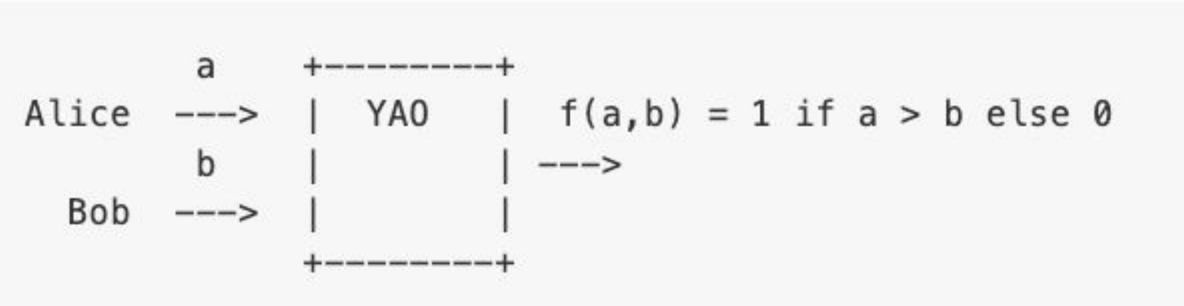
Expanding Blockchain Horizons through Privacy-Preserving Computation

Lorenzo Gentile

PhD thesis
IT University of Copenhagen
2023
Computer Science Department

MPC introduction: Yao’s Millionaires’ problem

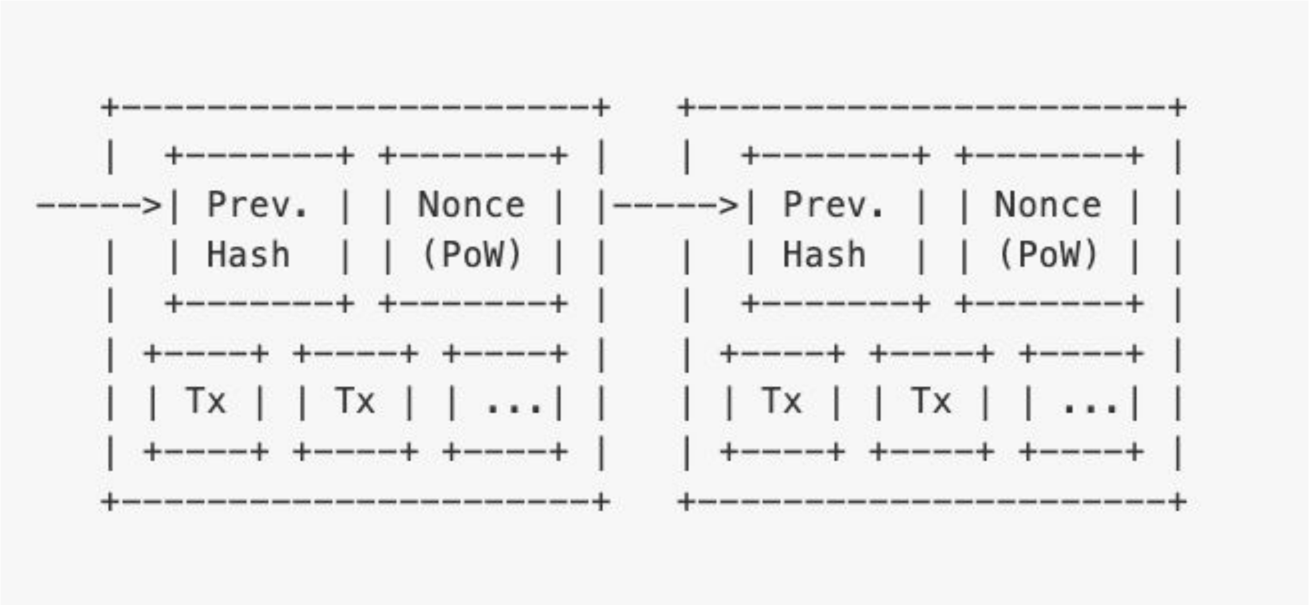
- Introduced in 1982 by computer scientist Andrew Yao: two millionaires, Alice and Bob, are interested in knowing which of them is richer without revealing their actual wealth.



```
        a    +---------+
Alice  --->  |  YAO    |  f(a,b) = 1 if a > b else 0
        b    |         | --->
  Bob  --->  |         |
             +---------+
```

- Compute $f(a, b)$ while preserving the privacy of a and b.
- Theoretical result shows that any function can be evaluated on private inputs.

Blockchain introduction: Bitcoin

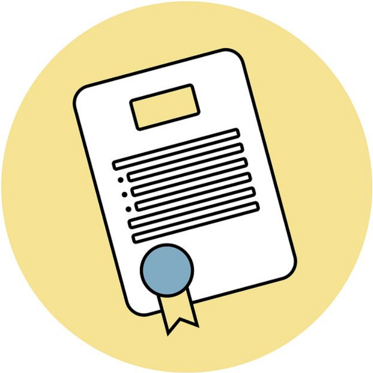


```
   +----------------------+   +----------------------+
   |  +-------+ +-------+ |   |  +-------+ +-------+ |
----->| Prev. | | Nonce | |----->| Prev. | | Nonce | |
   |  | Hash  | | (PoW) | |   |  | Hash  | | (PoW) | |
   |  +-------+ +-------+ |   |  +-------+ +-------+ |
   | +----+ +----+ +----+ |   | +----+ +----+ +----+ |
   | | Tx | | Tx | | ...| |   | | Tx | | Tx | | ...| |
   | +----+ +----+ +----+ |   | +----+ +----+ +----+ |
   +----------------------+   +----------------------+
```

Courtesy of Satoshi Nakamoto (2008)

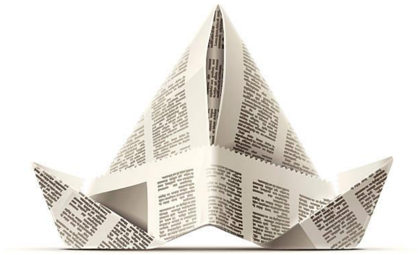
Blockchain introduction: smart contracts

- Smart contracts allow to describe **arbitrarily complex conditions** under which transactions might take place among the parties.
- In the context of this thesis we adopt a **public** blockchain and smart contracts to **automatically enforce** part of the protocols.



Research outputs

- FAST: Fair Auctions via Secret Transactions (ACNS 2022)
- SoK: Mitigation of Front-running in Decentralized Finance (DeFi 2022 - FC 2022 workshop)
- PAPR: Publicly Auditable Privacy Revocation for Anonymous Credentials (CT-RSA 2023)



FAST: Fair Auctions via Secret Transactions

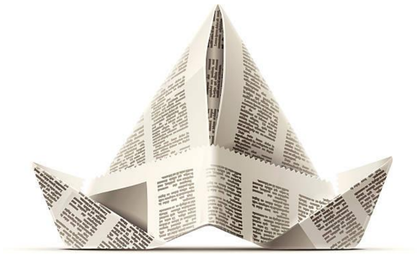
- Efficient **MPC protocols** for both **first and second-price sealed-bid auctions** with **fairness** against rational adversaries, leveraging **secret cryptocurrency transactions** and **public smart contracts**.
- **Cheaters** are identified and **financially punished** by losing a **secret collateral deposit .**
- It is always **more profitable to execute the protocol honestly** than to cheat.

SoK: Mitigation of Front-running in Decentralized Finance

- Front-running is the malicious act of both manipulating the order of pending trades and injecting additional trades to **make a profit at the cost of other users.**
- We describe **common front-running attacks**, propose a **schema of front-running mitigation categories**, assess the **state-of-the-art techniques** in each category and illustrate **remaining attacks.**

PAPR: Publicly Auditable Privacy Revocation for Anonymous Credentials

- We introduce the notion of a**nonymous credentials with Publicly Auditable Privacy Revocation (PAPR)**.
- Formalize it as an **ideal functionality** and propose a **realization** that is secure under **standard assumptions in the Universal Composability (UC) framework** against **static adversaries**.
- We show how to modify our construction to make it secure against **mobile adversaries**.



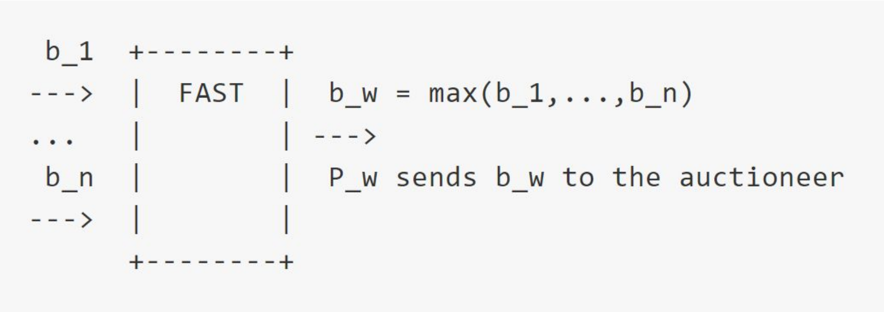
FAST: Fair Auctions via Secret Transactions

ACNS 2022

Bernardo David, IT University of Copenhagen
Lorenzo Gentile, IT University of Copenhagen
Mohsen Pourpouneh, University of Copenhagen

FAST protocol

- Parties $\mathscr{P}_i$ with $i \in 1, \ldots, n$.
- Bid $b_i = b_{i1} | \ldots | b_{il}$ with $b_{ir} \in \{0, 1\}$.



```
 b_1  +--------+
--->  |  FAST  |  b_w = max(b_1,...,b_n)
...   |        | --->
 b_n  |        |  P_w sends b_w to the auctioneer
--->  |        |
      +--------+
```

- Compute $max(b_1, \ldots, b_n)$ while preserving the privacy of $b_1, \ldots, b_n$ (similarly for second price).

Motivation

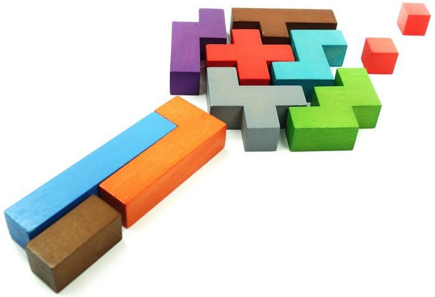
- It may be **not feasible** or **expensive** to find a trusted third party.
- A third party may cheat, without being detected, to **increase profit** (e.g., increase second price).

FAST in a nutshell

- Parties send **secret deposits** to a **smart contract**.
- Cheating parties **lost their deposits**.
- **Rational parties do not cheat**.
- **Fairness** is achieved.

Building blocks

- Secret deposits.
- Anonymous veto protocol.
- Non interactive zero knowledge proofs (NIZKs).
- Cheating detection.
- Recovery committee.



Secret deposits (novel technique)

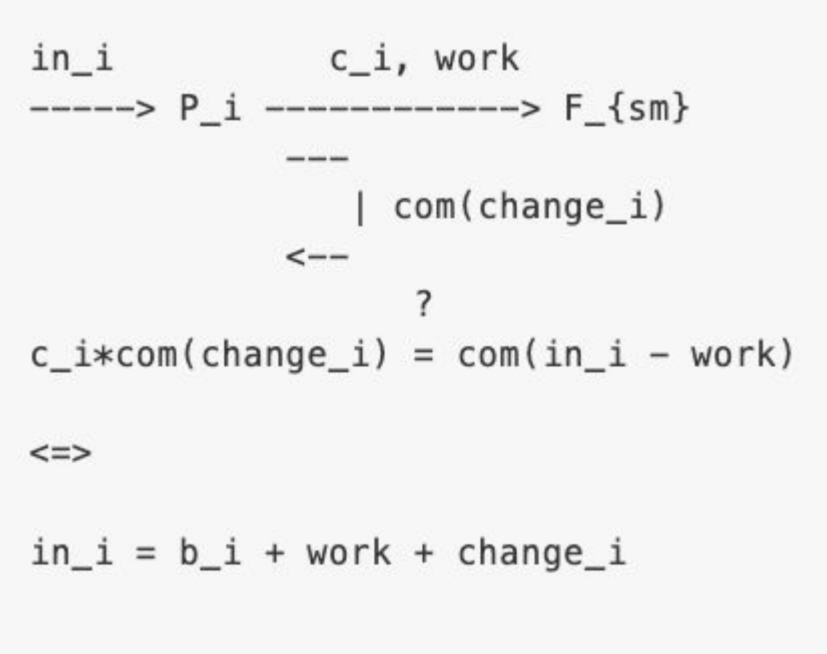
- In order to make rational parties do not cheat, **the deposits have to be equal to the bids plus work**.
- However, the **privacy of the bids has to be preserved**.
- Secret deposits are adopted (e.g., using **confidential transactions** by Greg Maxwell).

Confidential transactions (details)

- Parties $\mathscr{P}_i$ with $i \in 1, \ldots, n$.
- Bid $b_i = b_{i1} | \ldots | b_{il}$ with $b_{ir} \in \{0, 1\}$.
- $\mathscr{P}_i$ computes the bit commitments as $c_{ir} = g^{b_{ir}} h^{r_{ir}}$ to each bit b_{ir} of b_i (used in NIZKs later), and the bid commitment as:

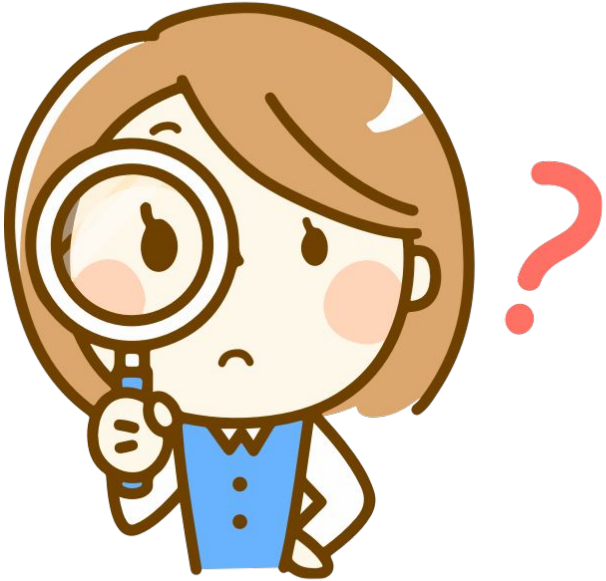
$$c_i = \prod_{r=1}^{l} c_{ir}^{2^{l-r}} = g^{b_i} h^{\sum_{r=1}^{l} 2^{l-r} r_{ir}}$$

- $\mathscr{P}_i$ send a confidential transaction to the smart contract:



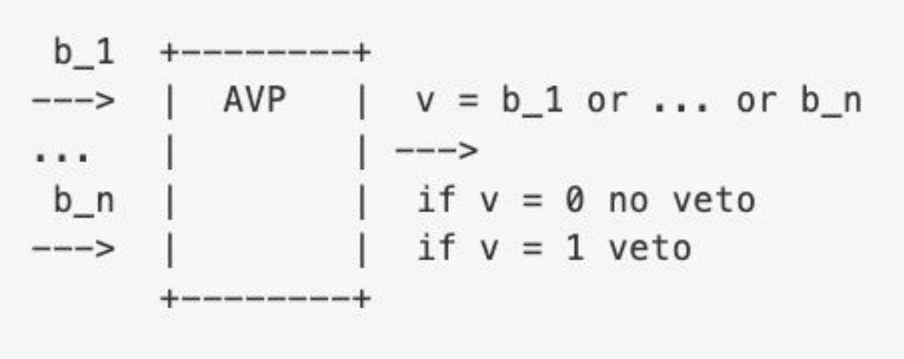
```
in_i          c_i, work
-----> P_i ------------> F_{sm}
            ---
               | com(change_i)
            <--
                  ?
c_i*com(change_i) = com(in_i - work)

<=>

in_i = b_i + work + change_i
```

- The smart contract verifies the validity of the confidential transaction (**inputs equal to outputs** and **range proofs**).
- $\mathscr{P}_i$ verifies for each other party $\mathscr{P}_j$ that $c_j = \prod_{k=1}^{l} c_{jk}^{2^{l-k}}$ for $j \in \{1, \ldots, n\} \setminus i$.



Anonymous veto protocol

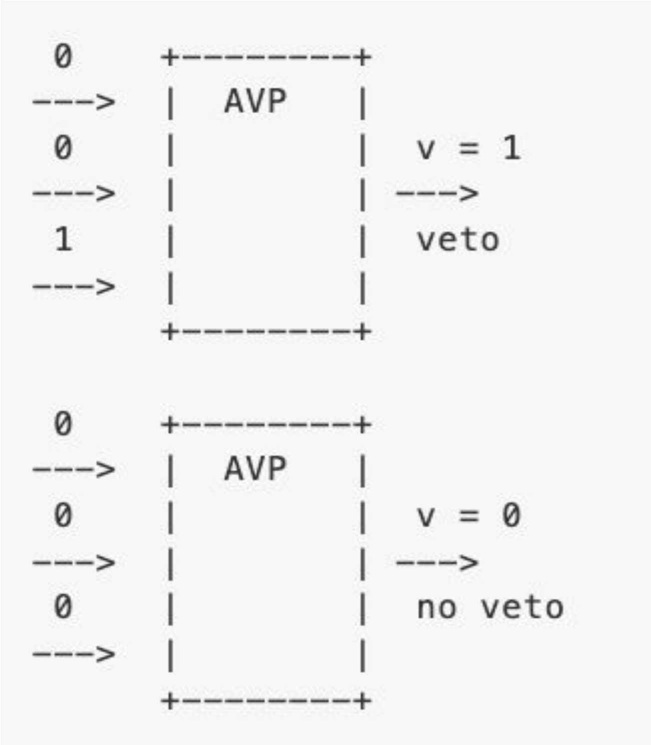
- Parties $\mathscr{P}_i$ with $i \in 1, \ldots, n$.
- Bit $b_i \in \{0, 1\}$.



```
 b_1  +--------+
--->  |  AVP   |  v = b_1 or ... or b_n
...   |        | --->
 b_n  |        |  if v = 0 no veto
--->  |        |  if v = 1 veto
      +--------+
```

- Compute $b_1 \vee \ldots \vee b_n$ while preserving the privacy of $b_1, \ldots, b_n$.

Anonymous veto protocol (examples)



```
 0    +---------+
--->  |  AVP    |
 0    |         |  v = 1
--->  |         | --->
 1    |         |  veto
--->  |         |
      +---------+

 0    +---------+
--->  |  AVP    |
 0    |         |  v = 0
--->  |         | --->
 0    |         |  no veto
--->  |         |
      +---------+
```

Anonymous veto protocol (details)

- **Round 1.** $\mathscr{P}_i$ chooses $x_i \xleftarrow{u} \mathbb{Z}_q$ (uniformly at random), computes $X_i = g^{x_i}$ and broadcasts X_i .
- **Round 2.** Upon receiving X_j from all other parties $\mathscr{P}_j$, $\mathscr{P}_i$ computes

$$Y_i = \prod_{k=1}^{i-1} X_k / \prod_{k=i+1}^{n} X_k = g^{(\sum_{k=1}^{i-1} x_k - \sum_{k=i+1}^{n} x_k)}$$

and then broadcasts the following message:

$$v_i = \begin{cases} Y_i^{x_i}, & \text{if } b_i = 0 \\ r \xleftarrow{u} \mathbb{Z}_q, g^r, & \text{if } b_i = 1 \end{cases}$$

- **Output.** All parties compute $V=\prod_{i=1}^{n} v_i$ after receiving all the v_i‘ s from the other parties. Note that:

$$V = 1 \Leftrightarrow b_i=0 \ \forall\, i \in \{1, \ldots, n\}$$

i.e., $V = 1$ if and only if there is no veto.

Anonymous veto protocol (detailed example)

$n = 3$

$X_1 = g^{x_1}, X_2 = g^{x_2}, X_3 = g^{x_3}$

$Y_1 = g^{-x_2 - x_3}, Y_2 = g^{x_1 - x_3}, Y_3 = g^{x_1 + x_2}$

if we assume $b_i = 0 \; \forall \; i \in \{1, 2, 3\}$, then:

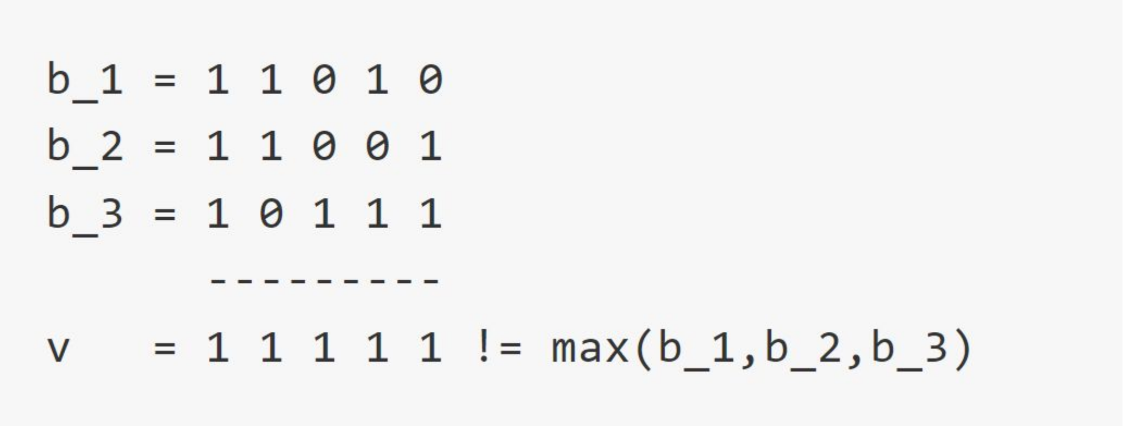
$V = v_1 \cdot v_2 \cdot v_3 = Y_1^{x_1} \cdot Y_2^{x_2} \cdot Y_3^{x_3}$

$= g^{-x_1(x_2 + x_3)} \cdot g^{x_2(x_1 - x_3)} \cdot g^{x_3(x_1 + x_2)}$

$= g^0 = 1 \Rightarrow$ no veto

Anonymous first price auction protocol

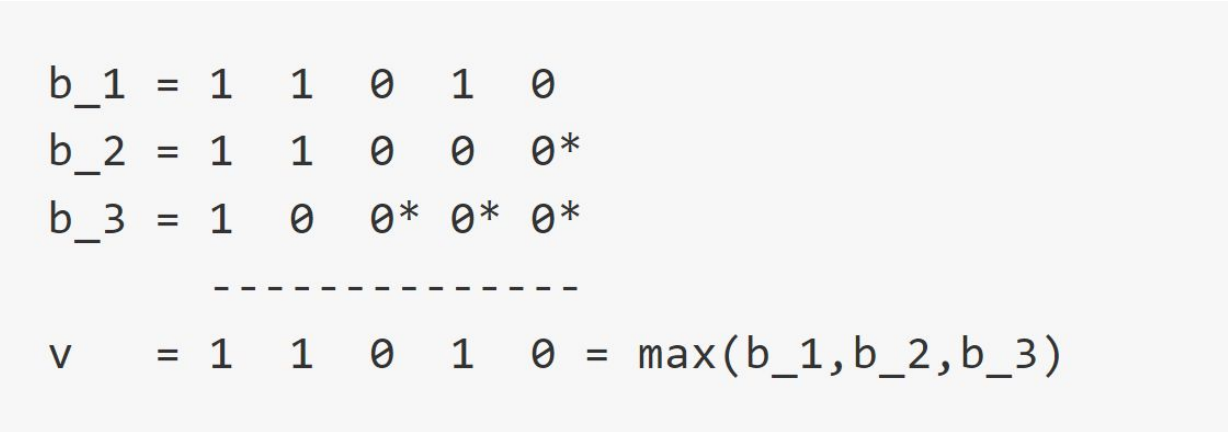
- (idea) Use bit-by-bit AVP.



```
b_1 = 1 1 0 1 0
b_2 = 1 1 0 0 1
b_3 = 1 0 1 1 1
      ---------
v   = 1 1 1 1 1 != max(b_1,b_2,b_3)
```

Anonymous first price auction protocol

- (idea) Modify input bits according to previous inputs and outputs.



```
b_1 = 1  1  0  1  0
b_2 = 1  1  0  0  0*
b_3 = 1  0  0* 0* 0*
      --------------
v   = 1  1  0  1  0 = max(b_1,b_2,b_3)
```

- if $v_r = 1$ but $b_{ir} = 0$ then $d_{ik} = 0$ for $k = r + 1, \ldots, l$, where d_{ik} stands for declared bit.

NIZK proofs

- How can we guarantee that the rule “if $v_r = 1$ but $b_{ir}=0$ then $d_{ik}=0$ for $k = r+1, \ldots, l$" is followed by the parties?
- Non interactive zero knowledge proofs guarantee that d_{ir} are correctly computed according to the inputs and outputs of the previous rounds.

NIZK proofs - Before First Veto (details)

$$v_{ir} = \begin{cases} Y_{ir}^{x_{ir}}, & \text{if } b_{ir} = 0 \\ g^{\bar{r}_{ir}}, & \text{if } b_{ir} = 1 \end{cases}$$

$$BV_{ir} \leftarrow BV\{b_{ir}, r_{ir}, x_{ir}, \bar{r}_{ir} \mid$$

$$\left(\frac{c_{ir}}{g^{b_{ir}}} = c_{ir} = h^{r_{ir}} \wedge v_{ir} = Y_{ir}^{x_{ir}} \wedge X_{ir} = g^{x_{ir}}\right) \vee$$

$$\left(\frac{c_{ir}}{g^{b_{ir}}} = \frac{c_{ir}}{g} = h^{r_{ir}} \wedge v_{ir} = g^{\bar{r}_{ir}}\right) \}$$

Logical condition to prove:

$$(b_{ir} = 0 \wedge d_{ir} = 0) \vee (b_{ir} = 1 \wedge d_{ir} = 1)$$

NIZK proofs - After First Veto (details)

$$v_{ir} = \begin{cases} Y_{ir}^{x_{ir}}, & \text{if } b_{ir} = 0 \\ g^{r_{ir}}, & \text{if } d_{i\hat{r}} = 1 \wedge b_{ir} = 1 \\ Y_{ir}^{x_{ir}}, & \text{if } d_{i\hat{r}} = 0 \wedge b_{ir} = 1 \end{cases}$$

$$AV_{ir} \leftarrow AV\{b_{ir}, r_{ir}, x_{ir}, \bar{r}_{i\hat{r}}, \bar{r}_{ir}, x_{i\hat{r}} \,|$$

$$\left(\frac{c_{ir}}{g^{b_{ir}}} = c_{ir} = h^{r_{ir}} \wedge v_{ir} = Y_{ir}^{x_{ir}} \wedge X_{ir} = g^{x_{ir}}\right) \vee$$

$$\left(\frac{c_{ir}}{g^{b_{ir}}} = \frac{c_{ir}}{g} = h^{r_{ir}} \wedge d_{i\hat{r}} = g^{\bar{r}_{i\hat{r}}} \wedge v_{ir} = g^{\bar{r}_{ir}}\right) \vee$$

$$\left(\frac{c_{ir}}{g^{b_{ir}}} = \frac{c_{ir}}{g} = h^{r_{ir}} \wedge \ d_{i\hat{r}} = Y_{i\hat{r}}^{x_{i\hat{r}}} \wedge X_{i\hat{r}} = g^{x_{i\hat{r}}}\right.$$

$$\left. \wedge\, v_{ir} = Y_{ir}^{x_{ir}} \wedge X_{ir} = g^{x_{ir}}\right) \}$$

Logical condition to prove:

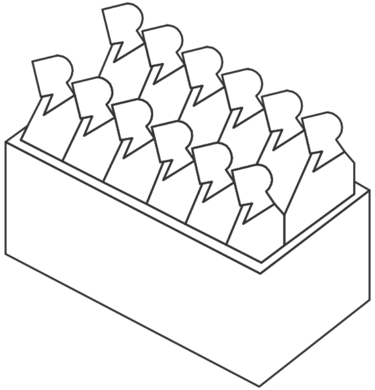
$$(b_{ir} = 0 \wedge d_{ir} = 0) \vee (b_{ir} = 1 \wedge d_{i\hat{r}} = 1 \wedge d_{ir} = 1) \vee (b_{ir} = 1 \wedge d_{i\hat{r}} = 0 \wedge d_{ir} = 0)$$

Cheating detection

- How can we detect cheating parties?
  - NIZK are publicly verifiable.
  - Signed messages allow to prove inconsistencies.
- If cheating is detected, a **recovery stage** is executed.

Recovery committee

- The opening of the confidential transaction ($c_i = g^{b_i} h^{\sum_{r=1}^{l} 2^{l-r} r_{ir}}$) committed amount is **secret shared** with a committee using **PVSS**.
- In the recovery stage the opening is reconstructed and the **confidential transaction is spent**.



Extension to second price auction

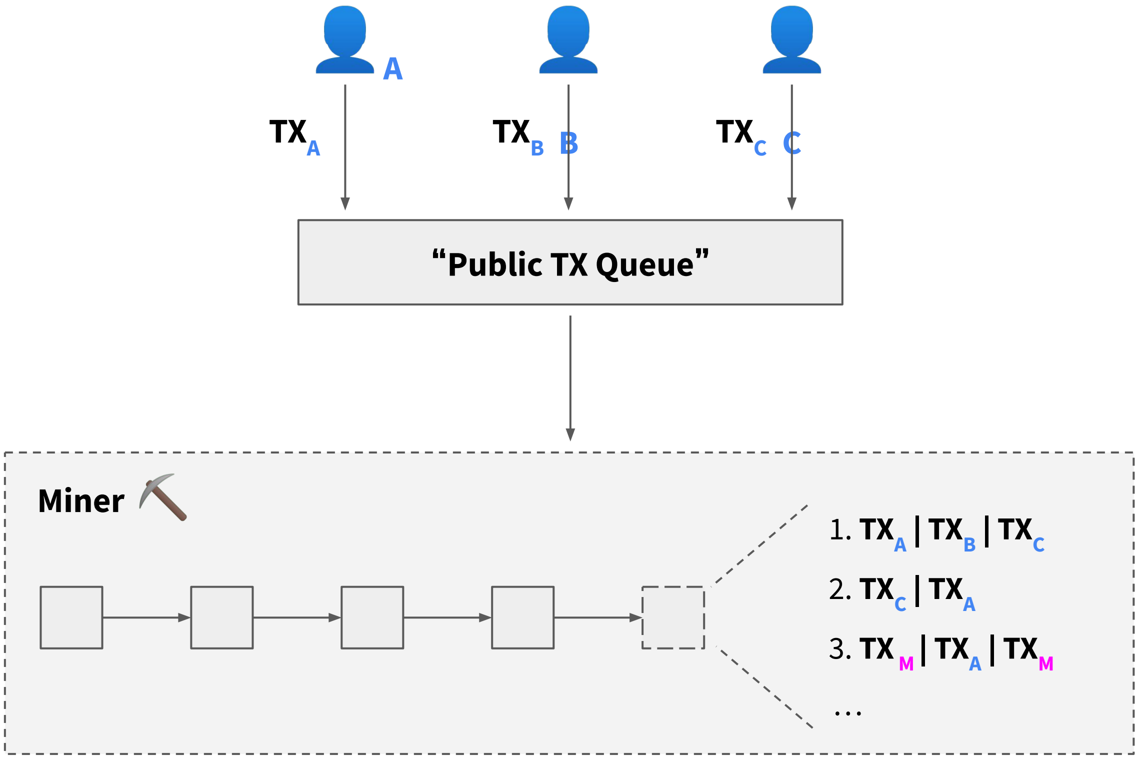
- (idea) Execute again the protocol without the winning party.
- (better idea) Once the winning party $\mathscr{P}_w$ is identified, conclude the execution to compute the second price without $\mathscr{P}_w$.
- From a game theory perspective, bidding truthfully is a **dominant strategy**.

SoK: Mitigation of Front-running in Decentralized Finance

DeFi 2022 - FC 2022 workshop

Carsten Baum, Technical University of Denmark
James Hsin-yu Chiang, Technical University of Denmark
Bernardo David, IT University of Copenhagen
Tore Kasper Frederiksen, Protocol Labs
Lorenzo Gentile, IT University of Copenhagen

Blockchain Interaction



User submits TX's to the network

(Does not participate in mining)

TX are gossiped the across network

Miner appends any valid TX sequence

(Rational miner will optimize for profit)

Front-running **Adversary**

Miner has the power to:

1. **Infer user intentions from …**

    the pending TX queue

    the blockchain state

Compute optimal strategy

(Causalities: Pending TX and State)

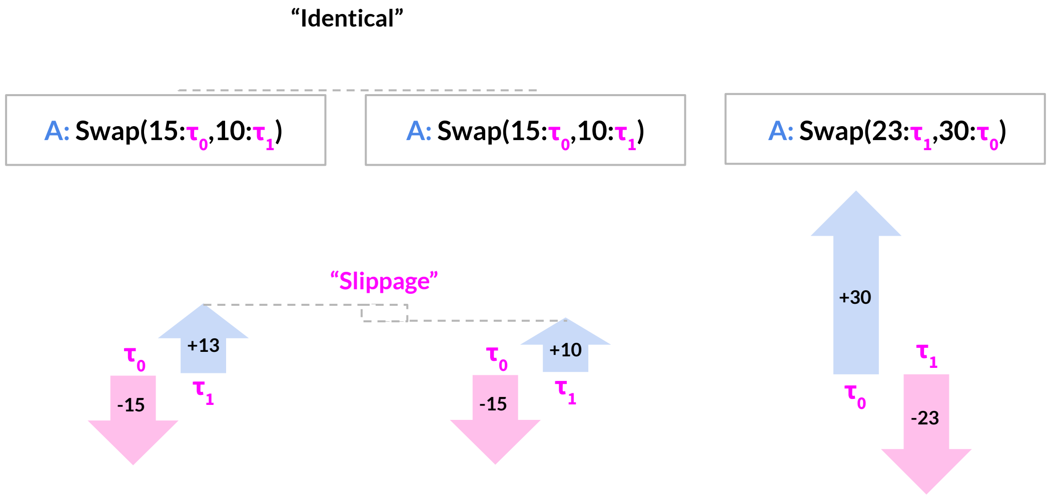
2. **Append TX sequence to the blockchain constructed from …**

    the pending TX queue

    its own TXs

Execute optimal strategy

AMM Slippage



“Identical”

A: Swap(15:τ_0,10:τ_1)

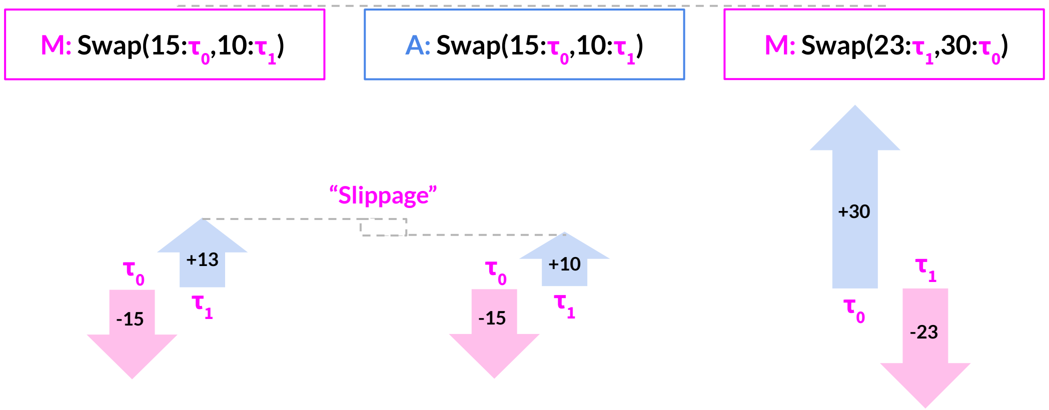
A: Swap(15:τ_0,10:τ_1)

A: Swap(23:τ_1,30:τ_0)

“Slippage”

AMM Sandwich Attack

“Sandwich attack” by M



M: Swap(15:τ_0,10:τ_1)

A: Swap(15:τ_0,10:τ_1)

M: Swap(23:τ_1,30:τ_0)

“Slippage”

τ_0 -15 +13 τ_1

τ_0 -15 +10 τ_1

+30 τ_0 τ_1 -23

ΔW_A = −15+10 = -5

ΔW_M = −15 +13 -23 +30 = +5

Front-running **is a Problem**

1. **Honest users incur a financial loss**

    Sandwich attacks

    Stolen Strategies (Arbitrage/Liquidation)

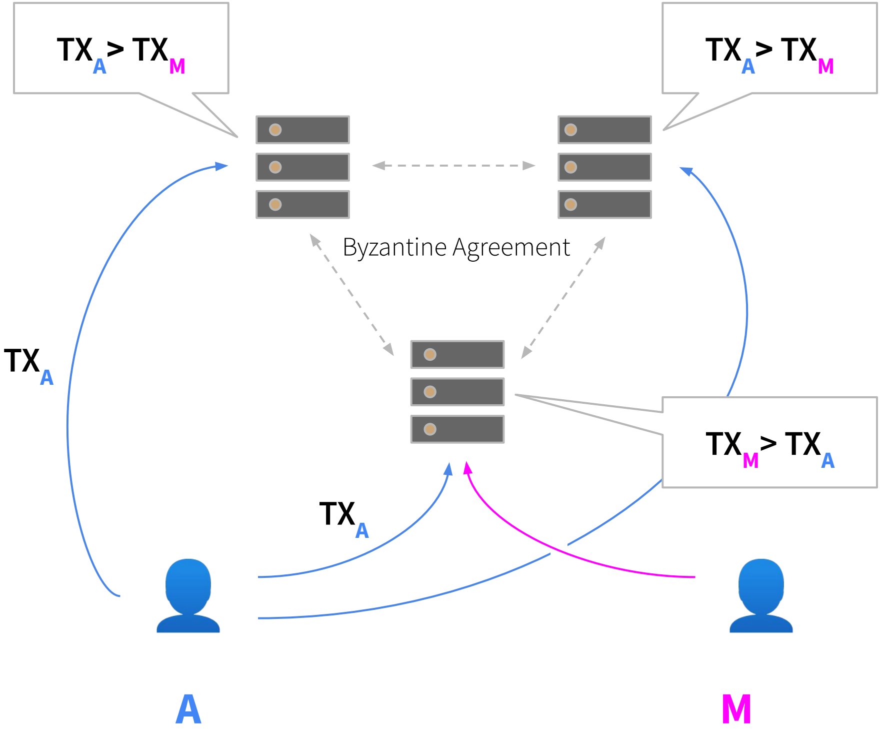
2. **Generates unnecessary demand for block-space**

    Network Congestion from front-running TXs

Front-running **Mitigation**

Miner powers	Mitigation	Proposed Techniques
Action sequencing	Fair Ordering	Fair Ordering Consensus
Inference of user intent	Batching of blinded inputs	(Hash Commitments) Time-lock Crypto Threshold Crypto
	Private balances & secret state + batching of blinded inputs	Secure Multi-Party Computation

Fair Ordering Consensus

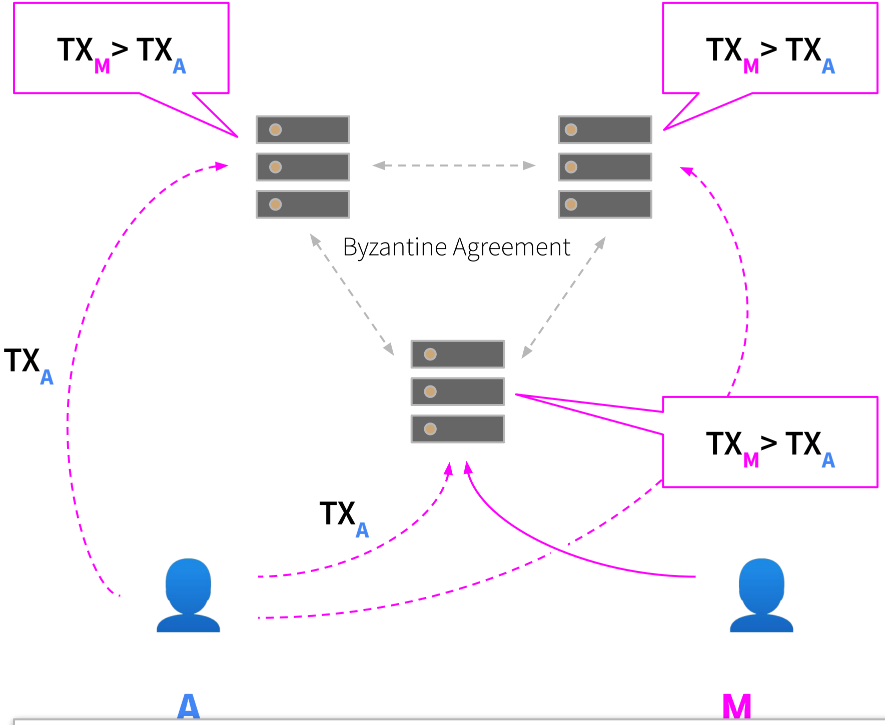


Fair-ordering BA consensus
[Wendy, KDK21, KDL+21, CSMZ21]

γ-receipt-order-fairness [KDK21, KDL+21]

TX_A will be finalized prior to TX_M if
TX_A is observed prior to TX_M by a γ-fraction of nodes

Fair Ordering Consensus



Fair-ordering BA consensus
[Wendy, KDK21, KDL$^+$21, CSMZ21]

γ-receipt-order-fairness [KDK21, KDL$^+$21]

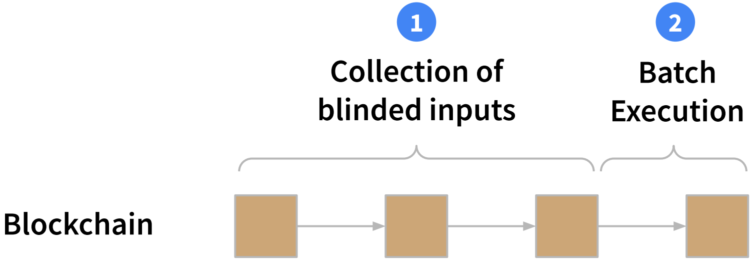
TX_A will be finalized prior to TX_M if
TX_A is observed prior to TX_M by a <u>γ-fraction</u> of nodes

Open challenges: P2P networks / Incentive compatibility

Front-running **Mitigation**

Miner powers	Mitigation	Proposed Techniques
Action sequencing	Fair Ordering	Fair Ordering Consensus
	Batching of blinded inputs	(Hash Commitments) Time-lock Crypto Threshold Crypto
Inference of user intent	Private balances & secret state + batching of blinded inputs	Secure Multi-Party Computation

Batching of Blinded Inputs

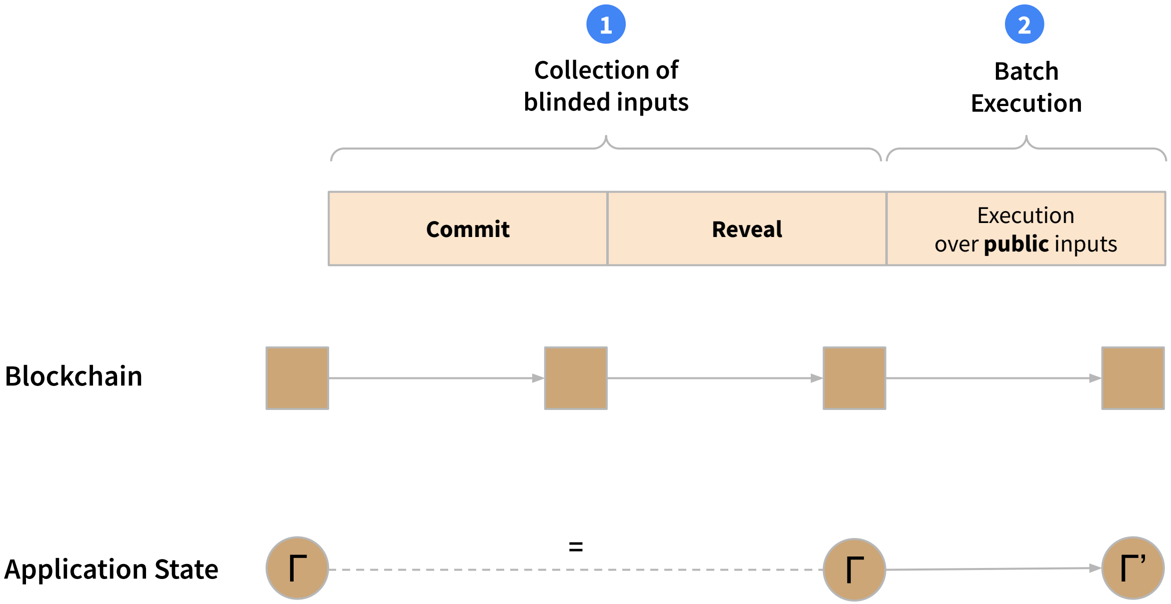


1. Inference of user intent
2. Action sequencing

❶ Inputs are blinded

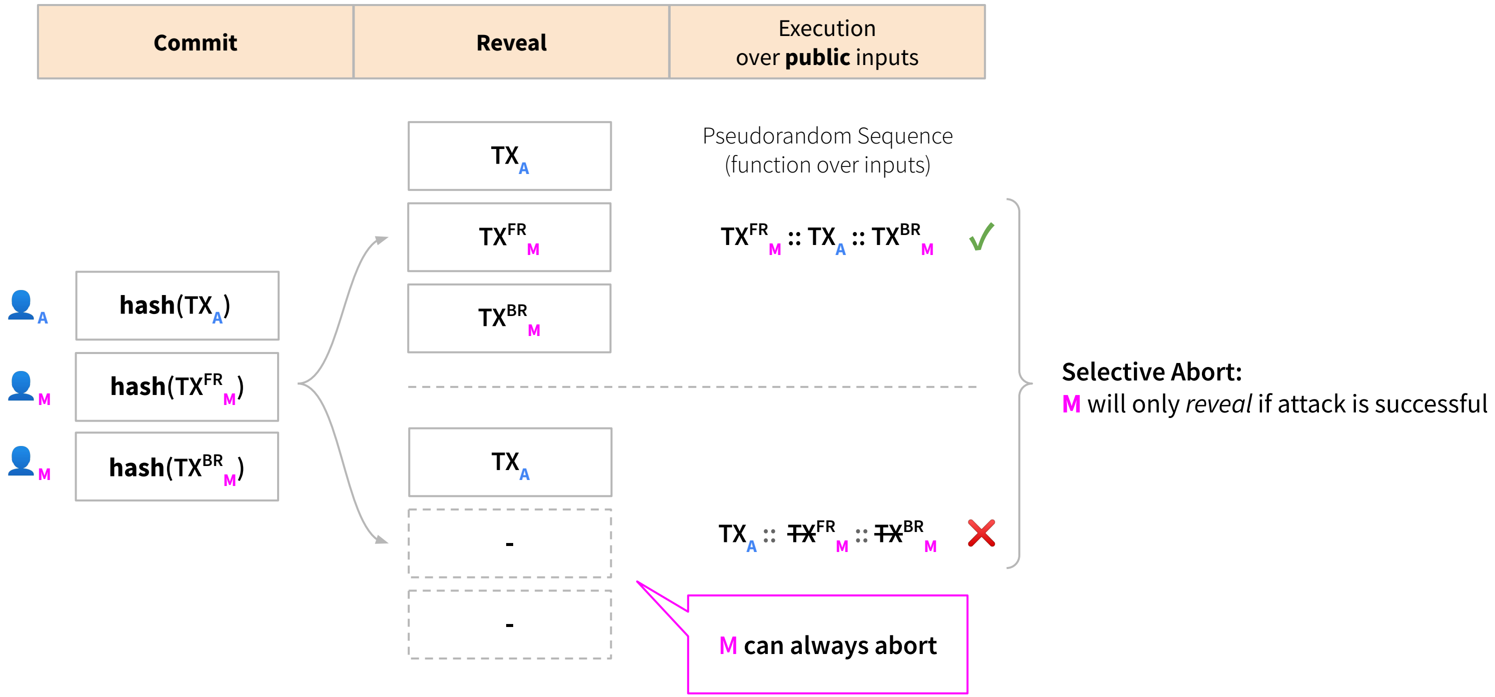
❷ Pseudorandom shuffling / (Input aggregation)

Batching of Blinded Inputs



Hash Commitments (with Selective Abort)
Threshold Encryption
Time-lock Crypto

Order Batching: **Hash Commitments**



Commit

Reveal

Execution over **public** inputs

A **hash**(TX_A)

M **hash**(TX^{FR}_M)

M **hash**(TX^{BR}_M)

TX_A

TX^{FR}_M

TX^{BR}_M

TX_A

-

-

Pseudorandom Sequence (function over inputs)

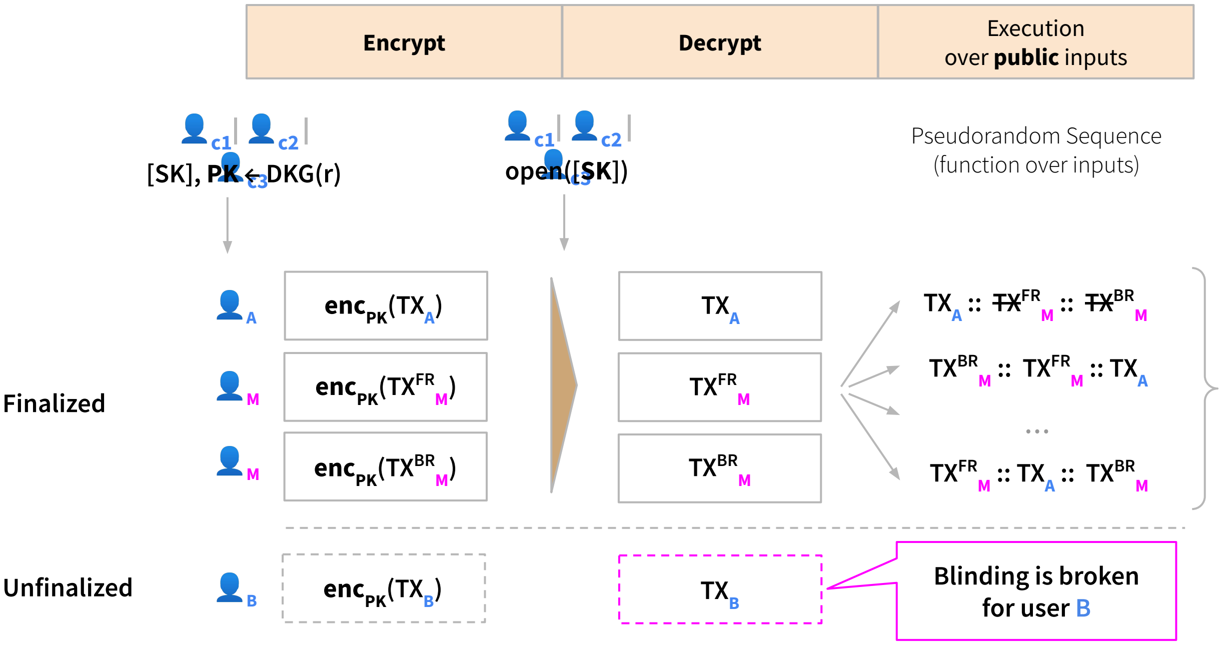
TX^{FR}_M :: TX_A :: TX^{BR}_M ✓

TX_A :: ~~TX~~$^{FR}_M$ :: ~~TX~~$^{BR}_M$ ✗

Selective Abort:
M will only *reveal* if attack is successful

M can always abort

Order Batching: **Threshold Encryption**



Encrypt

Decrypt

Execution
over **public** inputs

c_1 c_2 c_3

[SK], **PK** ← DKG(r)

c_1 c_2 c_3

open([**SK**])

Pseudorandom Sequence
(function over inputs)

Finalized

A $\text{enc}_{PK}(TX_A)$ → TX_A

M $\text{enc}_{PK}(TX^{FR}_M)$ → TX^{FR}_M

M $\text{enc}_{PK}(TX^{BR}_M)$ → TX^{BR}_M

$TX_A :: \cancel{TX}^{FR}_M :: \cancel{TX}^{BR}_M$

$TX^{BR}_M :: TX^{FR}_M :: TX_A$

...

$TX^{FR}_M :: TX_A :: TX^{BR}_M$

Unfinalized

B $\text{enc}_{PK}(TX_B)$ → TX_B

Blinding is broken for user B

Threshold Crypto System

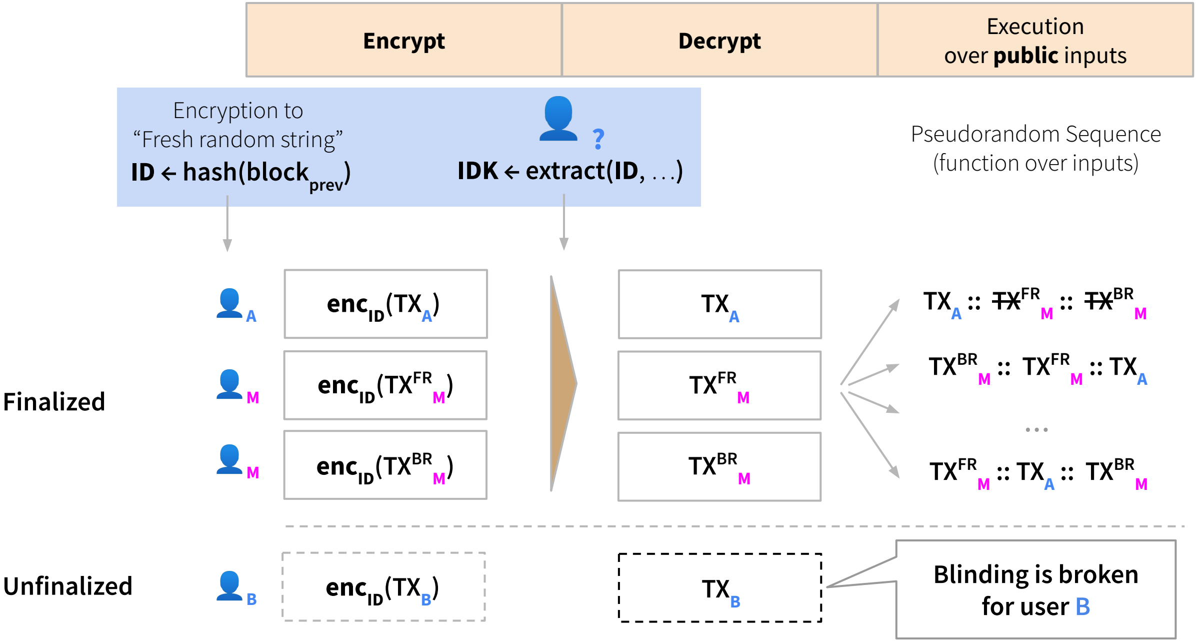
- Shutter Network

No abort possible

(Honest majority in DKG committee)

Additional honesty threshold assumption

Order Batching: **Delay Encryption**



Encrypt | **Decrypt** | Execution over **public** inputs

Encryption to “Fresh random string”

ID ← hash(block$_{prev}$)

?

IDK ← extract(ID, . . .)

Pseudorandom Sequence (function over inputs)

Finalized

A | **enc$_{ID}$(TX_A)** | TX_A

M | **enc$_{ID}$(TX^{FR}_M)** | TX^{FR}_M

M | **enc$_{ID}$(TX^{BR}_M)** | TX^{BR}_M

$TX_A :: \cancel{TX}^{FR}_M :: \cancel{TX}^{BR}_M$

$TX^{BR}_M :: TX^{FR}_M :: TX_A$

…

$TX^{FR}_M :: TX_A :: TX^{BR}_M$

Unfinalized

B | **enc$_{ID}$(TX_B)** | TX_B

Blinding is broken for user B

Delay Encryption [DeFeo, Burdges]

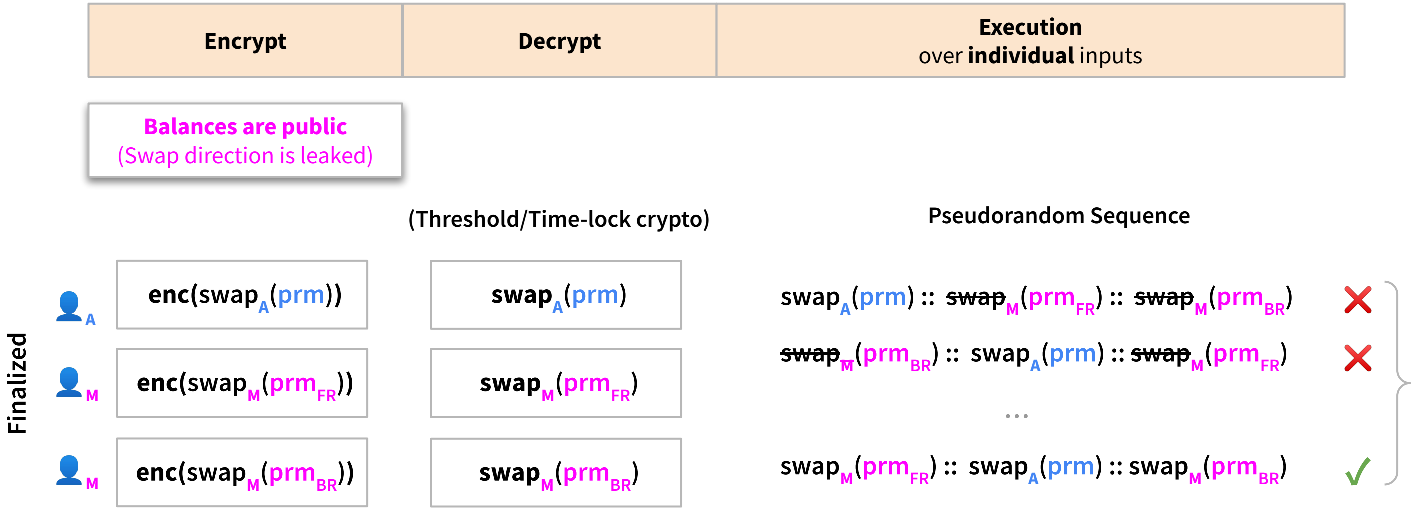
- Single extraction for all inputs

Alternatively: Time-lock Puzzles

- One extraction per input [RSW]

Open challenge: Delay-parameterization

However: **Batching is not enough**



Encrypt	Decrypt	Execution over **individual** inputs

Balances are public
(Swap direction is leaked)

(Threshold/Time-lock crypto)

Pseudorandom Sequence

Finalized

	Encrypt	Decrypt
👤$_A$	enc(swap$_A$(prm))	swap$_A$(prm)
👤$_M$	enc(swap$_M$(prm$_{FR}$))	swap$_M$(prm$_{FR}$)
👤$_M$	enc(swap$_M$(prm$_{BR}$))	swap$_M$(prm$_{BR}$)

swap$_A$(prm) :: ~~swap~~$_M$(prm$_{FR}$) :: ~~swap~~$_M$(prm$_{BR}$) ✗

~~swap~~$_M$(prm$_{BR}$) :: swap$_A$(prm) :: ~~swap~~$_M$(prm$_{FR}$) ✗

...

swap$_M$(prm$_{FR}$) :: swap$_A$(prm) :: swap$_M$(prm$_{BR}$) ✓

Selective Abort
By knowledge of **A**'s swap direction [BCD$^+$21] and choice of order parameters

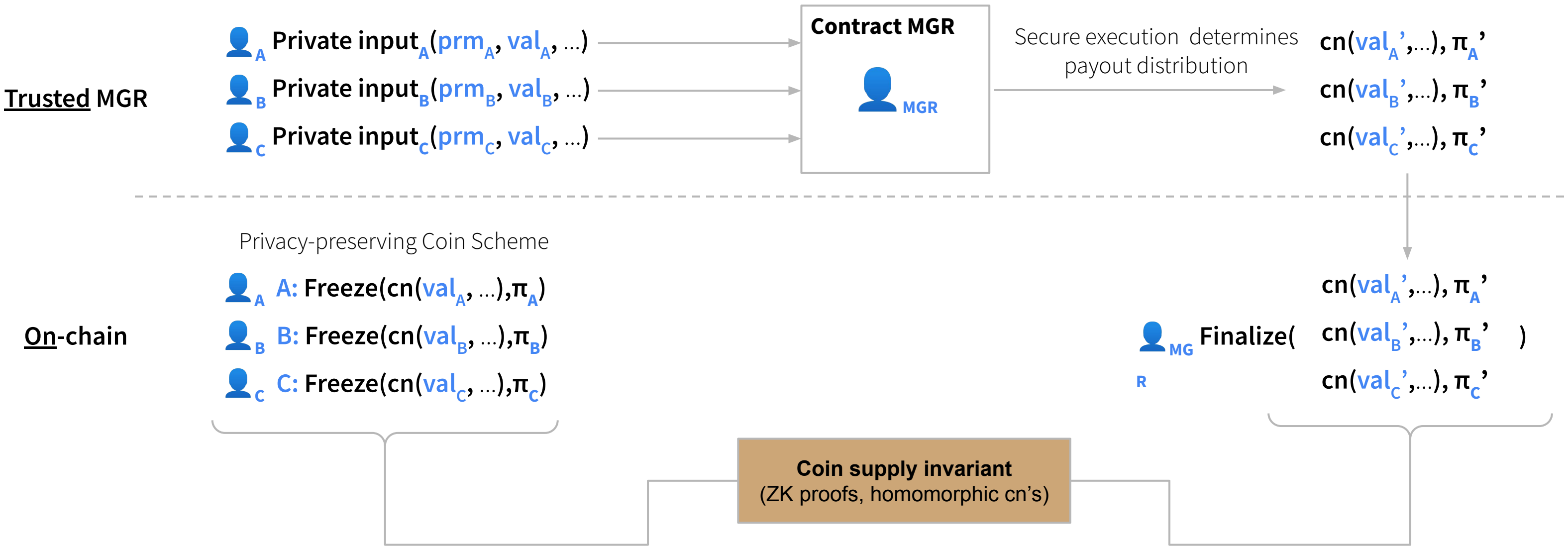
Private balances are necessary to prevent front-running

Front-running **Mitigation**

Miner powers	Mitigation	Proposed Techniques
Action sequencing	Fair Ordering	Fair Ordering Consensus
	Batching of blinded inputs	(Hash Commitments) Time-lock Crypto Threshold Crypto
Inference of user intent	Private balances & state + batching of blinded inputs	Secure Multi-Party Computation

Privacy-preserving Smart Contracts [Hawk]

Blinded Inputs	**Execution** over **blinded** inputs



Trusted MGR

Private input$_A$(prm$_A$, val$_A$, …)

Private input$_B$(prm$_B$, val$_B$, …)

Private input$_C$(prm$_C$, val$_C$, …)

Contract MGR

Secure execution determines payout distribution

cn(val$_A$',…), π_A'

cn(val$_B$',…), π_B'

cn(val$_C$',…), π_C'

On-chain

Privacy-preserving Coin Scheme

A: Freeze(cn(val$_A$, …),π_A)

B: Freeze(cn(val$_B$, …),π_B)

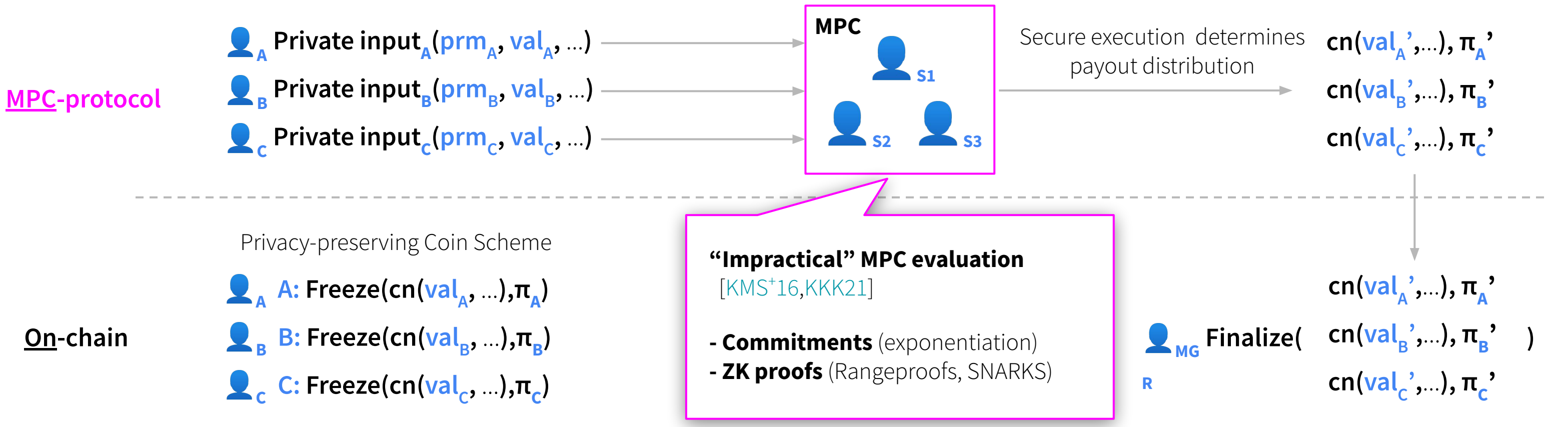
C: Freeze(cn(val$_C$, …),π_C)

Finalize(cn(val$_A$',…), π_A' cn(val$_B$',…), π_B' cn(val$_C$',…), π_C')

Coin supply invariant
(ZK proofs, homomorphic cn's)

Privacy-preserving Smart Contracts **with MPC**

Blinded Inputs	**Execution** over **blinded** inputs



MPC-protocol

Private input$_A$(prm_A, val_A, ...)

Private input$_B$(prm_B, val_B, ...)

Private input$_C$(prm_C, val_C, ...)

Secure execution determines payout distribution

$cn(val_A', ...), \pi_A'$

$cn(val_B', ...), \pi_B'$

$cn(val_C', ...), \pi_C'$

On-chain

Privacy-preserving Coin Scheme

A: Freeze($cn(val_A, ...), \pi_A$)

B: Freeze($cn(val_B, ...), \pi_B$)

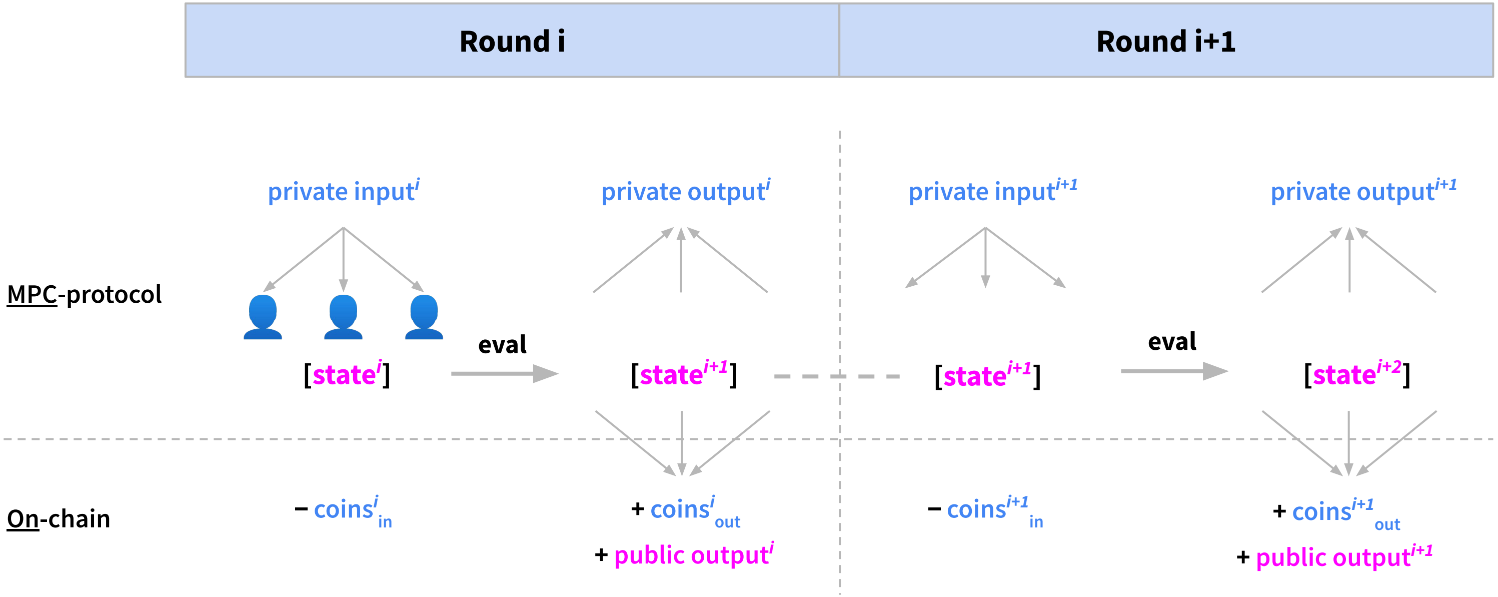
C: Freeze($cn(val_C, ...), \pi_C$)

“Impractical” MPC evaluation
[KMS+16,KKK21]

- **Commitments** (exponentiation)
- **ZK proofs** (Rangeproofs, SNARKS)

Finalize($cn(val_A', ...), \pi_A'$ $cn(val_B', ...), \pi_B'$ $cn(val_C', ...), \pi_C'$)

MPC: Secret Application State



Round i

Round i+1

MPC-protocol

private input^{i}

[state^{i}]

eval

private output^{i}

[state^{i+1}]

private input^{i+1}

[state^{i+1}]

eval

private output^{i+1}

[state^{i+2}]

On-chain

− coins$^{i}_{\text{in}}$

+ coins$^{i}_{\text{out}}$

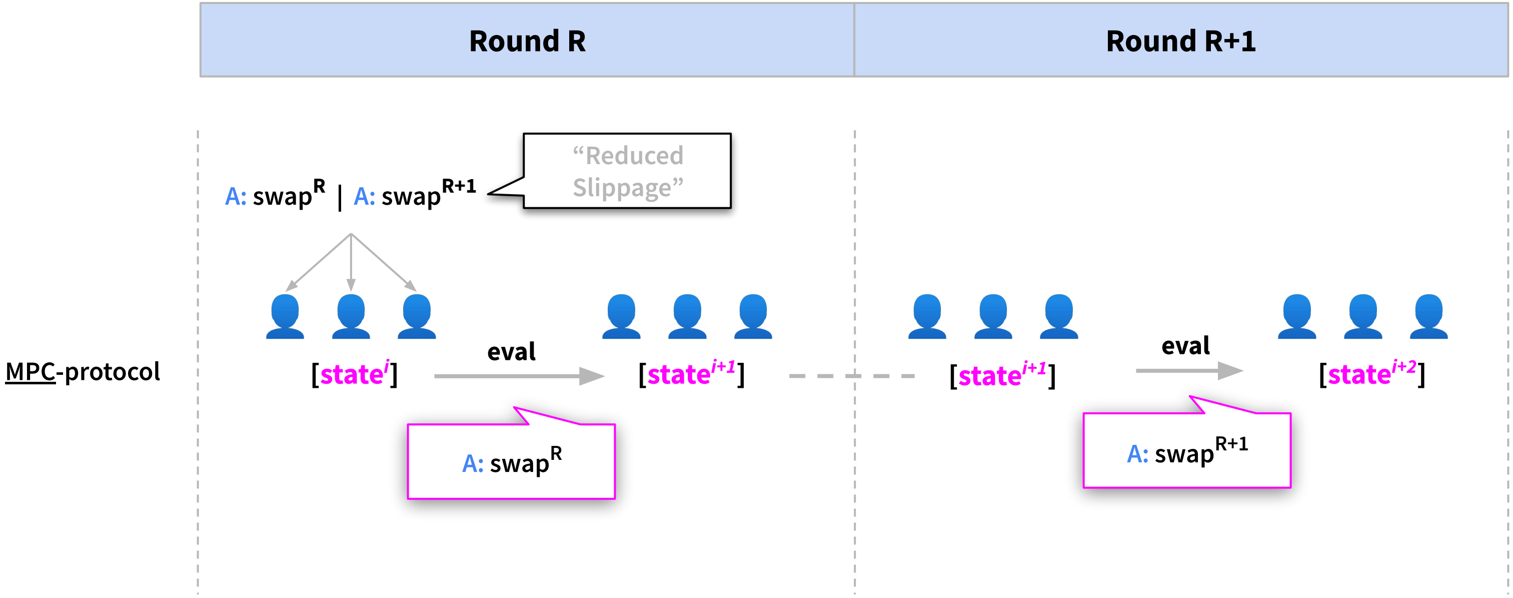
+ public output^{i}

− coins$^{i+1}_{\text{in}}$

+ coins$^{i+1}_{\text{out}}$

+ public output^{i+1}

MPC: Fairly Scheduled Orders



In contrast: Public order schedule can be front-run!

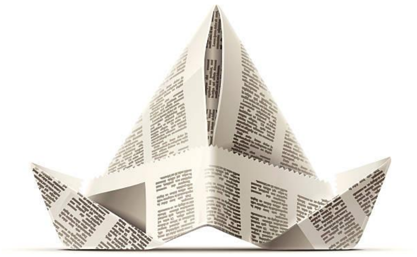
Front-running **Mitigation**

Miner powers	Mitigation	Proposed Techniques
Action sequencing	Fair Ordering	Fair Ordering Consensus
	Batching of blinded inputs	(Hash Commitments) Time-lock Crypto Threshold Crypto
Inference of user intent	Private balances & secret state + batching of blinded inputs	Secure Multi-Party Computation

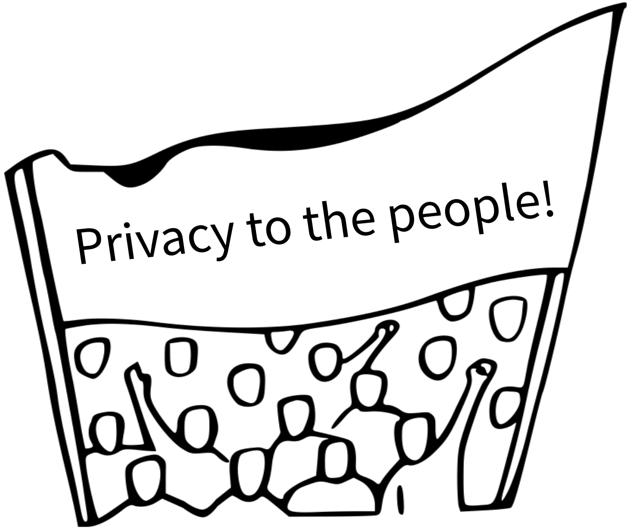
PAPR: Publicly Auditable Privacy Revocation for Anonymous Credentials

CT-RSA 2023

Joakim Brorsson, Lund University
Bernardo David, IT University of Copenhagen
Lorenzo Gentile, IT University of Copenhagen
Elena Pagnin, Chalmers University of Technology
Paul Stankovski Wagner, Lund University



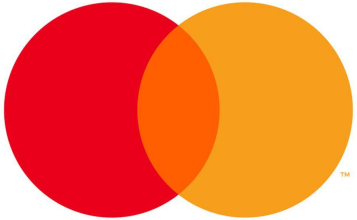
Conflicting interests: user privacy and accountability



More surveillance of
criminals!

Conflict interests: examples

Regulations:
(KYC, AML)



Legal cases:

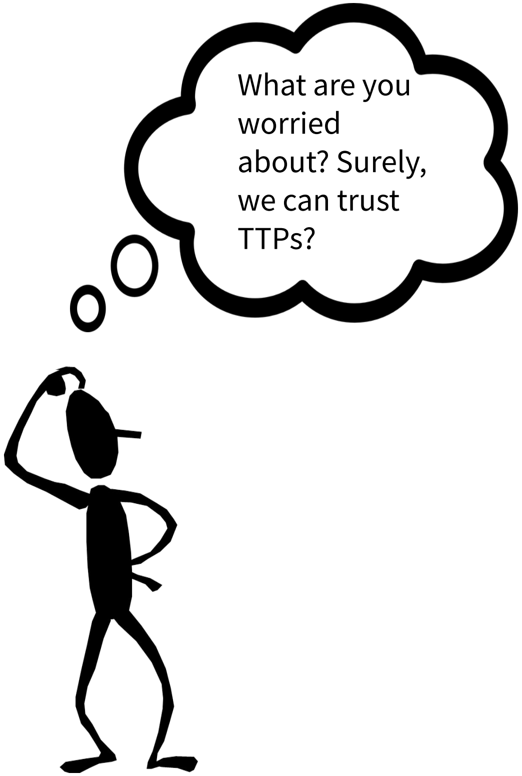
vs.

Conditional privacy

- Conditional privacy avoids privacy vs. accountability conflict
    - Privacy given by default
    - If misbehavior occurs, the privacy can be revoked
- Two flavors of conditional privacy:
    - Identity tracing by ”Self-Revocation”
        - Suitable for well defined misbehavior
        - E.g., double spend in e-cash
        - Does not rely on TTP
    - Central authorities (or central committee) can trace real identity at will
        - Does not limit what can be considered as misbehavior
        - Relies on TTP

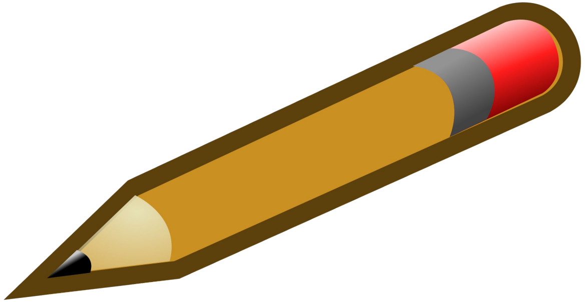
Trusting TTPs

- Are TTPs trustable?
  - e.g. use of IP tracing laws.
- Are TTPs competent?
  - Countless data leaks.
  - Even if we trust honesty of TTP, it might be subject to attacks.



Outline

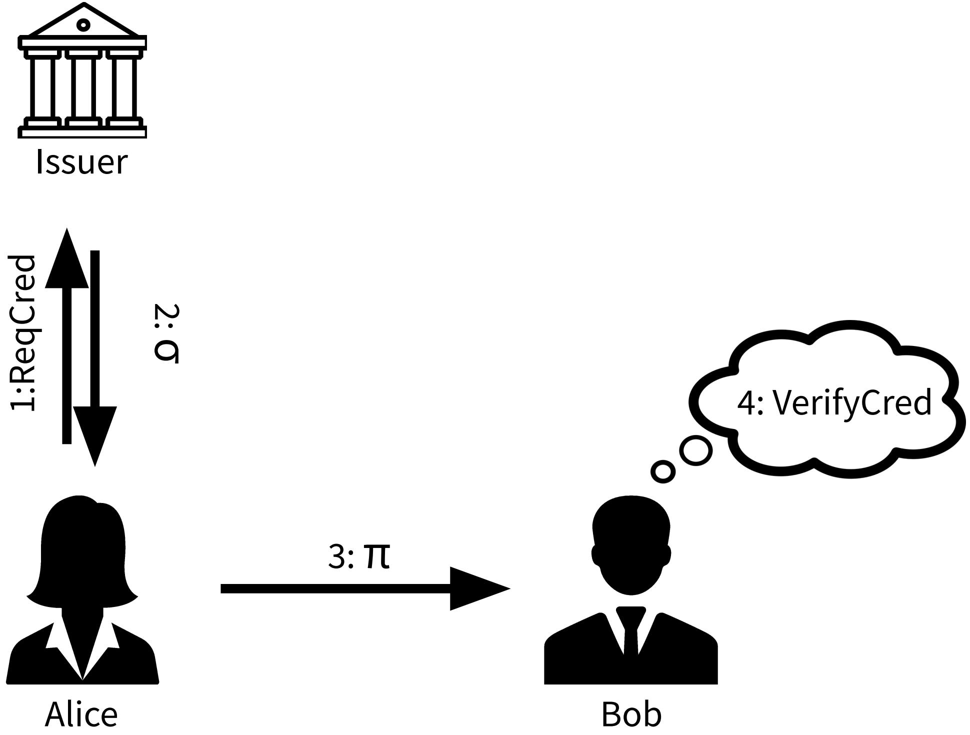
- We will discuss how to create privacy revocation with *public auditability*.
- Apply this tool to anonymous credentials



Background on credentials

Credentials:

- Setup()
- KeyGen() → sk, pk
- ReqCred(pk, ID) → σ
- ShowCred(sk, σ) → π
- VerifyCred(pk, σ, π) → 0/1



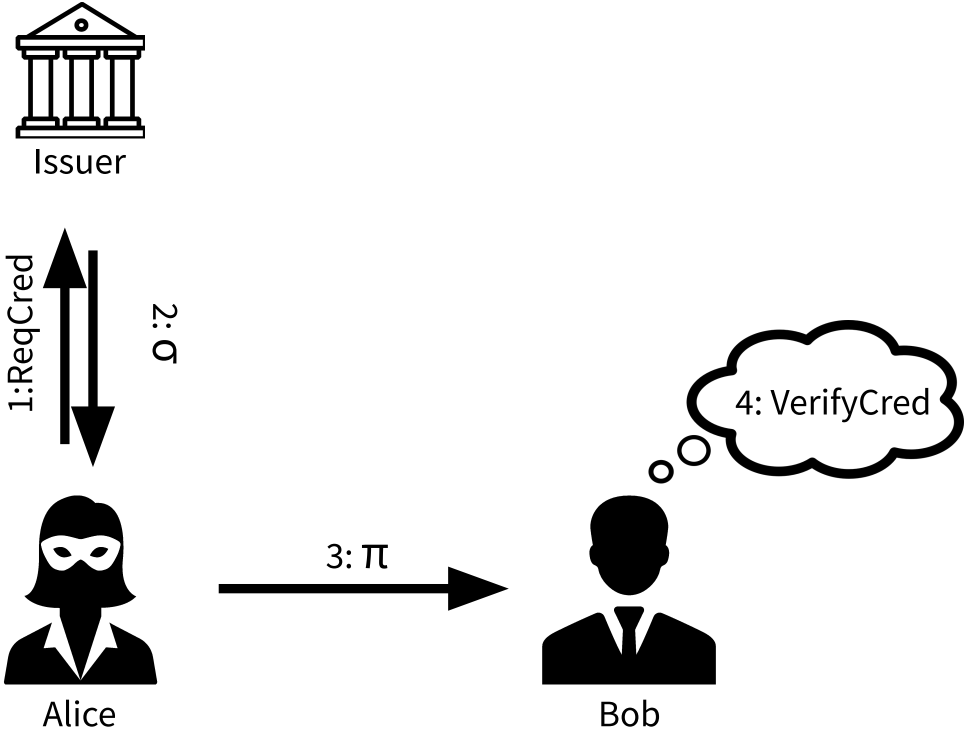
Background on credentials

Anonymous Credentials:

- Anonymous Showing

Credentials:

- Setup()
- KeyGen() → sk, pk
- ReqCred(pk, ID) → σ
- ShowCred(k, σ) → π
- VerifyCred(pk, σ, π) → 0/1

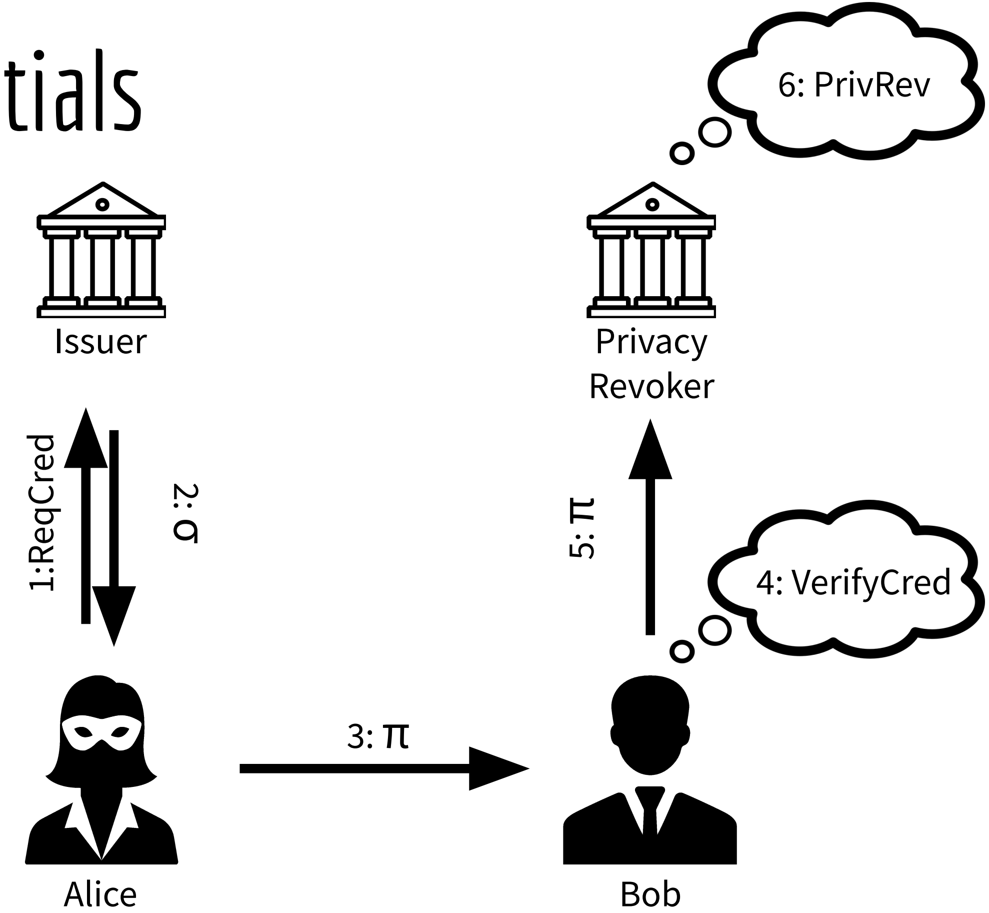


Background on credentials

Revokable Privacy:
- PrivRev(π) → ID

Anonymous Credentials:
- Anonymous Showing

Credentials:
- Setup()
- KeyGen() → sk, pk
- ReqCred(pk, ID) → σ
- ShowCred(sk, σ) → π
- VerifyCred(pk, σ, π) → 0/1



Security properties of PAPR

Definition:
An Anonymous Credential Scheme with *Publicly Auditable Privacy Revocation* has:

Security properties of PAPR

Definition:
An Anonymous Credential Scheme with *Publicly Auditable Privacy Revocation* has:

1. Basic properties of Anonymous Credentials
    - e.g. unforgeability, anonymity

Security properties of PAPR

Definition:
An Anonymous Credential Scheme with *Publicly Auditable Privacy Revocation* has:

1. Basic properties of Anonymous Credentials
    - e.g. unforgeability, anonymity
2. Privacy Revocations possible, but only upon public announcement
    - Models a malicious revocation authority

Security properties of PAPR

Definition:
An Anonymous Credential Scheme with *Publicly Auditable Privacy Revocation* has:

1. Basic properties of Anonymous Credentials
    - e.g. unforgeability, anonymity
2. Privacy Revocations possible, but only upon public announcement
    - Models a malicious revocation authority
3. Guaranteed identity tracing
    - Models a malicious user

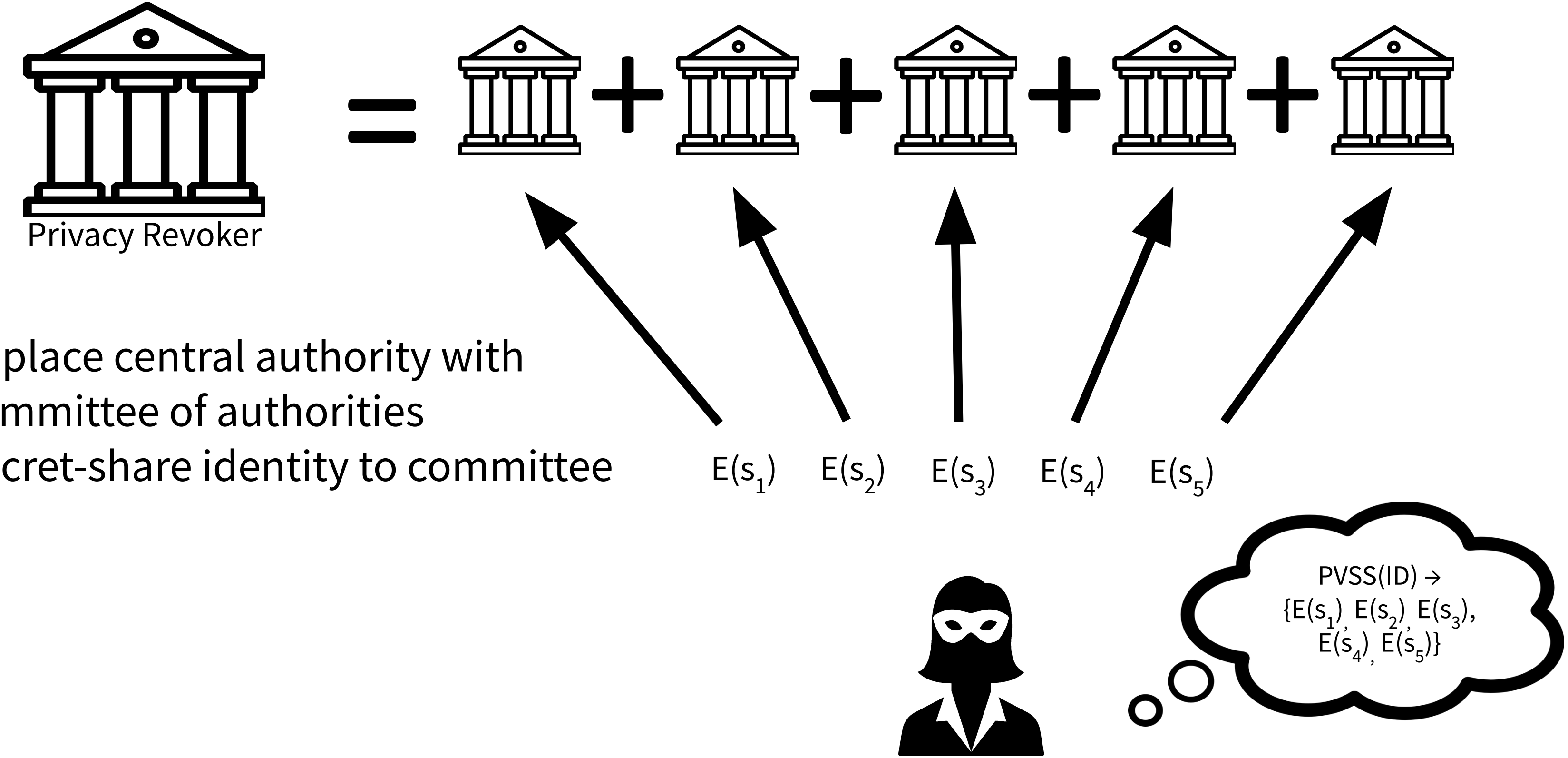
*

Problem

How to guarantee that the privacy revoker is not a “wolf in sheep clothing”?

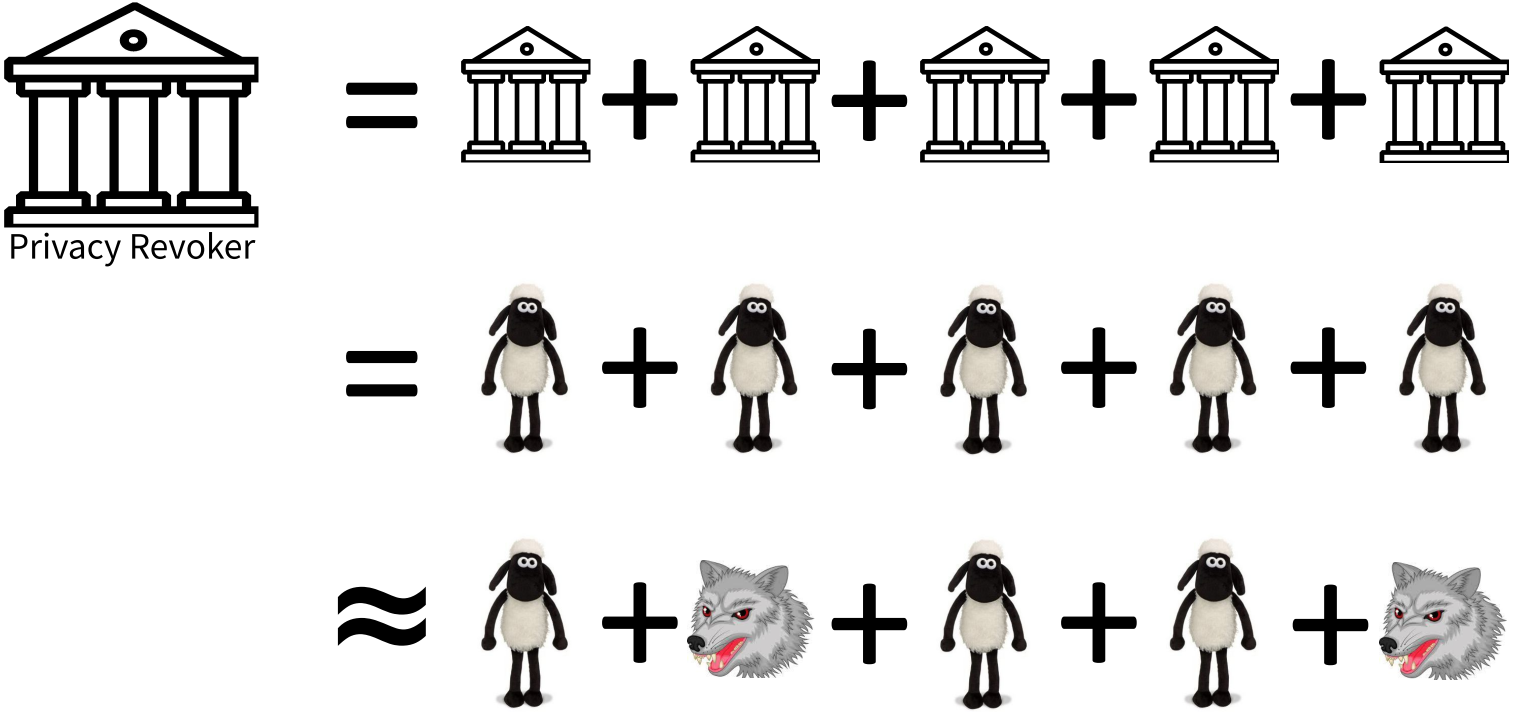
[*] Neither animals were harmed nor cryptographers exposed to risks. Thanks to DALL-E for generating the picture.

Known solutions



- Replace central authority with committee of authorities
- Secret-share identity to committee

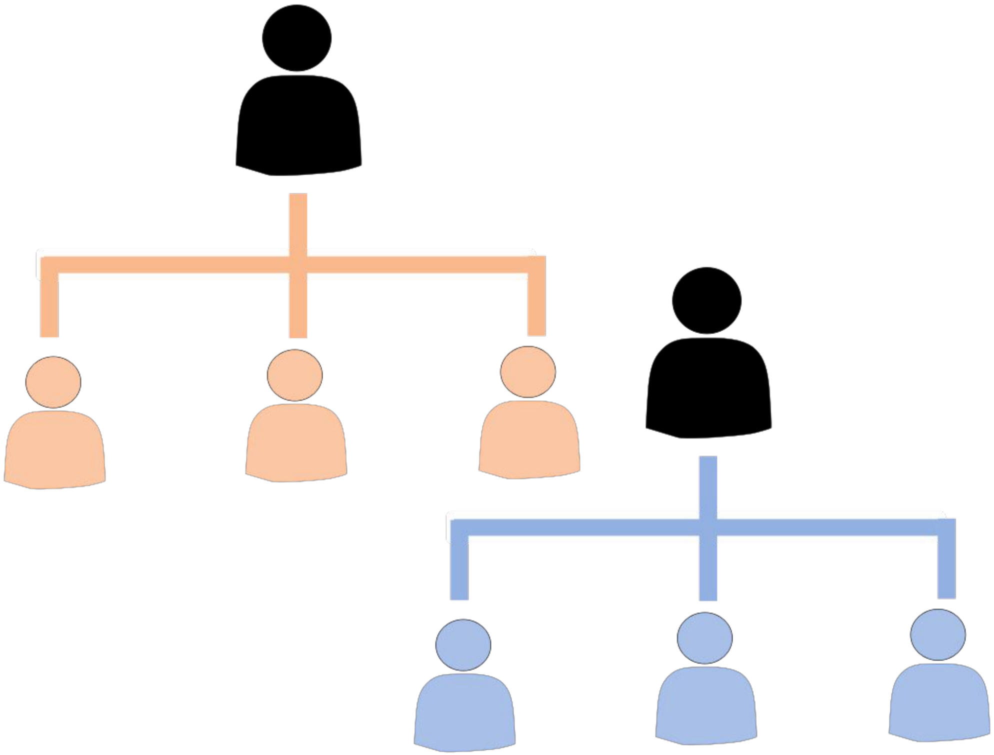
Known solutions



Privacy Revoker

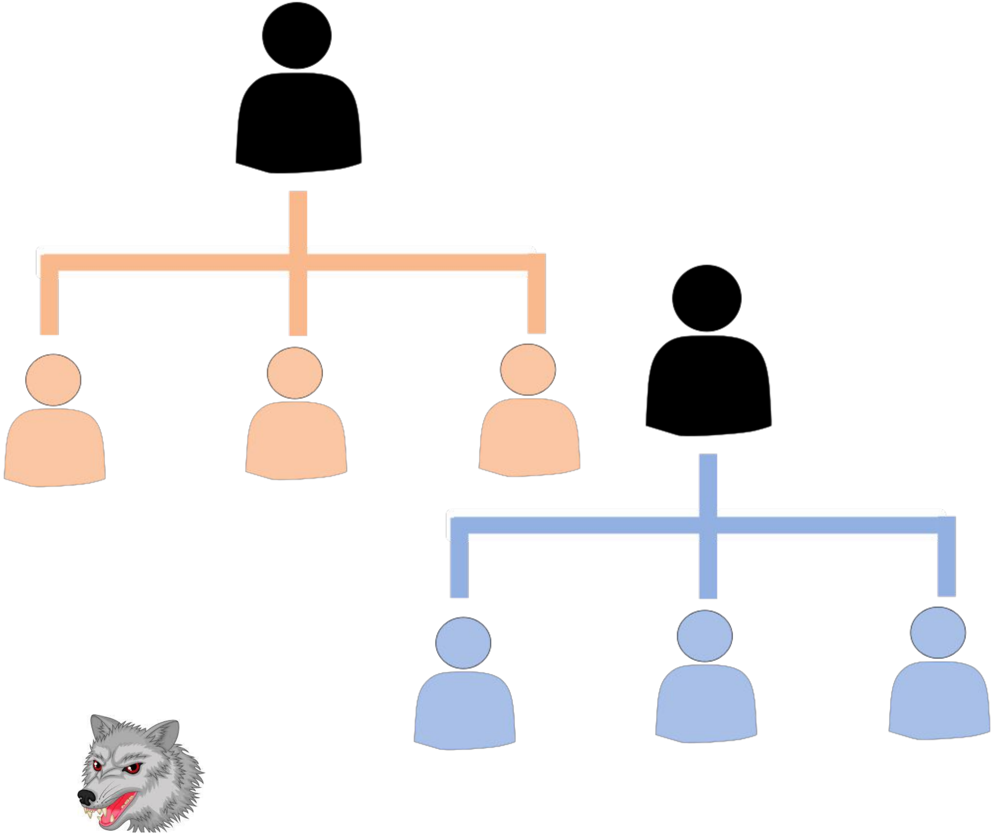
Finding trusted parties

- Hard to find a privacy revoking committee trusted by all users



Finding trusted parties

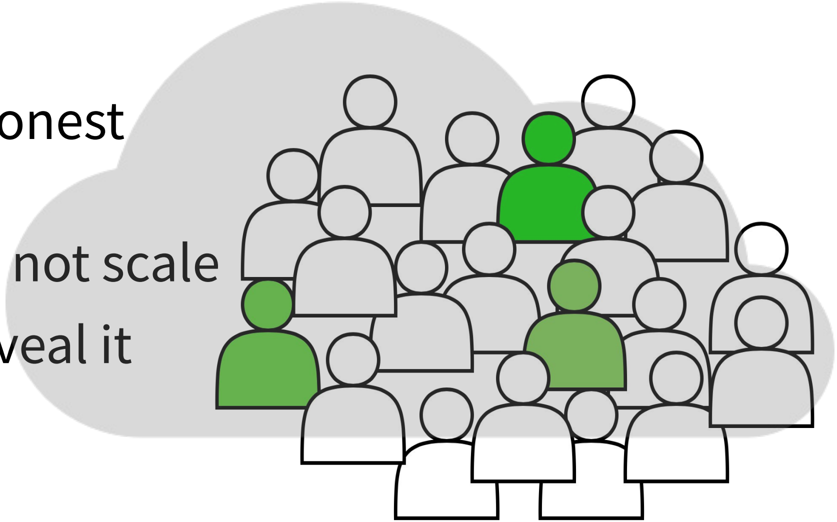
- Hard to find a privacy revoking committee trusted by all users
- A known committee is targetable by powerful adversary
    - Recall examples from introduction



Our solution

Our Solution: Hidden Committees

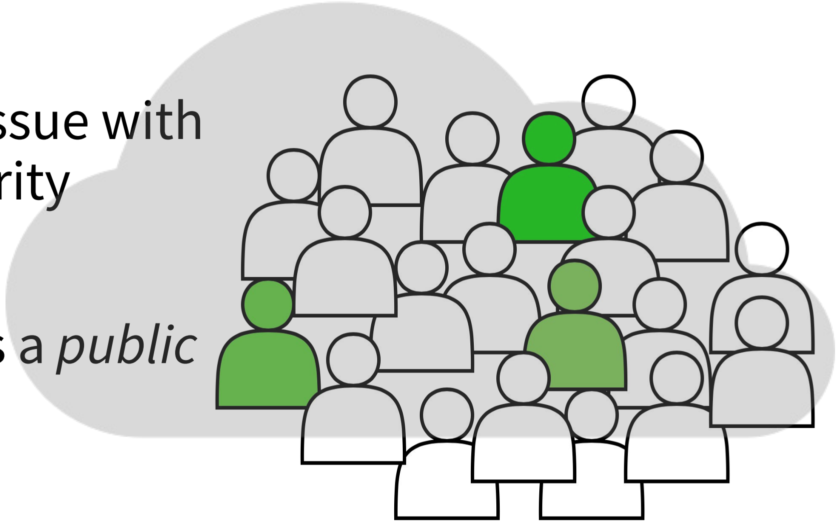
- Assume a large set of candidates with honest majority, e.g. users
- Using all candidates as committee does not scale
- Select a committee at random. Don't reveal it
- Store revocation data with committee



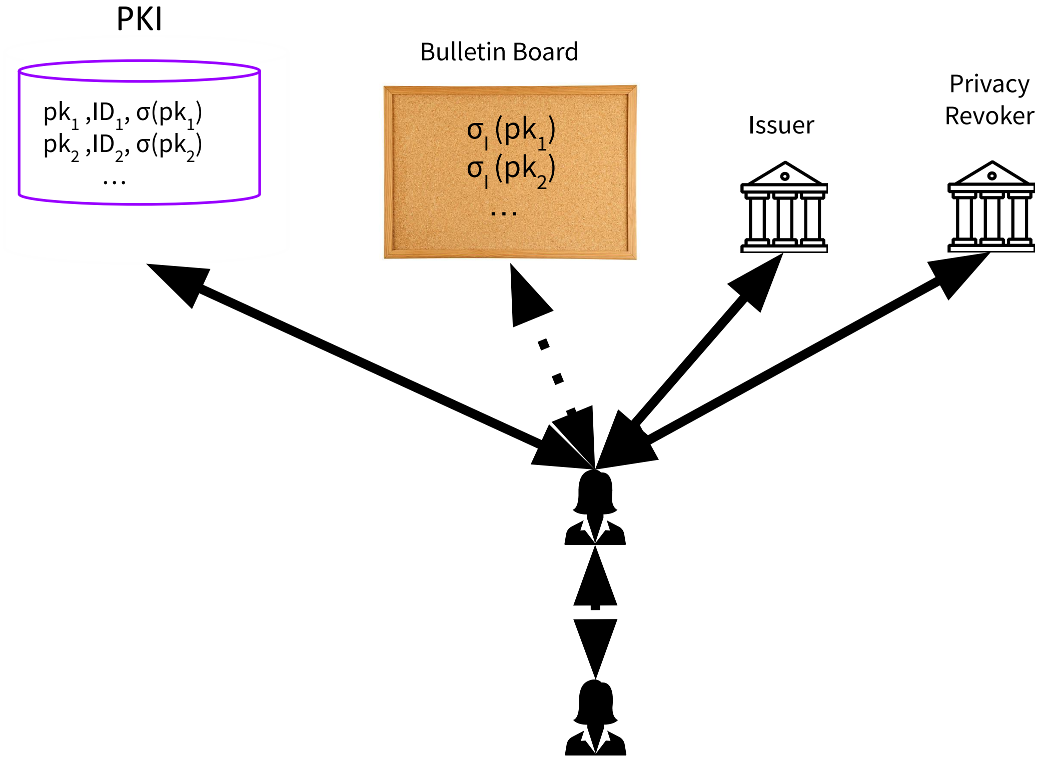
Our solution

How does it solve our problem?

- Finding committee members is a non-issue with random selection from an honest majority
- A Hidden Committee is not targetable
- Thus access to revocation data requires a *public request* for committee cooperation

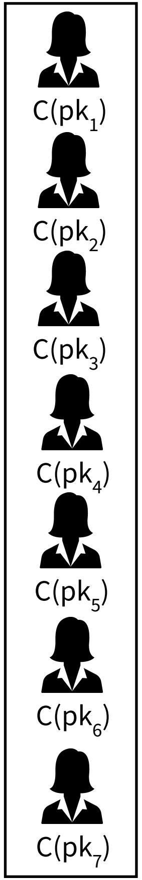


System entities



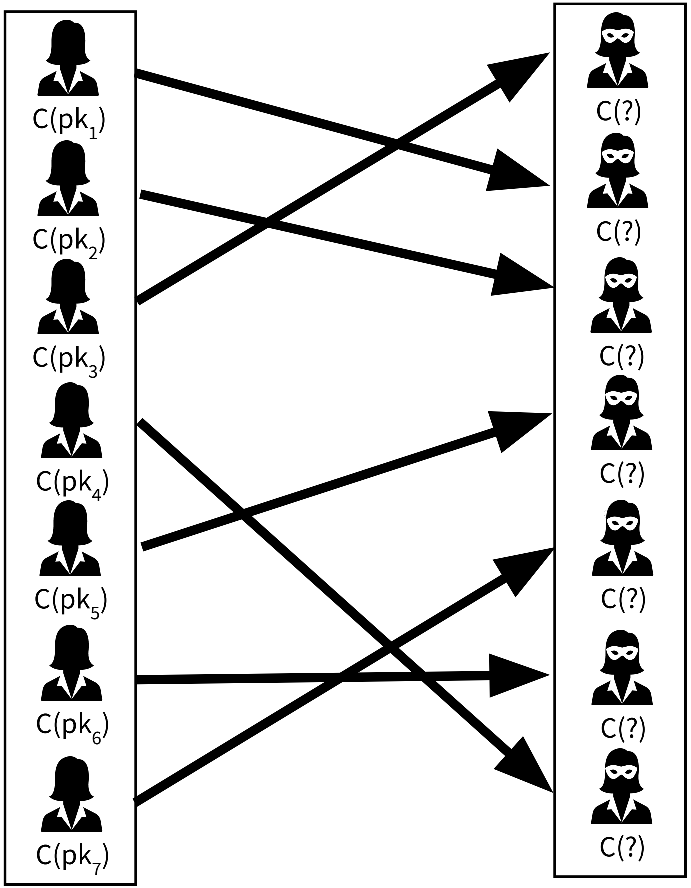
- **PKI** with a list of user public keys and identities
- **Bulletin Board** which users can post anonymously to
- **Users** who can interact anonymously
- **Issuer** issues anonymous credentials
- **Privacy Revoker** revokes anonymity

Local hidden committees



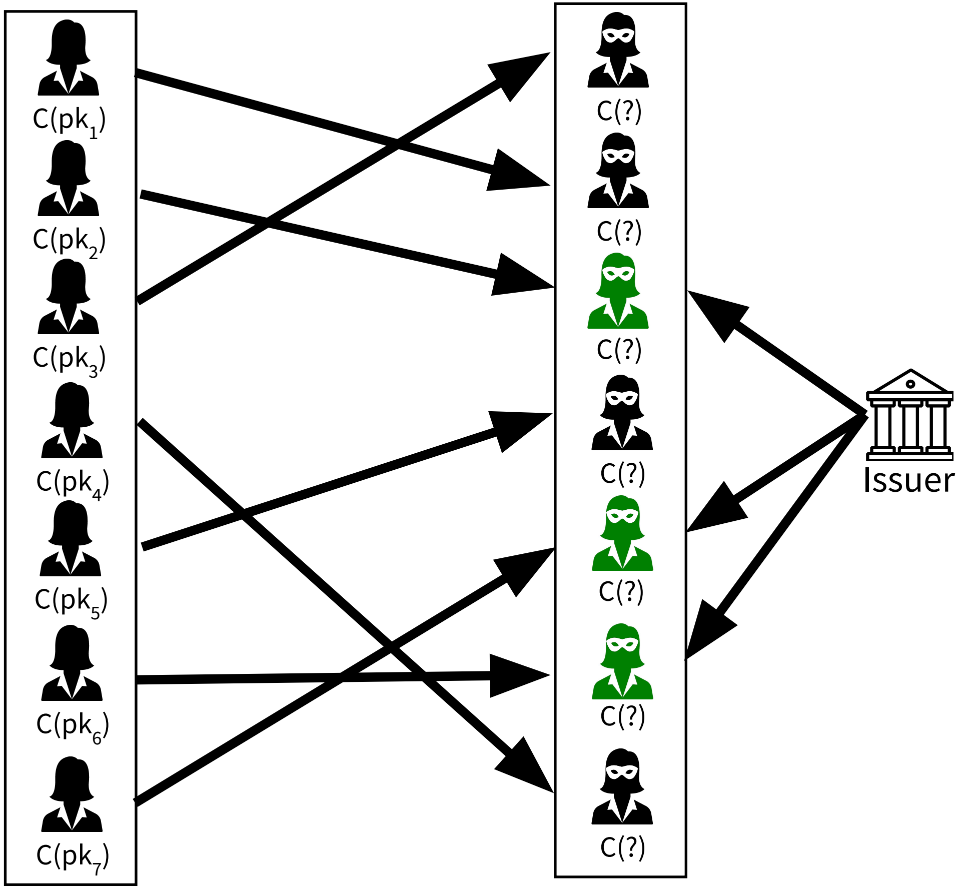
- Each user locally establishes a *random* and *anonymous* committee by:
    1. Obtain list of *all enrolled* public keys and *openly commit to them*

Local hidden committees



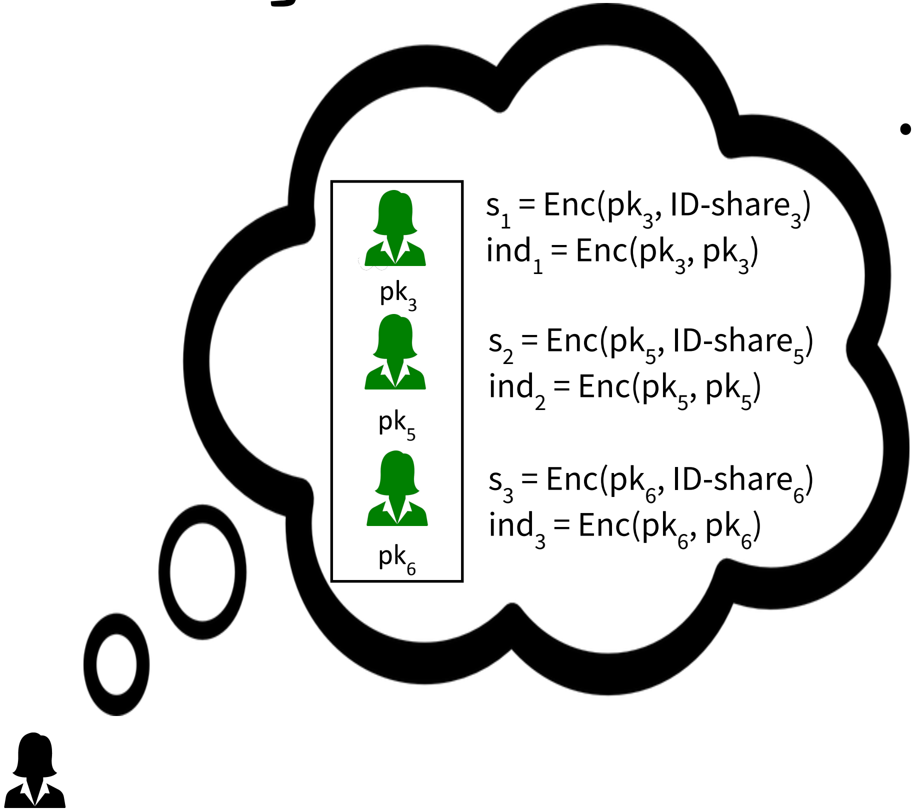
- Each user locally establishes a *random* and *anonymous* committee by:
  1. Obtain list of *all enrolled* public keys and *openly commit to them*
  2. Randomly Shuffle the list and re-randomize the commitments (local operation)
  3. Prove correct shuffling in zero-knowledge
     - Publish on Bulletin Board

Establishing the committee



- Each user locally establishes a *random* and *anonymous* committee by:
  1. Obtain list of *all enrolled* public keys and *openly commit to them*
  2. Randomly Shuffle the list and re-randomize the commitments (local operation)
  3. Prove correct shuffling in zero-knowledge
     - Publish on Bulletin Board
  4. Await issuer randomly selecting a subset of these entries
     - Publish on Bulletin Board

Sharing to committee



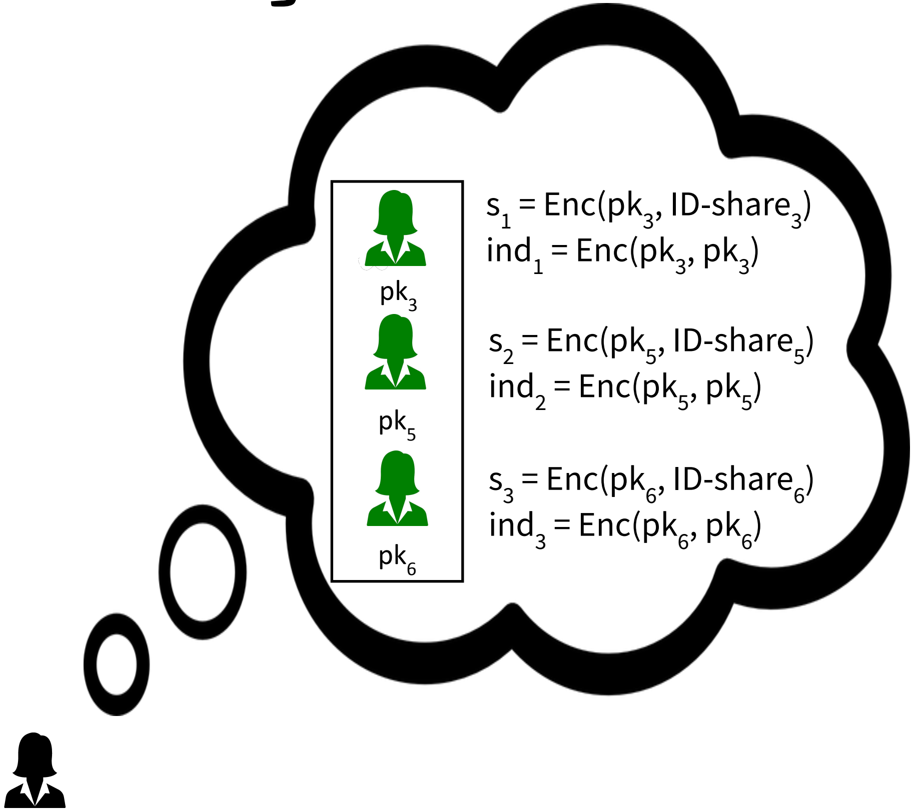
s_1 = Enc(pk_3, ID-$share_3$)
ind_1 = Enc(pk_3, pk_3)

s_2 = Enc(pk_5, ID-$share_5$)
ind_2 = Enc(pk_5, pk_5)

s_3 = Enc(pk_6, ID-$share_6$)
ind_3 = Enc(pk_6, pk_6)

- Escrow Identity:
  1. Construct secret shares of identity
  2. Encrypt *shares* and *indicators* for selected committee
     - *target anonymous encryption*
     - prove correctness of
       - Identity
       - Encrypted Shares
       - Committee
     - Publish on Bulletin Board
  3. Issuer signs credential
     - Publish on Bulletin Board

Sharing to committee



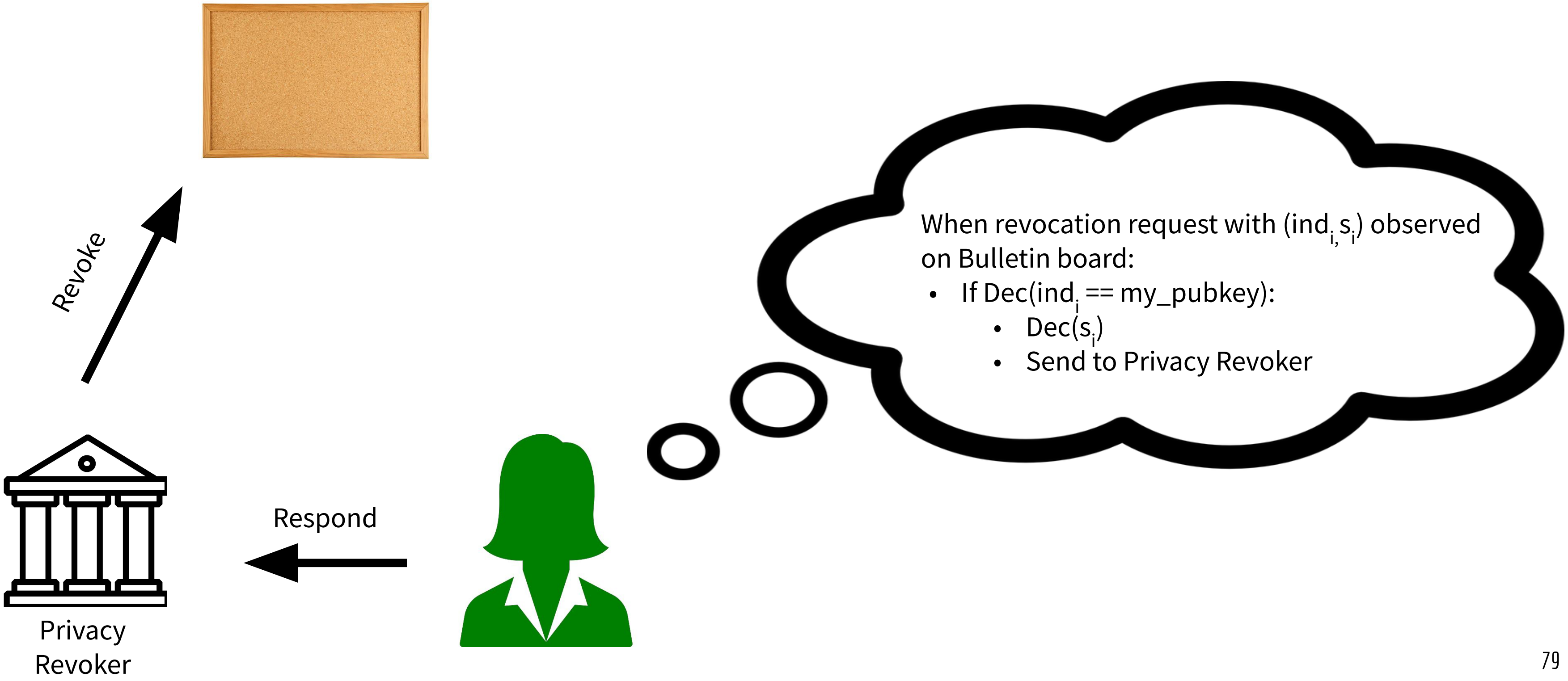
$s_1 = \text{Enc}(pk_3, \text{ID-share}_3)$
$ind_1 = \text{Enc}(pk_3, pk_3)$

$s_2 = \text{Enc}(pk_5, \text{ID-share}_5)$
$ind_2 = \text{Enc}(pk_5, pk_5)$

$s_3 = \text{Enc}(pk_6, \text{ID-share}_6)$
$ind_3 = \text{Enc}(pk_6, pk_6)$

- Result:
1. a hidden committee which can reconstruct the identity of a user
- Note:
1. no global randomness
2. no interaction with committee

Privacy revocation

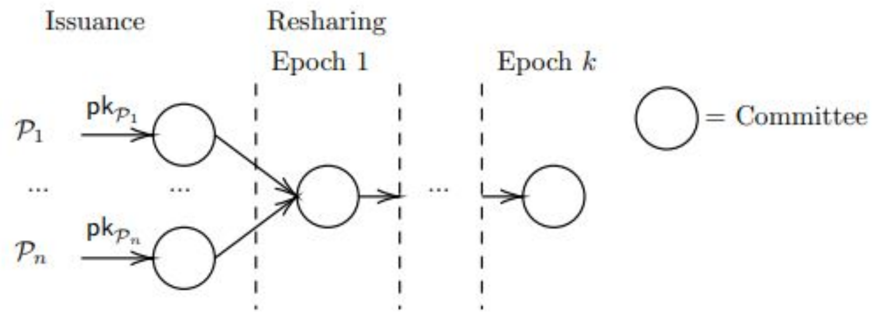


When revocation request with $(ind_{i,} s_i)$ observed on Bulletin board:

```
• If Dec(ind_i == my_pubkey):
    • Dec(s_i)
    • Send to Privacy Revoker
```

From static to mobile adversary

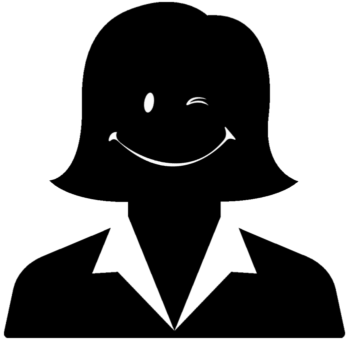
- YOSO proactive secret sharing:
  - Before the start of each epoch, the committees **reshare the identities towards a new single anonymous committee**.



- YOSO threshold encryption:
  - Hidden committee **holds shares of the secret key for threshold encryption**, necessary to **decrypt the identities that are encrypted under the corresponding public key for threshold encryption.**
  - Communication complexity is independent from the number of credentials issued.

Summary

- Alice is now happy, since she has an anonymous credential and will know if her privacy is revoked



- Authorities are happy since they can trace identities of criminals

Conclusion

- In the context of **auctions**, we proposed **efficient MPC protocols for first and second-price sealed-bid auctions** based on **secret deposits**, which represent a novel technique. As **future work**, this technique may be **extended to other applications**.
- In the context of **decentralized finance**, we proposed a **schema of frontrunning mitigation categories**, assessed **state-of-the-art techniques** and illustrated **remaining attacks**. As **future work**, protocols **efficiently realizing these mitigation technique** may be developed.
- In the context of **anonymous credentials**, we introduced the notion of **Publicly Auditable Privacy Revocation (PAPR)** through an **ideal functionality** and proposed a **realization** that is **secure in the Universal Composability (UC) framework**. As **future work**, **efficient non-UC instantiations** may be studied.

Thanks for listening, and all the rest.

Facts about my PhD journey:

- # nationalities of the coauthors: 7 🌏
- # visited countries: 5
- # heartbeats according to my smartwatch: 134.784.000 ❤️
- # lost hairs according to my barber: non-negligible ✂️
- # cool colleagues and friends met: countless 🌈