Delegatable Attribute-based Anonymous Credentials from Dynamically Malleable Signatures

Johannes Blömer, **Jan Bobolz**

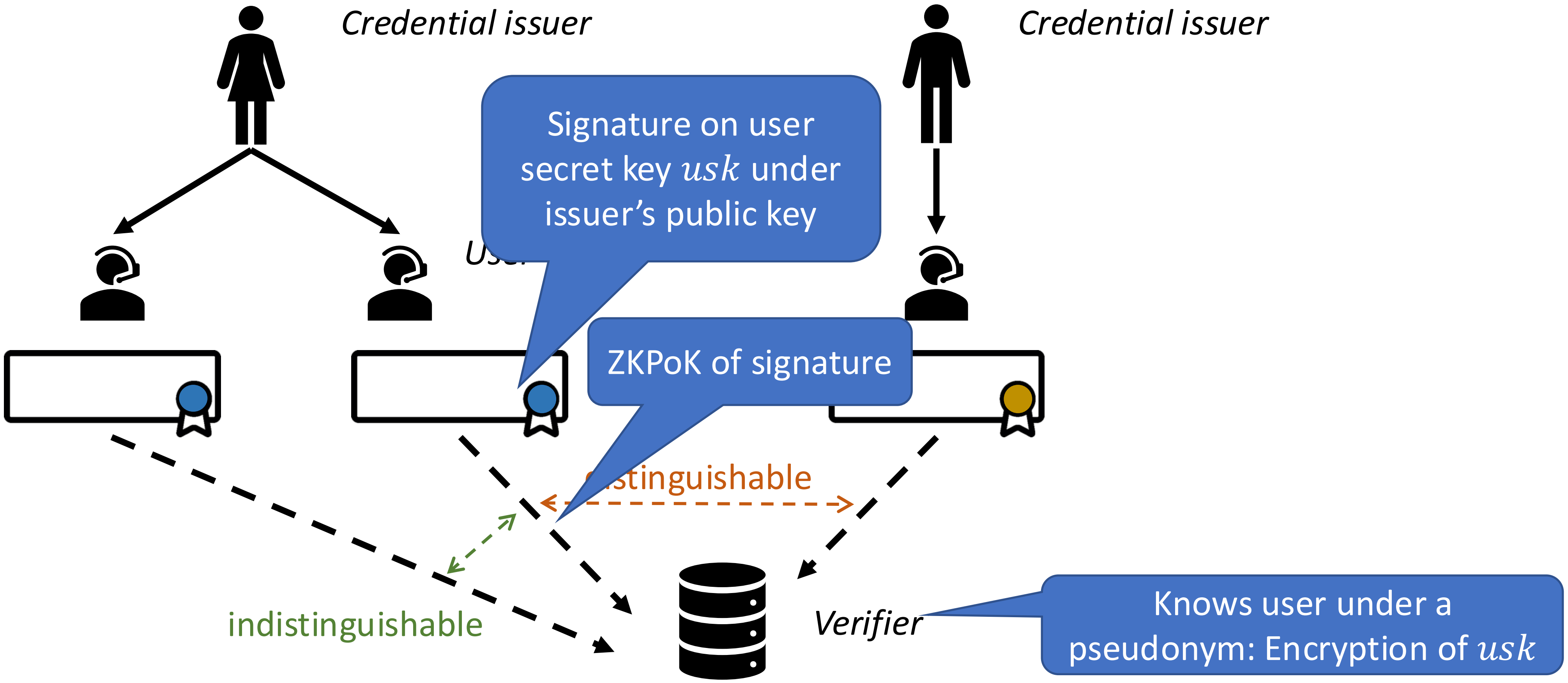
Paderborn University, Germany

SFB 901

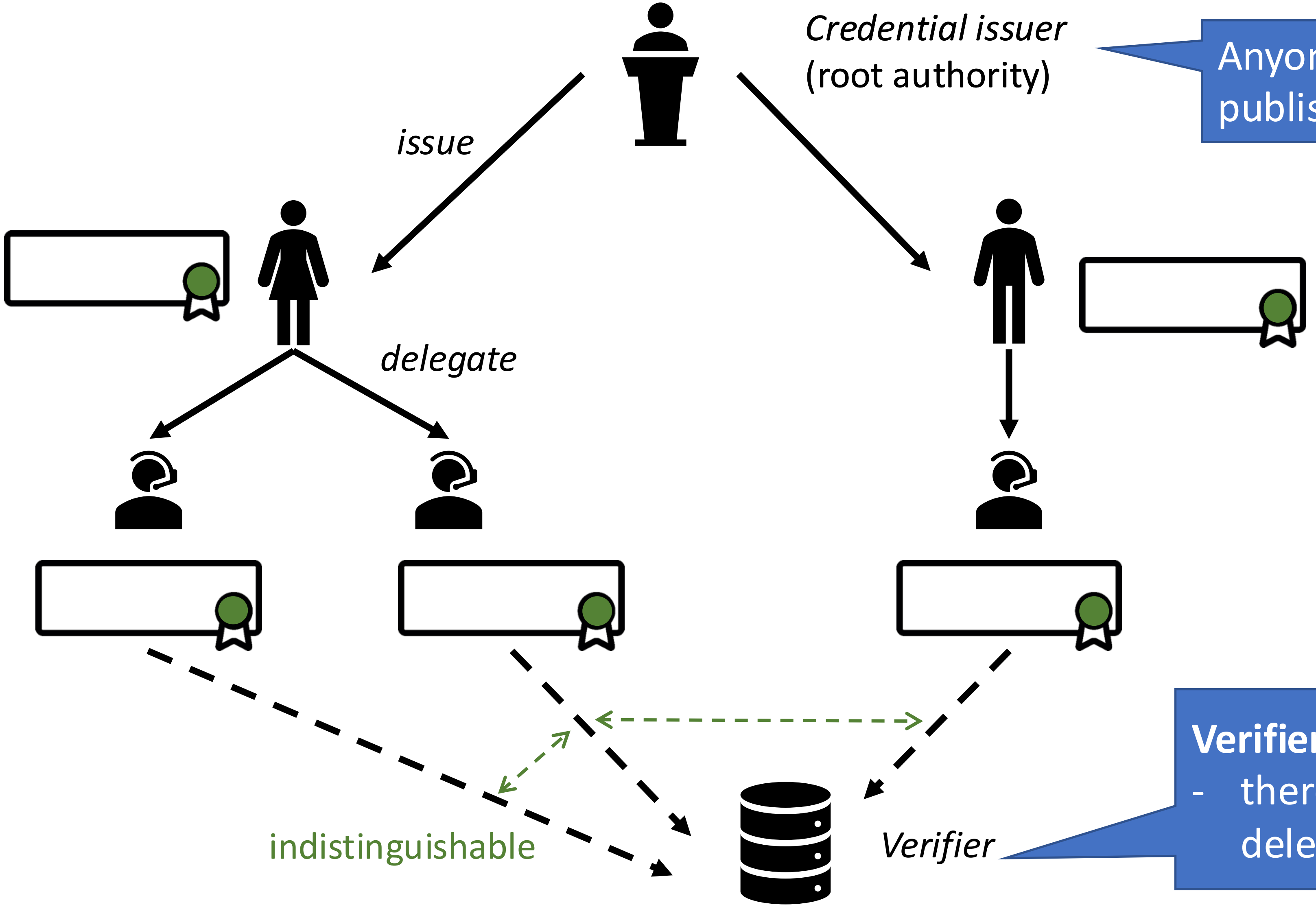
ON - THE - FLY COMPUTING

PADERBORN UNIVERSITY

Quick refresher: Anonymous credentials



Delegatable credentials (chain of trust)



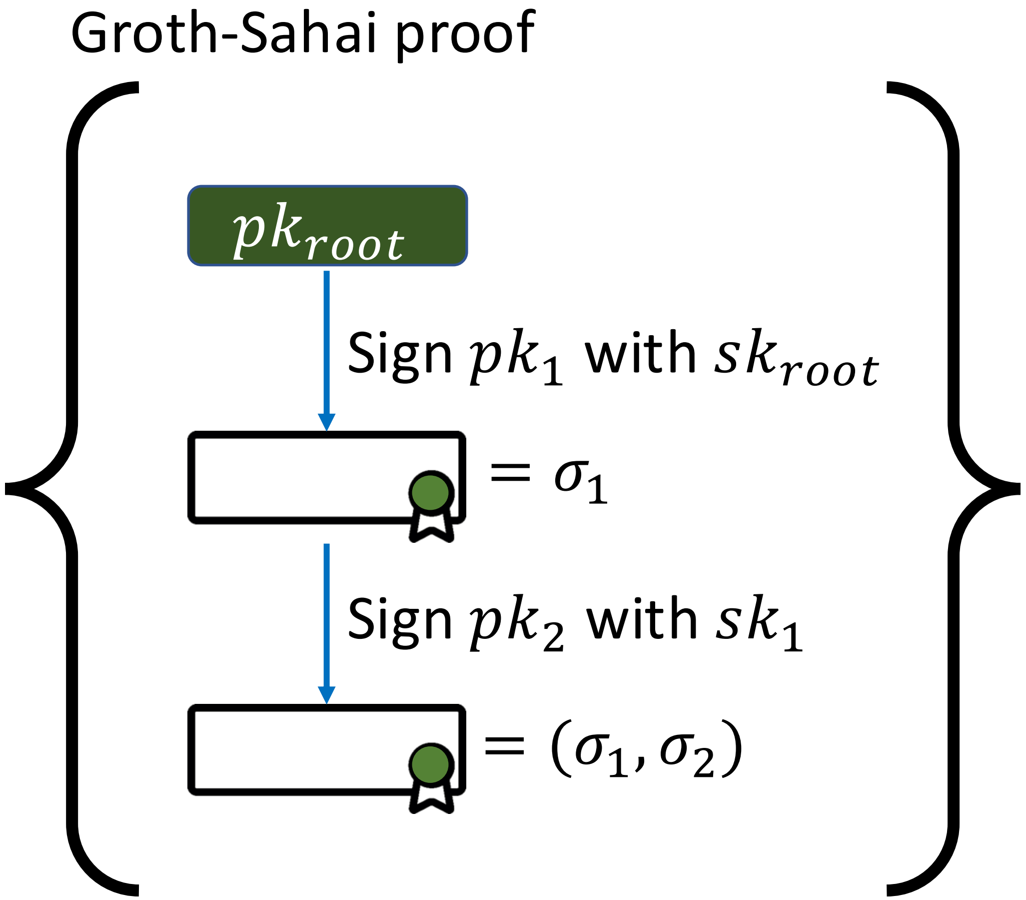
Anyone can become root issuer by publishing a pseudonym pk_{root}.

Verifier learns:
- there is a valid chain of credential delegations rooted at pk_{root}.

Prior construction ideas

- Idea: certificate chains (as in TLS)
- e.g., [CDD17]

- Idea: credential is GS proof of knowledge of chain, hiding pk_i, σ_i.
- Delegation: Use malleability of GS to extend chain.
- Show: just send (re-randomized) credential
- [BCCKLS09] and then many more

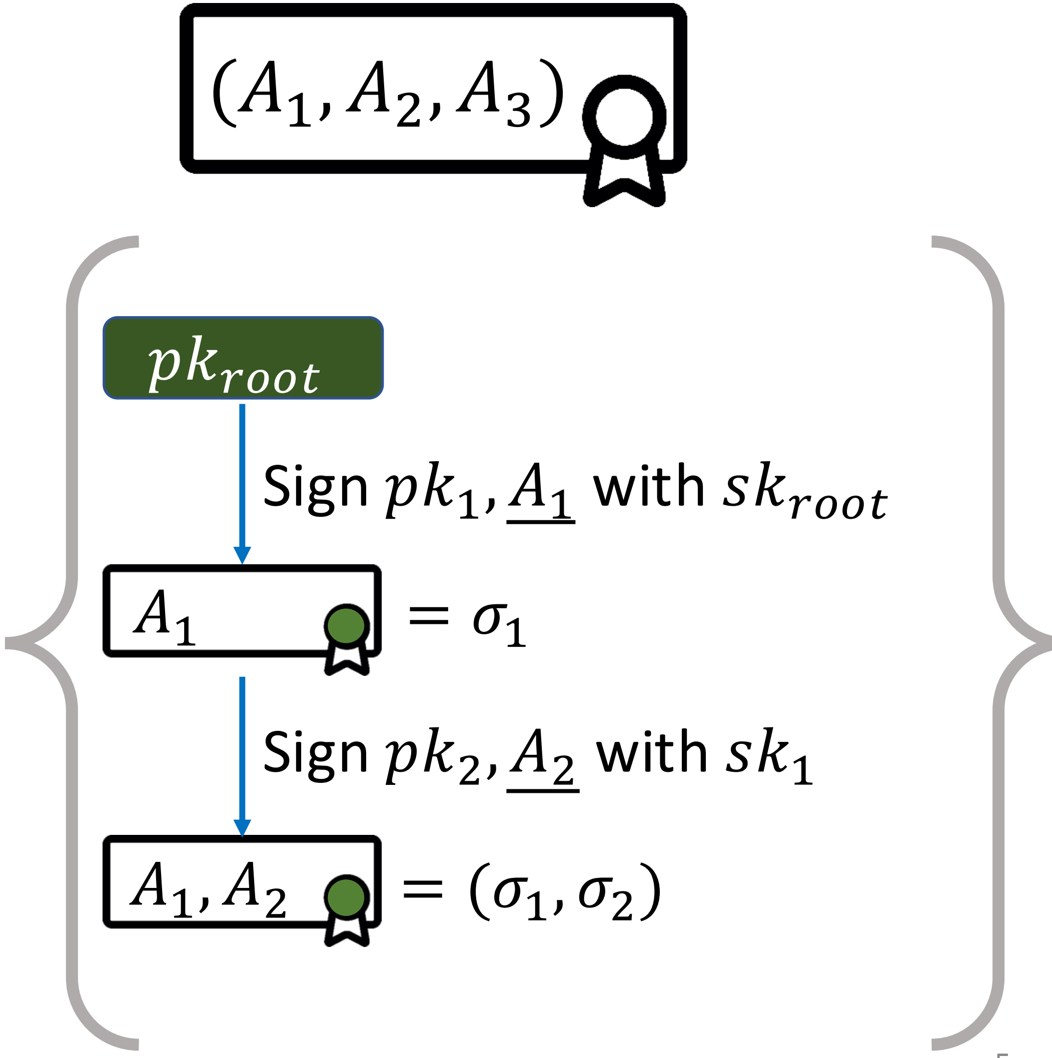


[CDD17] Jan Camenisch, Manu Drijvers, and Maria Dubovitskaya. *Practical UC-secure delegatable credentials with attributes and their application to blockchain*. ACM CCS 2017

[BCCKLS09] Mira Belenkiy, Jan Camenisch, Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Hovav Shacham. *Randomizable proofs and delegatable anonymous credentials*. CRYPTO 2009

Credentials with attributes

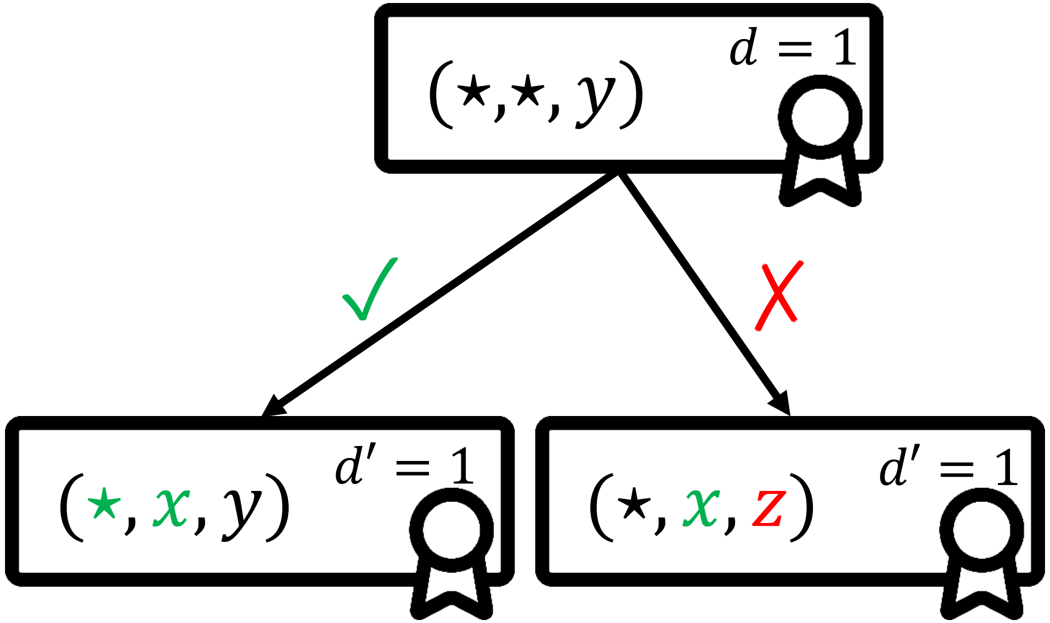
- Credentials have associated *attributes*.
  - e.g., `(Name, Access level,…)`
- Question: given credential with certain attributes, what can I delegate?
- Prior work: can *append* new attributes
  - given (A_1), can delegate (A_1, A_2)
- **But**: credential holder must know all attributes
  ⇒ **No anonymity for delegators**



Our model of attribute delegation

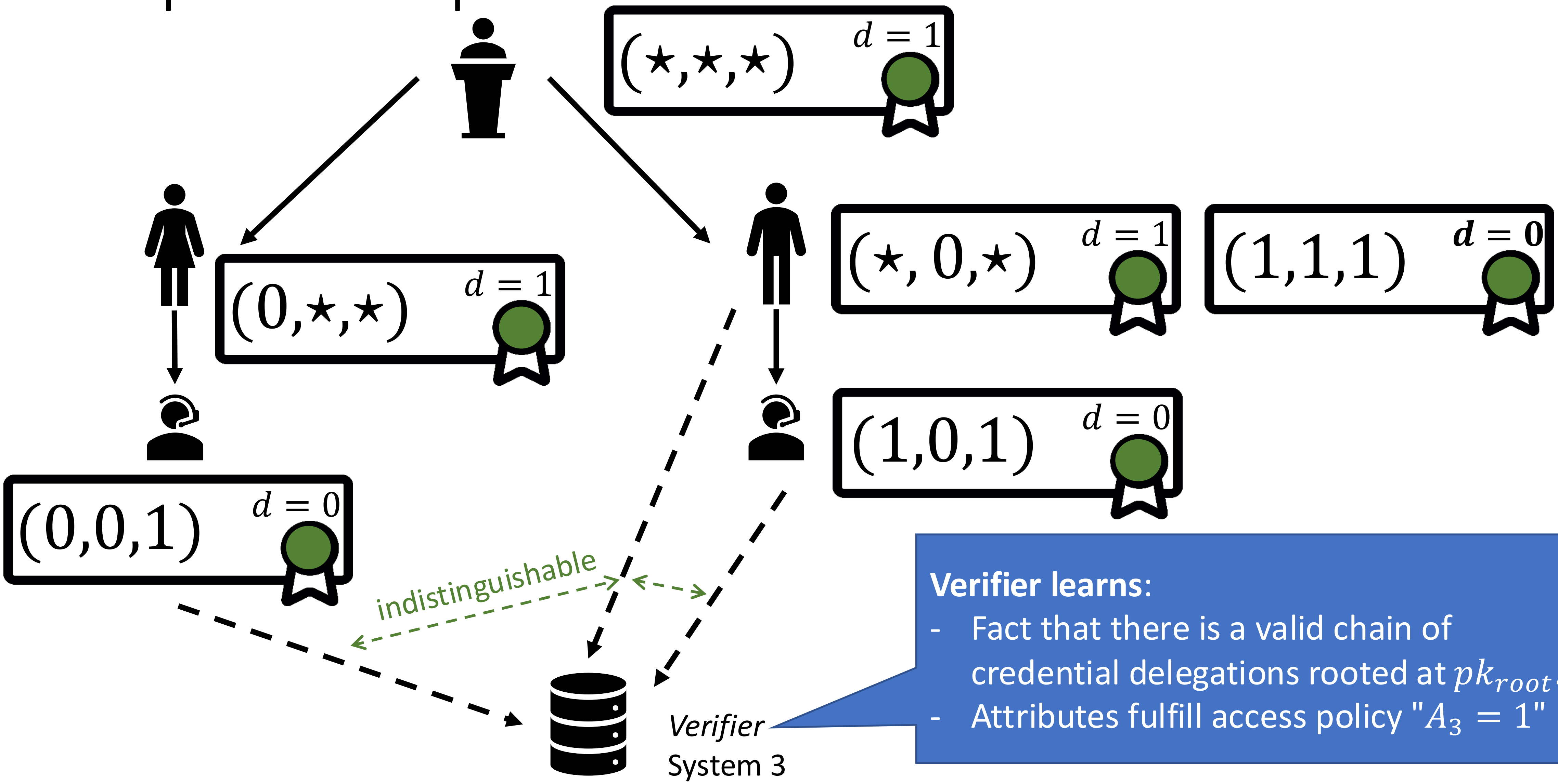
Delegation of Attributes *in our construction*

- Idea: attributes determine what I can delegate.
- Credential parameterized with:
  - Attributes $\vec{A} = (A_1, \ldots, A_n) \in (\{0,1\}^* \cup \{\star\})^n$
  - Delegation flag $d \in \{0,1\}$
- Can *issue* someone else $(A'_1, \ldots, A'_n)$ with delegation flag $d' \in \{0,1\}$ iff
  - $d = 1$ (my own delegation flag is 1)
  - and $A_i \neq \star \;\; \Rightarrow \;\; A_i = A'_i$
- Can prove having attributes $(A'_1, \ldots, A'_n)$ if
  - $A_i \neq \star \;\; \Rightarrow \;\; A_i = A'_i$



Policies: need $A_i = 1$ to access system i.

Simple example



Verifier learns:

- Fact that there is a valid chain of credential delegations rooted at pk_{root}.
- Attributes fulfill access policy "$A_3 = 1$"

Building block: DMS

Dynamically malleable signatures (DMS)

Dynamically malleable signatures

A DMS scheme works as follows:

- $\mathrm{KeyGen}(1^\lambda, 1^n) \to (pk, sk)$
- $\mathrm{Sign}(sk, \vec{m}, \boldsymbol{I}) \to (\sigma, \boldsymbol{mk})$ for $\vec{m} \in M^n, \boldsymbol{I} \subseteq \{\mathbf{1}, \ldots, \boldsymbol{n}\}$ outputs signature σ **and malleability key $\boldsymbol{mk}$.**
- $\mathrm{Verify}(pk, \vec{m}, \sigma)$ verifies the signature.
- $\mathbf{Transform}(\vec{m}, \vec{m}', \sigma, mk, I') \to (\sigma', mk')$
  - Outputs (σ', mk') with same distribution as $\mathrm{Sign}(sk, \vec{m}', I')$.
  - Required: $\vec{m}_i = \vec{m}_i'$ for all $i \notin I$, and $I' \subseteq I$.

Signature malleable at positions in I.

DMS unforgeability definition

- Adversary A gets pk, can query oracle

$$\left(\vec{m}_j, I_j\right) \mapsto \left(\sigma_j, mk_j\right) \leftarrow \mathrm{Sign}\left(sk, \vec{m}_j, I_j\right)$$

- Eventually A outputs $\vec{m}^*, \sigma^*$
- A wins if
  - $\mathrm{Verify}(pk, \vec{m}^*, \sigma^*) = 1$ and
  - Signature on $\vec{m}^*$ cannot be trivially derived from any queried $\left(\vec{m}_j, I_j\right)$

Prevents collusion: individual signatures cannot be combined.

Example DMS: extension of [PS16]

- $sk = (x, y_1, \dots, y_n) \in \mathbb{Z}_p^{n+1},\ pk = (\tilde{g}, \tilde{g}^x, \tilde{g}^{y_1}, \dots, \tilde{g}^{y_n}) \in G_2^{n+2}$

- Signatures: $\left(h, h^{x+\sum y_i m_i}\right)$ for random $h \leftarrow G_1 \setminus \{1\}$
- Malleability key: $\mathrm{mk} = \{\, h^{y_i} \mid i \in I \,\}$

- $\mathrm{Transform}(\vec{m}, \vec{m}', \sigma, mk, I') \rightarrow (\sigma', mk')$
  - Compute $h^{\sum y_i m_i'} = h^{\sum y_i m_i} \cdot \prod_{i \in I} (h^{y_i})^{m_i' - m_i}$
  - Re-randomize resulting signature $\left(h, h^{\sum y_i m_i'}\right)$ as $\sigma' = \left(h^r, \left(h^{\sum y_i m_i'}\right)^r\right)$
  - Re-randomize and weaken mk to $mk' = \{\, (h^{y_i})^r \mid i \in I' \,\}$.

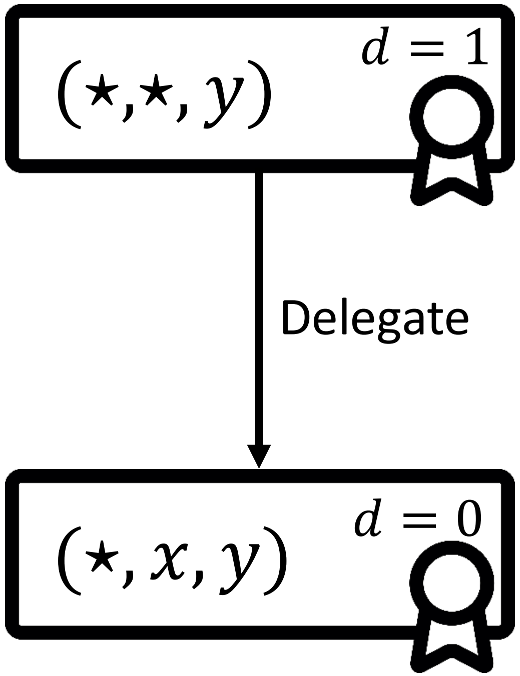
[PS16] Pointcheval D., Sanders O. (2016) Short Randomizable Signatures. CT-RSA 2016

DAAC from DMS

Delegatable attribute-based anonymous credentials from dynamically malleable signatures (generically)

Delegatable credentials from DMS

- Standard template for (non-delegatable!) attribute-based credentials:
  - But use DMS malleability to implement “⋆” placeholder and delegation.



σ on $(\underline{usk}, \underline{0}, \underline{0}, y)$ and mk for $I = \{0,1,2\}$

(Blind) transform

σ' on $(usk', \underline{0}, x, y)$ and mk' for $I = \{1\}$

Pseudonyms:
$\mathrm{Enc}(\mathrm{OWP}(usk))$
+ DMS pk

Show:
ZKPoK of σ (...)

Open:
Decrypt Pseudonym

Overview/Comparison

Scheme	Delegate anonymously	Hide attributes from verifier	Attribute delegation restrictions	Extract delegation chain?	Efficiency	Assumption (including concrete building blocks)
This work (using PS-style DMS)	Yes	Yes. Arbitrary Boolean formulae	Using “⋆” during delegation	No, only first and last.	Efficient Schnorr. O(1) proofs and credentials.	For PS-DMS: Non-standard, interactive.
[BCCKLS09] + related work	Yes (no attributes) No (with attributes)	Attribute chain revealed	Checked by verifier.	Yes	Groth-Sahai proofs. O(L) proofs and credentials.	DLIN construction available [CKLM14]
[CDD17]	No (all attributes revealed)	Arbitrary Boolean formulae	Proven in zk during *show*.	Yes	Efficient Schnorr. O(L) proofs and credentials.	Groth-Sibling Sig: GGM

[CKLM14] Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Sarah Meiklejohn. Malleable signatures: New definitions and delegatable anonymous credentials. CSF 2014 & ePrint 2013/179 (full)

That’s it.

Example: Textual attributes and proxying

- City of Paderborn gets credential $(\star, \star, \text{Paderborn}, \star)$ from federal state.
- With that credential, the city can issue passport credentials $(\text{John}, \text{Doe}, \text{Paderborn}, \star)$ for Paderborn's citizens but not for Berlin's.
- When user wants to let someone else proxy him for some transaction, he can delegate a credential $(\text{John}, \text{Doe}, \text{Paderborn}, \text{"may fetch my paycheck"})$ to someone else.

Example: Anonymous delegation

- Car owner is issued credential for his car.
- Can delegate credential to any (unknown) person for car sharing.
  - May include expiry date and some other restrictions as attributes.
- Neither car owner nor car loaner have a reason to reveal their identity.
- In case of dispute, a judge can trace pseudonyms.

Examples

Access rights in an organization: attributes are Booleans "service i access granted/not granted".

- e.g., manager gets two credentials: $(\star, \star, 0, 0)$, $d = 1$ and $(1, 1, 1, 0)$, $d = 0$ from his superior.
- extension: read/write/both.

Door access credentials: attributes are *buildingId*, *doorId*, *expires*.

- Landlord gets $(3, \star, \star)$ from authority.
- Tenant gets $(3, 1, \star)$ from landlord.
- Tenant's guests get $(3, 1, 1484218277)$, $d \in \{0, 1\}$.

Service provider sells access to the university: attributes are *name*, *accessLevel*, and *priceLimit*.

- University gets credentials: $[(\star, i, \star), d = 1]_{i=1}^{5}$ from service provider.
- Students are delegated $(\text{"Alice"}, 3, 100)$, $d = 0$.

Other special signatures

- (Invisible) sanitizable signatures
  - Specific sanitizing party. Only one level (sanitized signature cannot be sanitized again).
- Malleable signatures
  - Allowed malleability relation usually fixed in the beginning, cannot be weakened for a signature over time.
- Homomorphic signatures
  - Can combine multiple signatures, but all signatures from a single signer can be freely combined.
  - Basically our *PS-inspired construction without re-randomization*.