Inspired from: "Structure-Preserving Signatures on Equivalence Classes and Constant-Size Anonymous Credentials," Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig, *JoC 2019.*



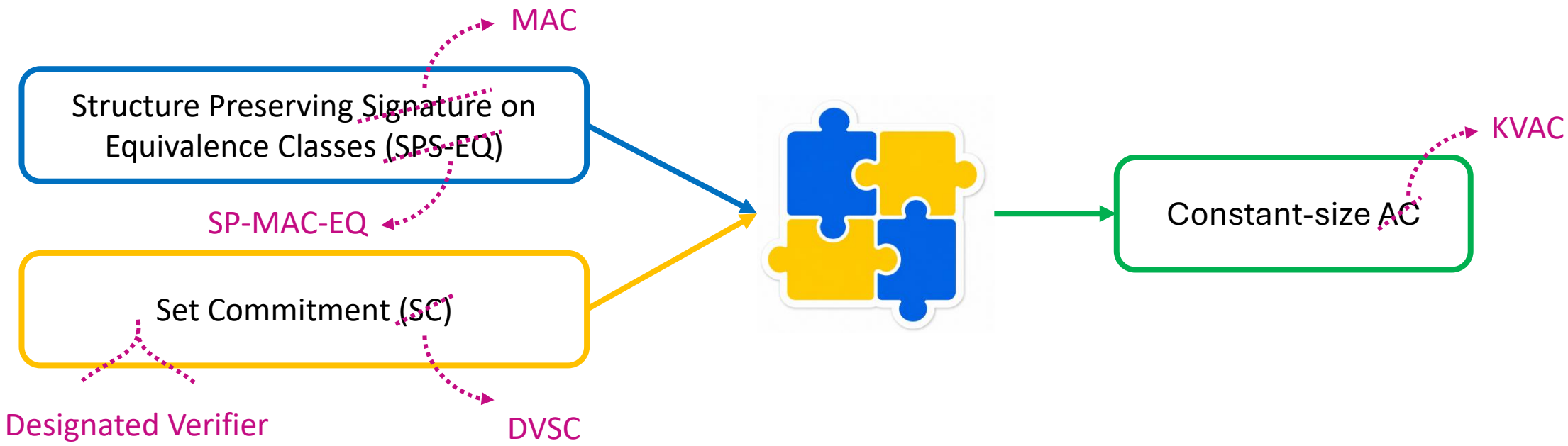
SPS-EQ and Anonymous Credentials

Inspired from: "Structure-Preserving Signatures on Equivalence Classes and Constant-Size Anonymous Credentials," Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig, *JoC 2019.*



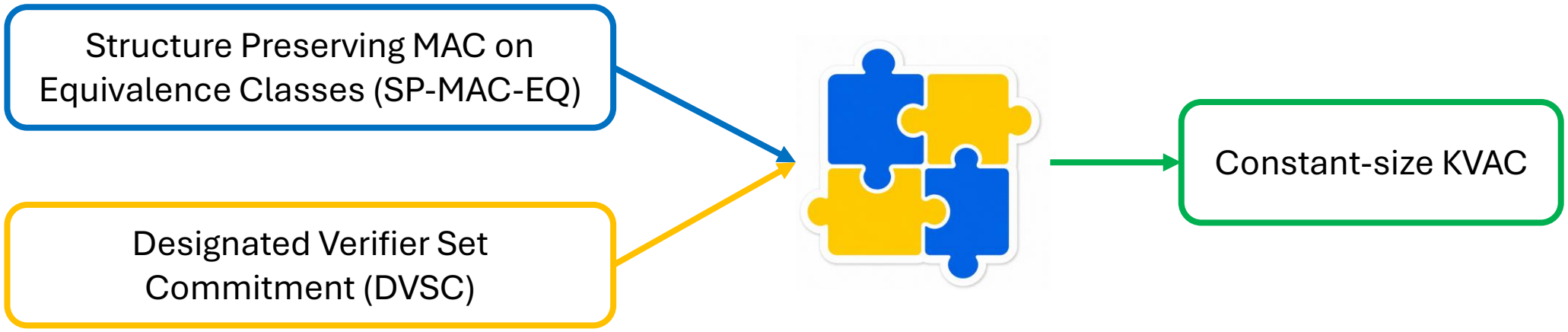
SPS-EQ and Anonymous Credentials

Inspired from: "Structure-Preserving Signatures on Equivalence Classes and Constant-Size Anonymous Credentials," Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig, *JoC 2019.*



Our First KVAC

Inspired from: "Structure-Preserving Signatures on Equivalence Classes and Constant-Size Anonymous Credentials," Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig, *JoC 2019.*



SP-MAC-EQ

Signer | **User** | **Verifier**

Signer Secret Key (isk)

Signer Secret Key (isk)

$$\tau \leftarrow \mathrm{MAC}(\vec{M}, \mathrm{isk})$$

τ is constant-size

$$\tau = (R, T)$$
$$:= (v, \sum \mathrm{isk}_i \cdot M_i, v^{-1} \mathrm{G}_2)$$

τ →

SP-MAC-EQ

Signer | **User** | **Verifier**

Signer Secret Key (isk)

Signer Secret Key (isk)

$\tau \leftarrow \mathrm{MAC}(\overrightarrow{\boldsymbol{M}}, \mathrm{isk})$

τ →

Sample random μ

$\left(\overrightarrow{\boldsymbol{M}'}, \tau'\right) \leftarrow \mathrm{ChangeRep}(\overrightarrow{\boldsymbol{M}}, \tau, \mu)$

$\overrightarrow{\boldsymbol{M}'} = \mu\overrightarrow{\boldsymbol{M}}$

$\tau' = (R', T') \coloneqq (\zeta\mu R, \zeta^{-1}T)$

$\left(\overrightarrow{\boldsymbol{M}'}, \tau'\right)$ →

SP-MAC-EQ

Signer

Signer Secret Key (isk)

User

Verifier

Signer Secret Key (isk)

$\tau \leftarrow \mathrm{MAC}(\overrightarrow{M}, \mathrm{isk})$

τ →

Sample random μ

$(\overrightarrow{M'}, \tau') \leftarrow \mathrm{ChangeRep}(\overrightarrow{M}, \tau, \mu)$

$(\overrightarrow{M'}, \tau')$ →

$\mathrm{Verify}(\overrightarrow{M'}, \tau', \mathrm{isk}) = 1$

$(\overrightarrow{M'}, \tau')$ *look random*

$e(\sum \mathrm{isk}_i \cdot M'_i, \mathrm{G}_2) = e(R', T')$

SP-MAC-EQ only requires 2 pairings while SPS-EQ requires $n + 3$ pairings

Designated Verifier Set Commitment

User

Public parameters (par)

Verifier

Secret Key (sk)

$$C \coloneqq \mathrm{Commit}(\vec{A}, \mathrm{par})$$

C is constant-size

$$f_{\vec{A}}(x) \coloneqq \prod_{a \in \vec{A}} (x - a) = \sum_{i=0}^{|\vec{A}|} f_i x^i$$

$$C \coloneqq f_{\vec{A}}(\mathrm{sk})\mathrm{G} = \sum_{i=0}^{|\vec{A}|} f_i \mathrm{sk}^i \mathrm{G}$$

Designated Verifier Set Commitment

User

Public parameters (par)

Verifier

Secret Key (sk)

$C \coloneqq \mathrm{Commit}(\vec{A}, \mathrm{par})$

Sample random μ

$C' \leftarrow \mathrm{Randomize}(\mathrm{C}, \mu)$

$\vec{D} \subseteq \vec{A}: W \leftarrow \mathrm{Open}(\vec{A}, \vec{D}, \mu, \mathrm{par})$

C', W are constant-size

$C' \coloneqq \mu C, W \coloneqq \mu f_{\vec{A}\setminus\vec{D}}(\mathrm{sk})\mathrm{G}$

C', W →

Designated Verifier Set Commitment

User

Public parameters (par)

Verifier

Secret Key (sk)

$C \coloneqq \mathrm{Commit}(\vec{A}, \mathrm{par})$

Sample random μ

$C' \leftarrow \mathrm{Randomize}(\mathrm{C}, \mu)$

$W \leftarrow \mathrm{Open}(\vec{A}, \vec{D}, \mu, \mathrm{par})$

C', W →

$\mathrm{Verify}(\vec{D}, W, C', \mathrm{sk}) = 1$

(C', W) *look random*

$C' = f_{\vec{D}}(\mathrm{sk})W$

Our First KVAC Construction

Issuer

Issuer Secret Key (isk)

$C \coloneqq \text{Commit}(\vec{\boldsymbol{A}}, \text{ipar})$

$\tau \leftarrow \text{MAC}(C, \text{isk})$

$\pi \leftarrow \text{PoK}\{\text{isk}: \tau, \text{ipar}\}$

τ, π

User

Issuer Public Parameters (ipar)

Verify π

$C \coloneqq \text{Commit}(\vec{\boldsymbol{A}}, \text{ipar})$

Sample random μ

$(C', \tau') \leftarrow \text{ChangeRep}(C, \tau, \mu)$

$W \leftarrow \text{Open}(\vec{\boldsymbol{A}}, \vec{\boldsymbol{D}}, \mu, \text{ipar})$

W, C', τ'

Verifier

Issuer Secret Key (isk)

$0/1 \leftarrow \text{VerifySPMAC}(C', \tau', \text{isk})$

$\wedge \text{VerifyDVSC}(\vec{\boldsymbol{D}}, W, C', \text{isk})$

Our First KVAC Construction

Issuer

Issuer Secret Key (isk)

$C \coloneqq \text{Commit}(\vec{A}, \text{ipar})$

$\tau \leftarrow \text{MAC}(C, \text{isk})$

$\pi \leftarrow \text{PoK}\{\text{isk}: \tau, \text{ipar}\}$

User

Issuer Public Parameters (ipar)

τ, π →

Verify π

$C \coloneqq \text{Commit}(\vec{A}, \text{ipar})$

Sample random μ

$(C', \tau') \leftarrow \text{ChangeRep}(C, \tau, \mu)$

$W \leftarrow \text{Open}(\vec{A}, \vec{D}, \mu, \text{ipar})$

W, C', τ' →

Verifier

Issuer Secret Key (isk)

$0/1 \leftarrow \text{VerifySPMAC}(C', \tau', \text{isk})$

$\wedge \text{VerifyDVSC}(\vec{D}, W, C', \text{isk})$

But the verification needs pairing!

- Motivation
- KVAC Introduction
- Problem Statement
- Our First KVAC Construction
- **Our Second KVAC Construction**

Idea for the second KVAC Construction

Issuer

Issuer Secret Key (isk)

$C := \text{Commit}(\vec{A}, \text{ipar})$

$\tau \leftarrow \text{isk} \cdot C$

$\pi \leftarrow \text{PoK}\{\text{isk}: \tau, \text{ipar}\}$

τ, π

User

Issuer Public Parameters (ipar)

Verify π

$C := \text{Commit}(\vec{A}, \text{ipar})$

Sample random μ

$(C', \tau') \leftarrow (\mu C, \mu\tau)$

$W \leftarrow \text{Open}(\vec{A}, \vec{D}, \mu, \text{ipar})$

W, C', τ'

Verifier

Issuer Secret Key (isk)

$0/1 \leftarrow (\tau' == \text{isk} \cdot C')$

$\wedge \text{VerifyDVSC}(\vec{D}, W, C', \text{isk})$

Idea for the second KVAC Construction

Issuer

Issuer Secret Key (isk)

$C \coloneqq \text{Commit}(\vec{\boldsymbol{A}}, \text{ipar})$

$\tau \leftarrow \text{isk} \cdot C$

$\pi \leftarrow \text{PoK}\{\text{isk}: \tau, \text{ipar}\}$

Issuer Public Parameters (ipar)

User

τ, π →

Verify π

$C \coloneqq \text{Commit}(\vec{\boldsymbol{A}}, \text{ipar})$

Sample random μ

$(C', \tau') \leftarrow (\mu C, \mu\tau)$

$W \leftarrow \text{Open}(\vec{\boldsymbol{A}}, \vec{\boldsymbol{D}}, \mu, \text{ipar})$

This is forgeable!

Verifier

Issuer Secret Key (isk)

W, C', τ' →

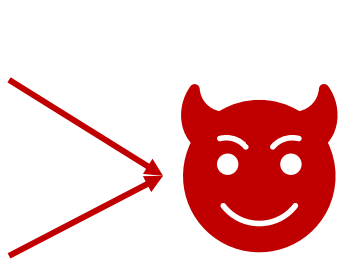
$0/1 \leftarrow (\tau' == \text{isk} \cdot C')$

$\wedge \text{VerifyDVSC}(\vec{\boldsymbol{D}}, W, C', \text{isk})$

Idea for the second KVAC Construction

$C_1 \coloneqq \mathrm{Commit}(\boldsymbol{A}_1, \mathrm{ipar}), \tau_1 \coloneqq \mathrm{isk} \cdot C_1$

$C_2 \coloneqq \mathrm{Commit}(\boldsymbol{A}_2, \mathrm{ipar}), \tau_2 \coloneqq \mathrm{isk} \cdot C_2$



$C_3 \coloneqq \mathrm{C}_1 + C_2, \tau_3 \coloneqq \tau_1 + \tau_2$

τ_3 is a valid tag on C_3

Easy to find $\boldsymbol{A}_3$: $C_3 = \mathrm{Commit}(\boldsymbol{A}_3, \mathrm{ipar})$

Idea for the second KVAC Construction

$C_1 \coloneqq \mathrm{Commit}(\boldsymbol{A}_1, \mathrm{ipar}), \tau_1 \coloneqq \mathrm{isk} \cdot C_1$

$C_2 \coloneqq \mathrm{Commit}(\boldsymbol{A}_2, \mathrm{ipar}), \tau_2 \coloneqq \mathrm{isk} \cdot C_2$

$C_3 \coloneqq \mathrm{C}_1 + C_2, \tau_3 \coloneqq \tau_1 + \tau_2$

τ_3 is a valid tag on C_3

Easy to find $\boldsymbol{A}_3$: $C_3 = \mathrm{Commit}(\boldsymbol{A}_3, \mathrm{ipar})$

Should be made difficult

Idea for the second KVAC Construction

Issuer

Issuer Secret Key (isk)

Sample random y

$C \leftarrow \text{Commit}(\vec{A}, \text{ipar}, y)$

$\tau \leftarrow \text{isk} \cdot C$

compute aux_y

$\pi \leftarrow \text{PoK}\{\text{isk}: \tau, \text{ipar}, \text{aux}_y\}$

τ, aux_y, π →

User

Issuer Public Parameters (ipar)

Verify π

$C \coloneqq \text{Commit}(\vec{A}, \text{aux}_y)$

Sample random μ

$(C', \tau') \leftarrow (\mu C, \mu\tau)$

$W \leftarrow \text{Open}(\vec{A}, \vec{D}, \mu, \text{aux}_y)$

W, C', τ' →

Verifier

Issuer Secret Key (isk)

$0/1 \leftarrow (\tau' == \text{isk} \cdot C')$

$\wedge \text{VerifyDVSC}(\vec{D}, W, C', \text{isk})$

Thanks for your attention!

Full paper