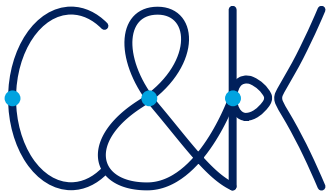
Updatable Anonymous Credentials and Applications to Incentive Systems

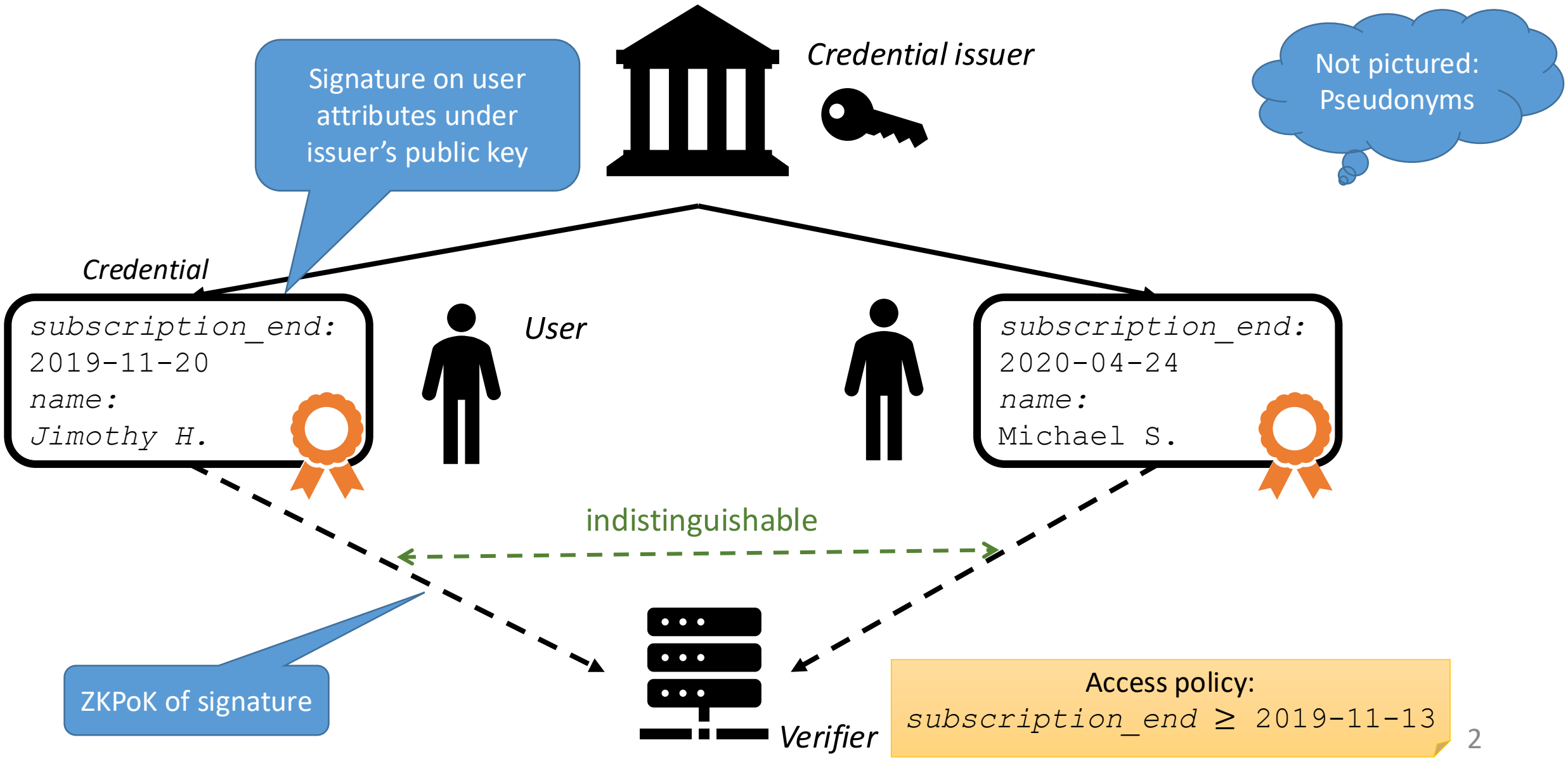
Johannes Blömer, **Jan Bobolz**, Denis Diemert, Fabian Eidens

Paderborn University

Codes & Cryptography group



Anonymous Credentials

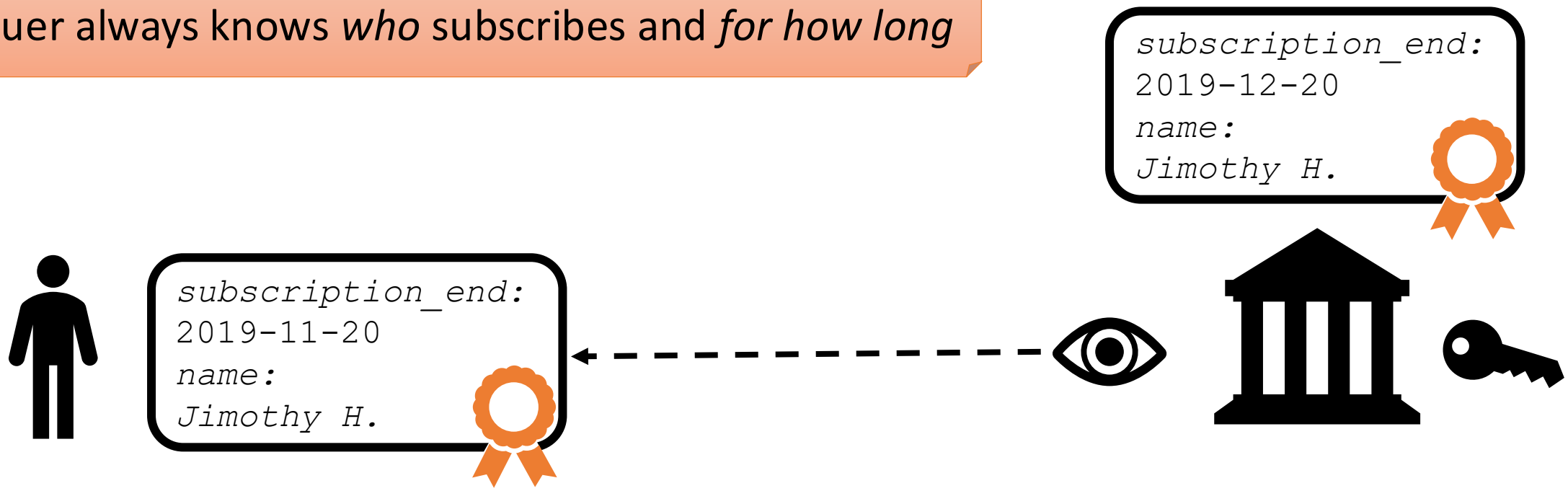


Updating attributes in credential systems

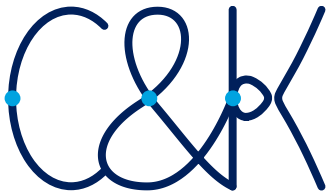
Naive:

- User *reveals all attributes* and shows old credential to issuer.
- Issuer issues new credential on new attributes.

Issuer always knows *who* subscribes and *for how long*



Updatable Credentials

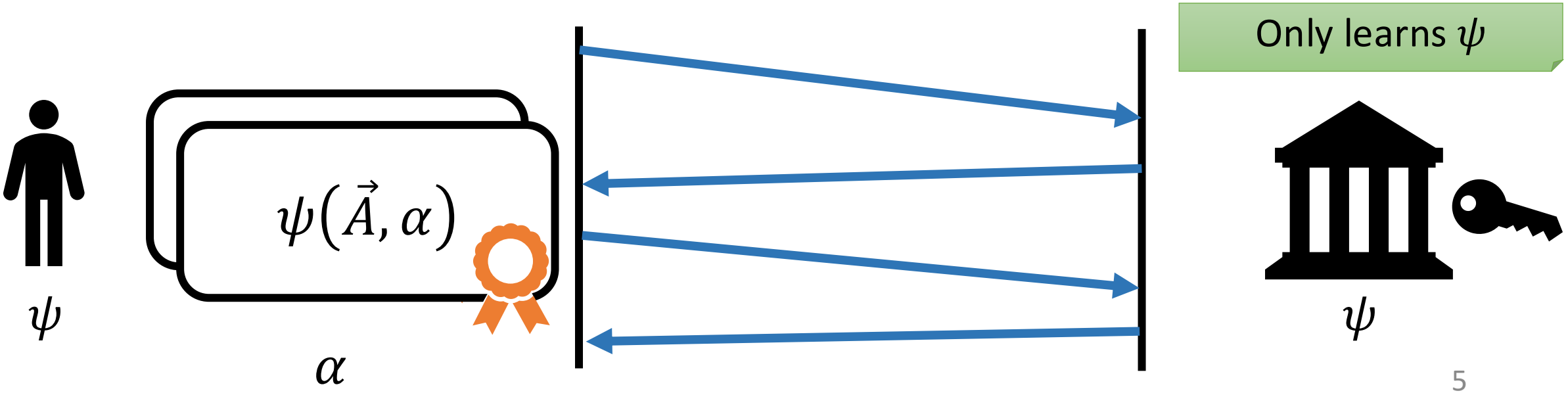


Updating anonymous credentials

User and issuer **agree on update function** ψ.

***Input** of* ψ: Old attributes $\vec{A}$ and hidden parameter α

***Output** of* ψ: New attributes $\vec{A}^*$ or error $\perp$



Examples

Extending subscription:

$$\psi\big((end, name), \alpha\big) = (end + 30, name)$$

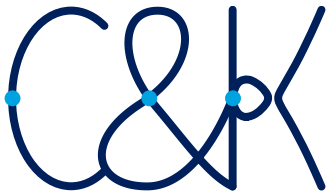
Issuer knows: "I'm adding 30 to *end*"

Issuer knows ψ
and learns whether $\psi\big(\vec{A}, \alpha\big) = \perp$

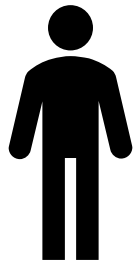
Abstract example with α:

$$\psi\big((x, y), \alpha\big) = \begin{cases} (x, y + \alpha) & \textbf{if } y + \alpha < x \\ \perp & \textbf{otherwise} \end{cases}$$

Constructing updatable credentials



Construction: Update Protocol



ψ

$\vec{A}$

α

$\vec{A}^* = \psi(\vec{A}, \alpha)$

$\vec{A}^*$

Pedersen

Commit to $\vec{A}^*$

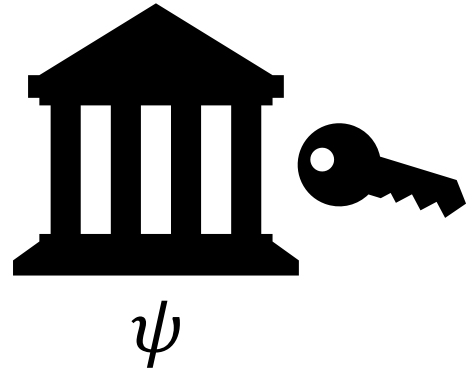
[CL04] or [PS16]

Zero-knowledge proof of knowledge

of $\vec{A}, \alpha, \vec{A}^*$, $\vec{A}$:

- $\vec{A}$ is a valid signature on $\vec{A}$
- Can open commitment to $\vec{A}^*$
- $\vec{A}^* = \psi(\vec{A}, \alpha) \neq \perp$

Variants of Schnorr



ψ

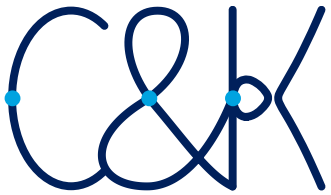
[CL04] or [PS16]

BlindSign$(Commitment)$

[CL04] Camenisch,Lysyanskaya. *Signature schemes and anonymous credentials from bilinear maps*. Crypto 2004

[PS16] Pointcheval, Sanders. *Short randomizable signatures*. CT-RSA 2016

Incentive Systems in Practice



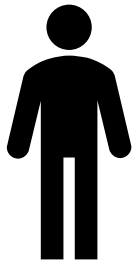
Incentive systems

Point collection

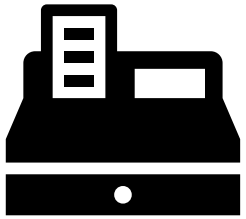
- Collect points
  - e.g., 1 point for every item I buy
- Spend points for reward

Examples:

- Loyalty cards (/trading stamps)
- Bonus miles
- …



User



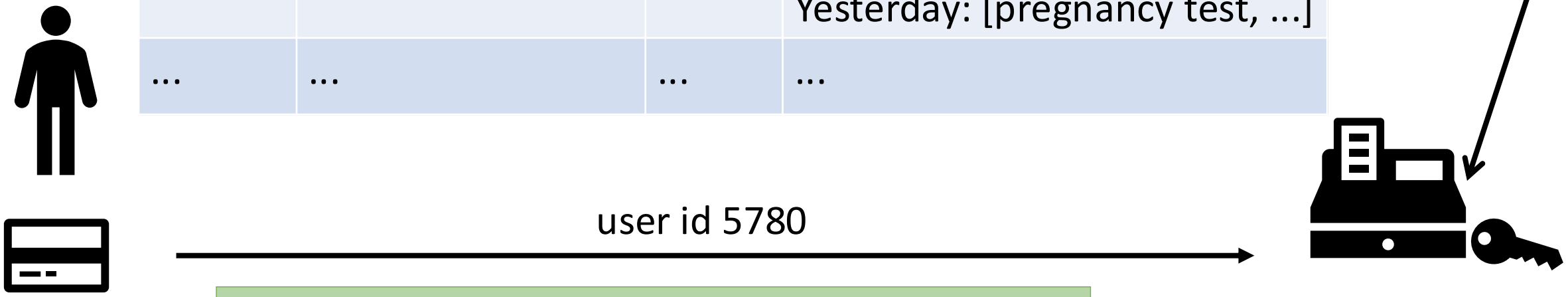
Store

Idea:

- Reward customer loyalty
- Incentivize customers to buy/do specific things (by promise of extra points)

Incentive systems

user id	name	points	Buying habits
…	…	…	…
5780	Jimothy Halpert	264	Today: [teddy bear, …] Yesterday: [pregnancy test, …]
…	…	…	…

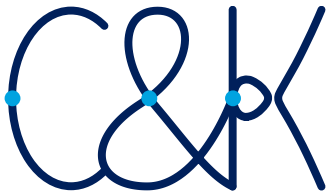


Idea:

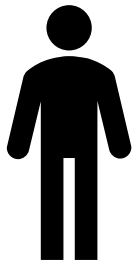
- Reward customer loyalty
- Incentivize customers to buy/do specific things (by promise of extra points)

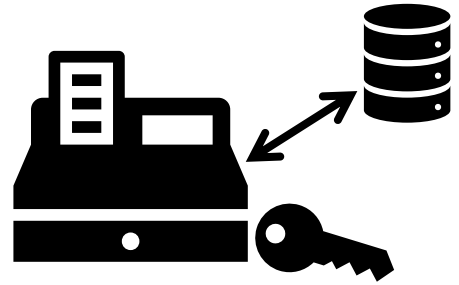
- Collect user data

Privacy-Preserving Incentive System
from Updatable Credentials



Earning k points: Naive version





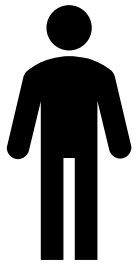
v

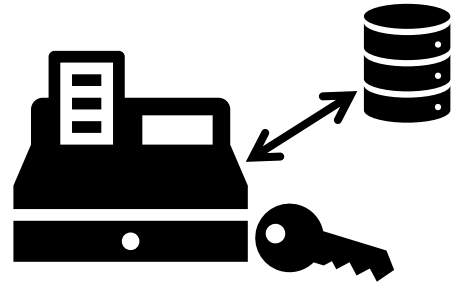
Update

$$\psi(v, \alpha) = v + k$$

$v + k$

Spending k points: Naive version



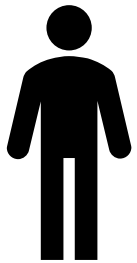


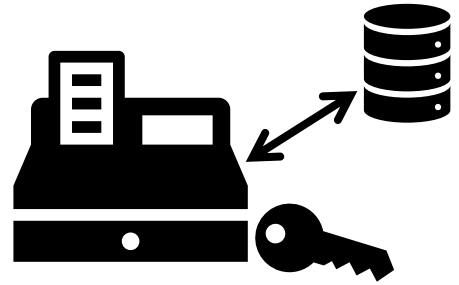
Problem: User can double-spend

Update

$$\psi(v,\alpha) = \begin{cases} v-k & \text{if } v \geq k \\ \perp & \text{otherwise} \end{cases}$$

Earning k points: Double-spending protection





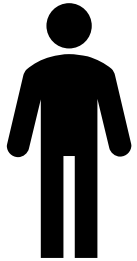
$dsid, v$

Update

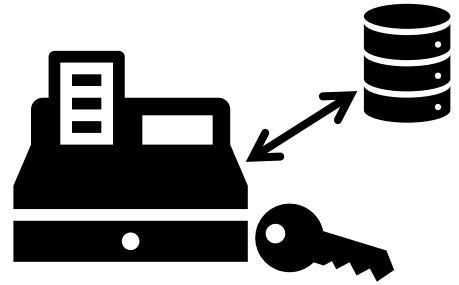
$$\psi\big((dsid, v), \alpha\big) = (dsid, v + k)$$

$dsid, v + k$

Spending k points: Double-spending protection



Problem: Database needs to be *always online* to prevent double-spending



$dsid, v$

$dsid$

Check dsid

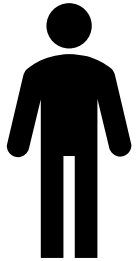
Choose random $dsid^*$

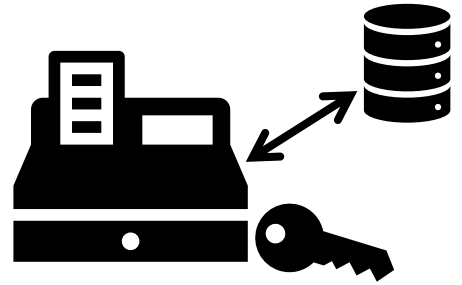
Update

$$\psi\big((dsid, v), dsid^*\big) = \begin{cases} (dsid^*, v-k) & \text{if } v \geq k \wedge \psi_{chk} \\ \perp & \text{otherwise} \end{cases}$$

$dsid^*, v-k$

Spending k points: Offline double-spending protection





$usk, dsid, dsrnd, v$

$dsid$

γ

$usk \cdot \gamma + dsrnd \mod p$

Choose rand. γ

Choose random $dsid^*, dsrnd^*$

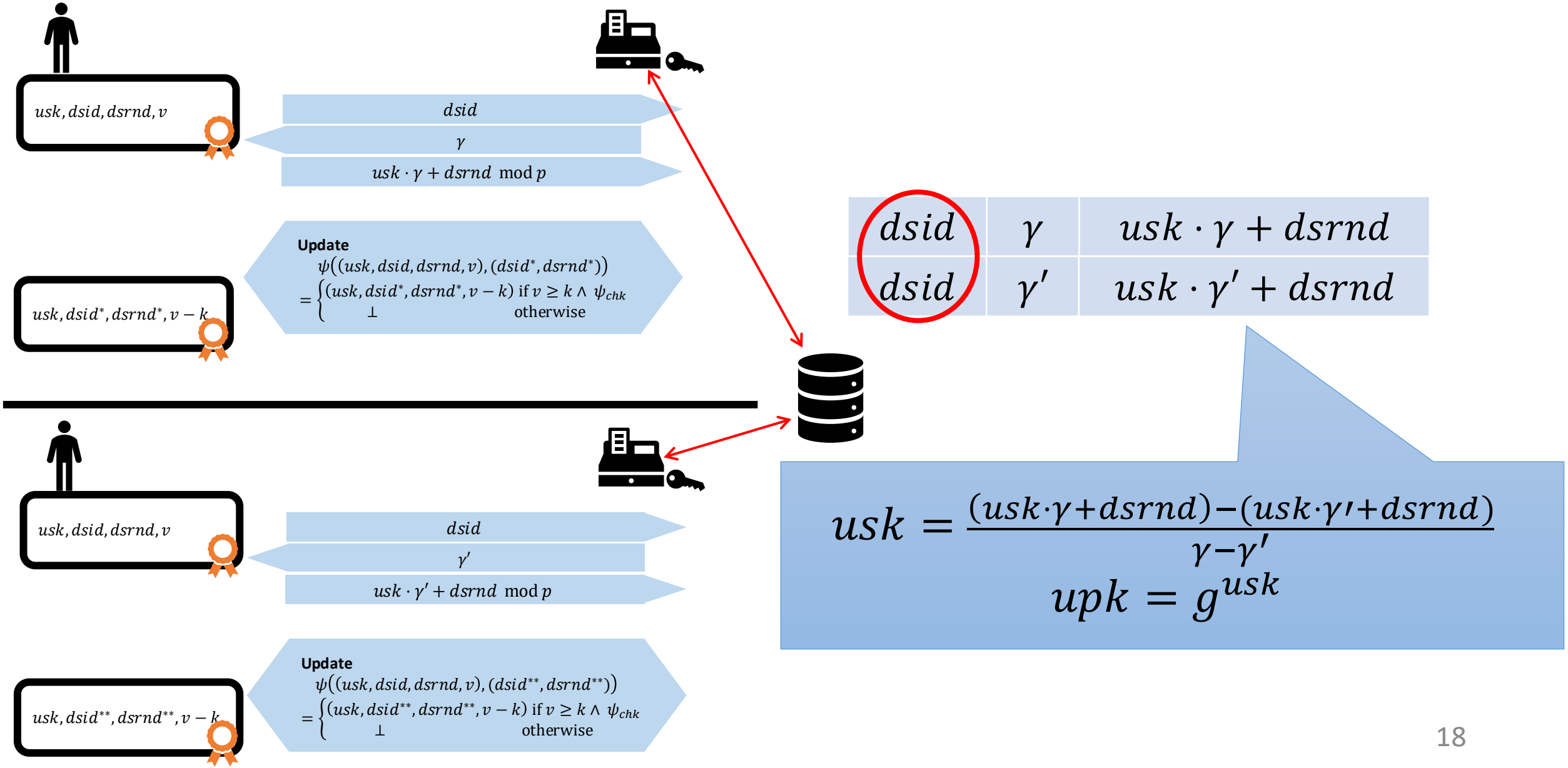
$usk, dsid^*, dsrnd^*, v - k$

Update

$$\psi\big((usk, dsid, dsrnd, v), (dsid^*, dsrnd^*)\big)$$

$$= \begin{cases} (usk, dsid^*, dsrnd^*, v - k) \text{ if } v \geq k \wedge \psi_{chk} \\ \perp \text{ otherwise} \end{cases}$$

Double-spending aftermath

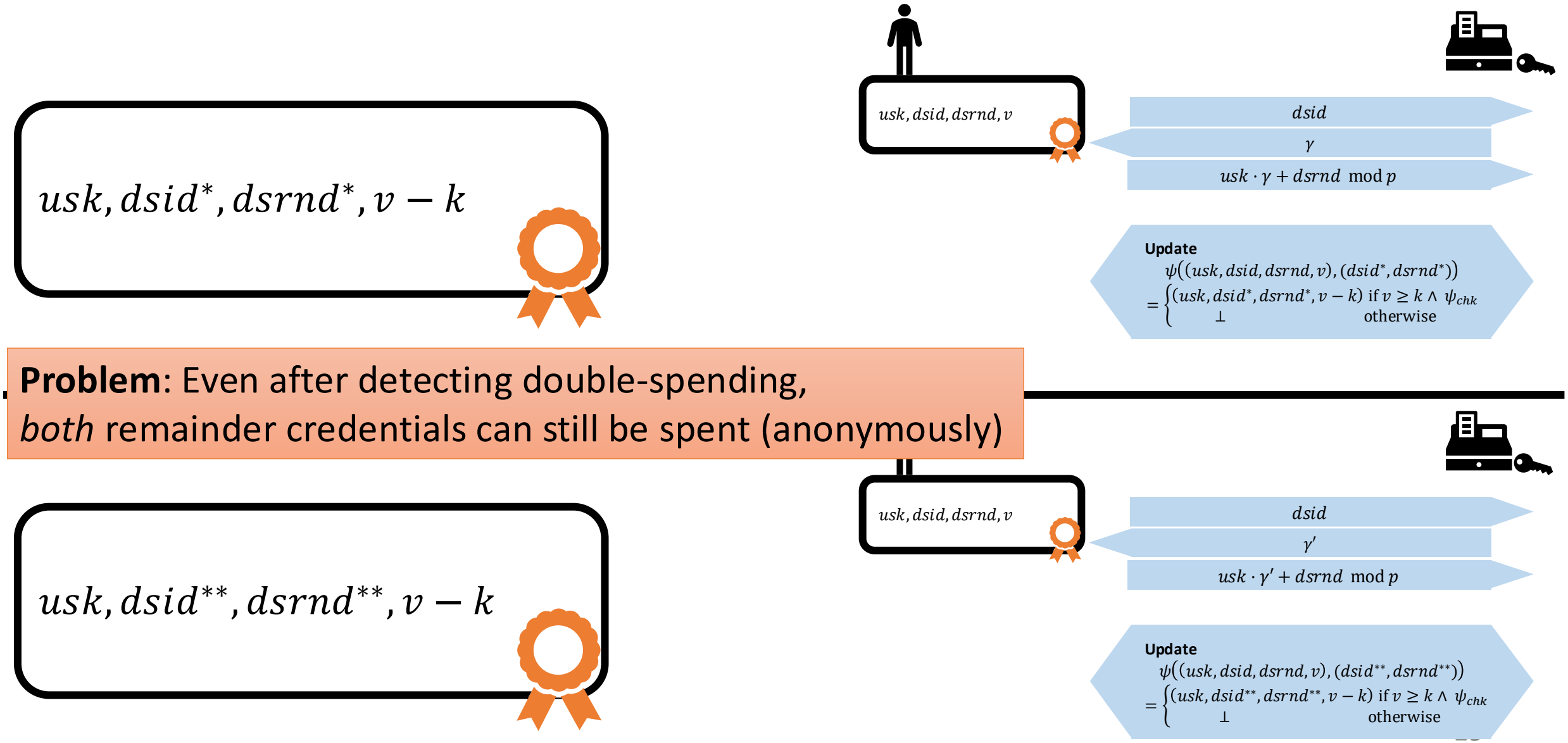


$dsid$	γ	$usk \cdot \gamma + dsrnd$
$dsid$	γ'	$usk \cdot \gamma' + dsrnd$

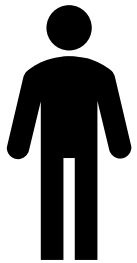
$$usk = \frac{(usk \cdot \gamma + dsrnd) - (usk \cdot \gamma' + dsrnd)}{\gamma - \gamma'}$$

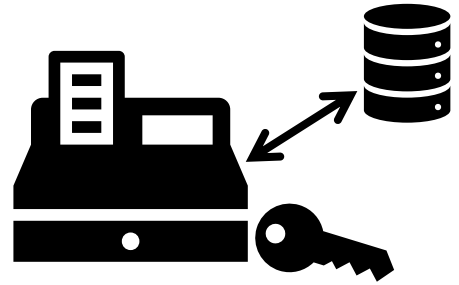
$$upk = g^{usk}$$

Double-spending aftermath



Spending k points: Final (still simplified) version





$usk, dsid, dsrnd, v$

Choose random
$dsid^*, dsrnd^*$

$usk, dsid^*, dsrnd^*, v - k$

$dsid$

γ

Choose rand. γ

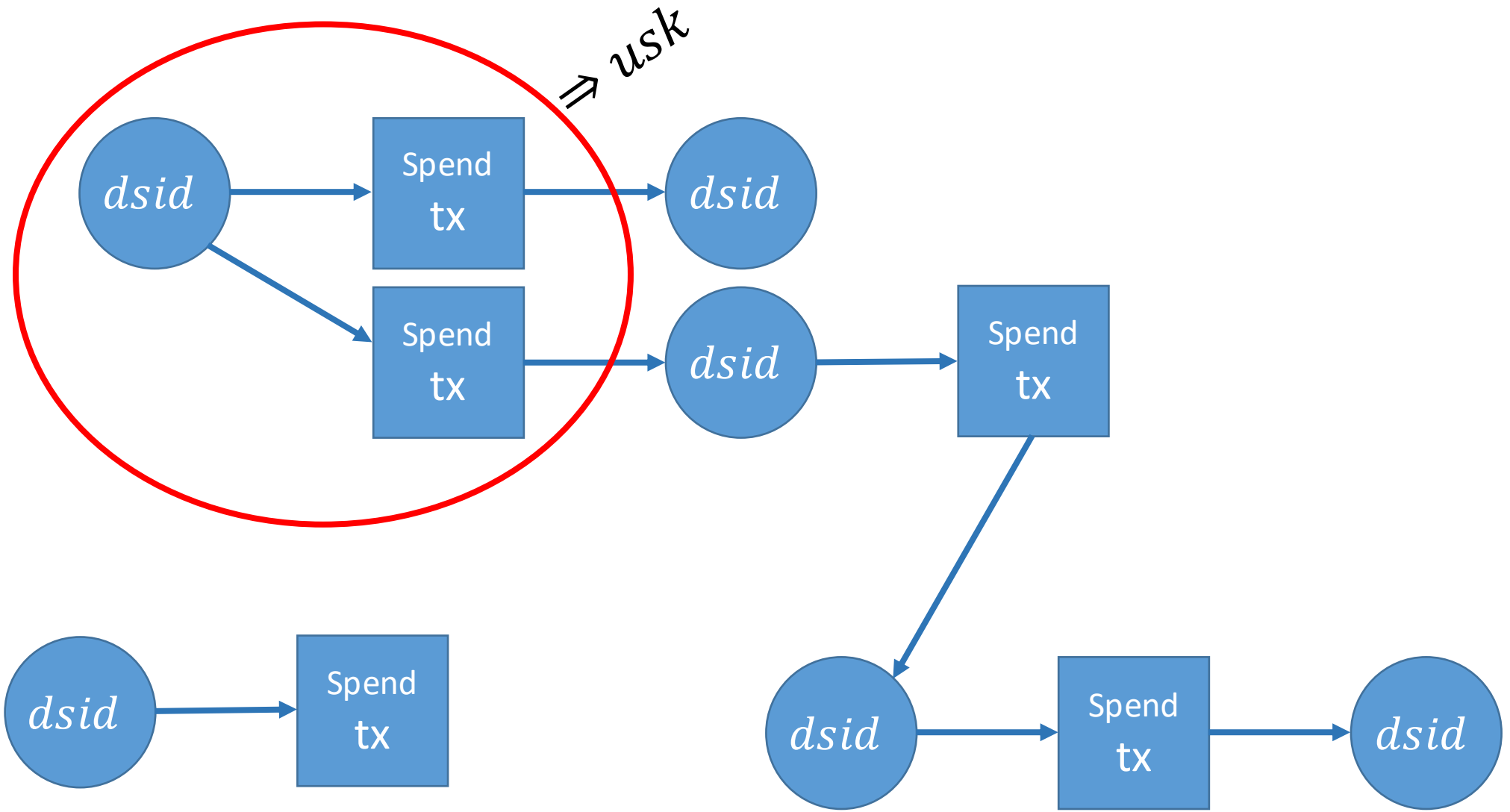
$usk \cdot \gamma + dsrnd \mod p$

$\text{Encrypt}(usk, dsid^*)$

Update

$$\psi\big((usk, dsid, dsrnd, v), (dsid^*, dsrnd^*)\big)$$
$$= \begin{cases} (usk, dsid^*, dsrnd^*, v - k) \text{ if } v \geq k \wedge \psi_{chk} \\ \perp \text{ otherwise} \end{cases}$$

Tracing remainder tokens: The database



Performance

PADERBORN UNIVERSITY

Device	Issue/Join	Credit/Earn	Deduct/Spend
User Google Pixel 1 (Smartphone)	76 ms	110 ms	390 ms
Provider Surface Book 2 i7 (Laptop)	10 ms	17 ms	64 ms
Sum	**86 ms**	**127 ms**	**454 ms**

- **Instantiation** with Pointcheval Sanders signatures, Pedersen, ElGamal & Schnorr
- **Libraries**: `upb.crypto` [1] with `mcl` [2] for bilinear group (BN256)
- **Numbers:** execution time *without* any precomputation

`upb.crypto`: github.com/upbcuk
`mcl`: github.com/herumi/mcl

Conclusion

Updating Anonymous Credentials

- Formal (security) model, proofs
- Credential update mechanism compatible with almost all credential systems

Construction of Incentive System from updatable credentials

- *More efficient* than prior Groth-Sahai-based schemes [HHNR17]
- *Remainder token tracing* missing from all other incentive systems
  - They cannot securely do offline partial spending

[HHNR17] Hartung, Hoffmann, Nagel, Rupp. *BBA+: Improving the security and applicability of privacy-preserving point collection.* CCS 2017