Blockchain Governance via Sharp Anonymous Multisignatures

Xiangyu Liu, Wonseok Choi, **Vassilis Zikas**
CISPA, DGIST, Georgia Tech

ACM Advances in Financia Technologies
AFT 2025

Thanks to Xiangyu for the slides!

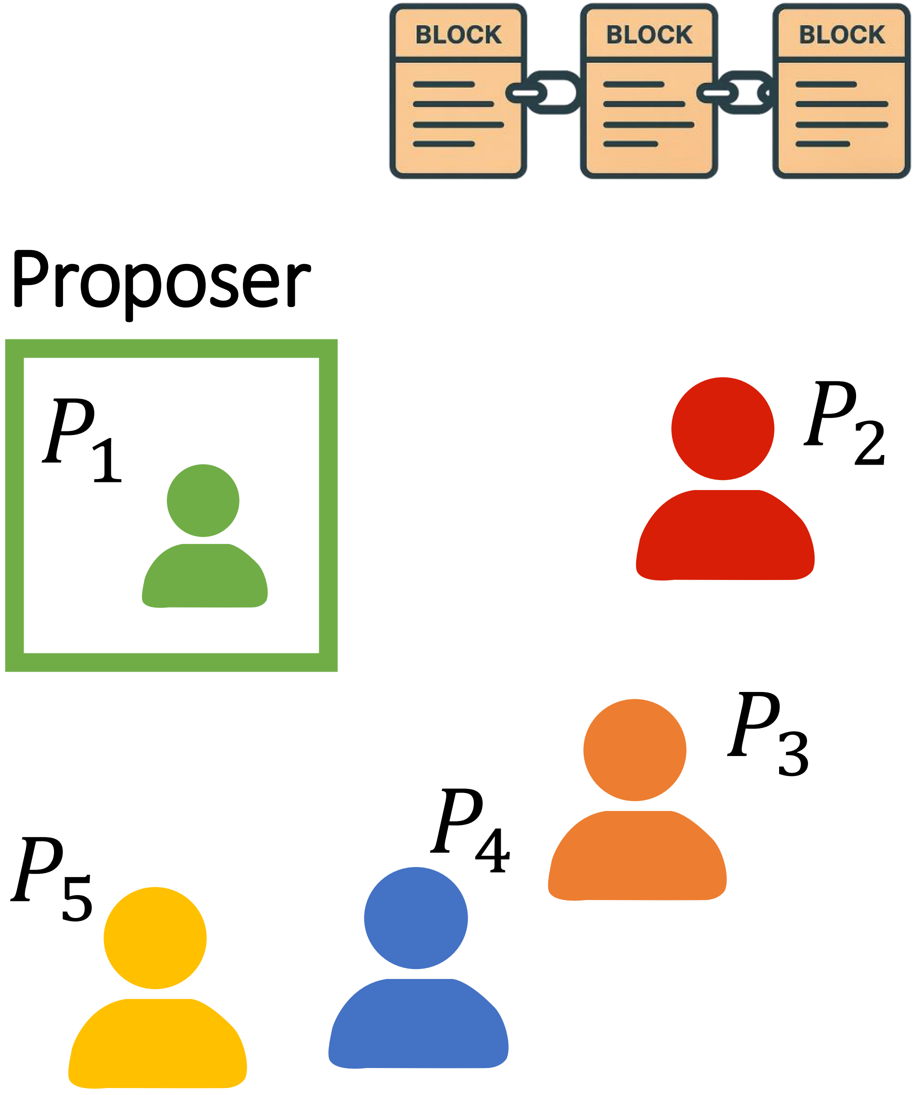
Blockchain Governance via Sharp Anonymous Multisignatures

Xiangyu Liu, Wonseok Choi, **Vassilis Zikas**
CISPA, DGIST, Georgia Tech

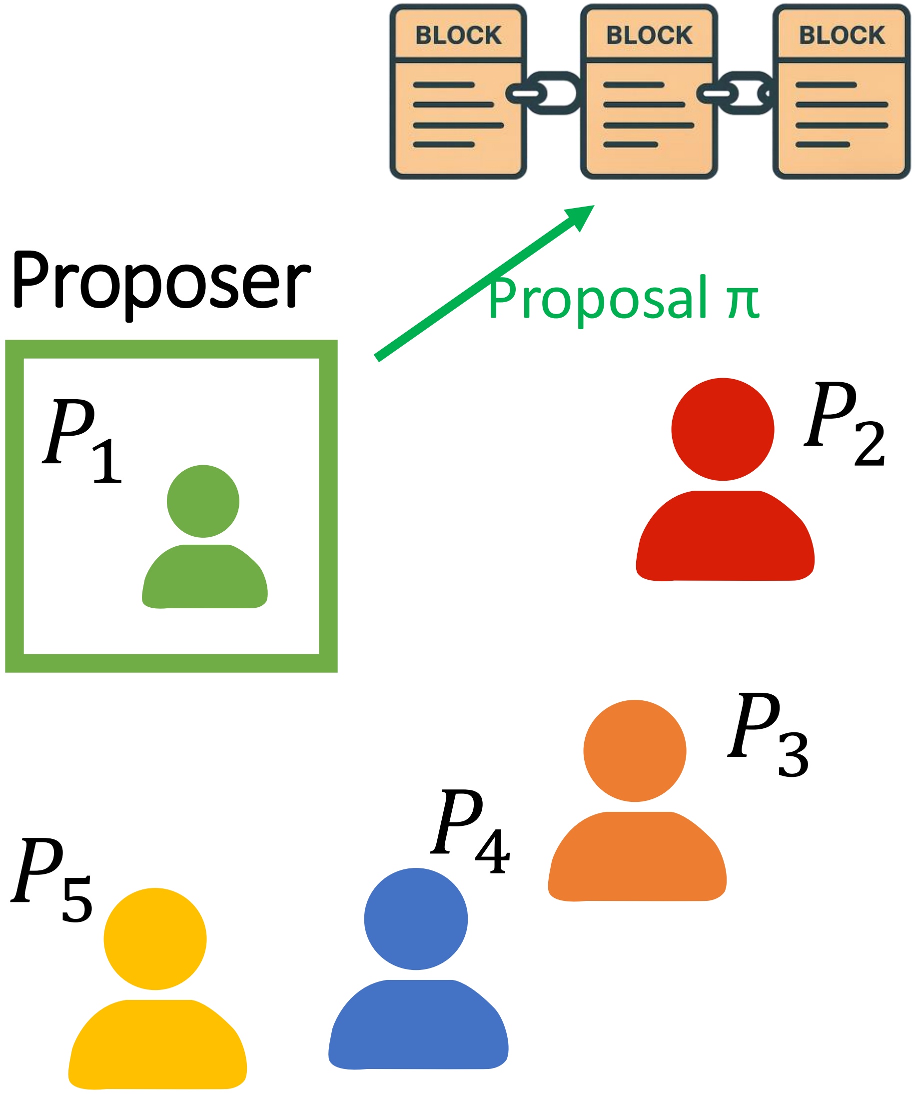
ACM Advances in Financia Technologies
AFT 2025

Thanks to Xiangyu for the slides!

Our Motivating Application: Blockchain Governance

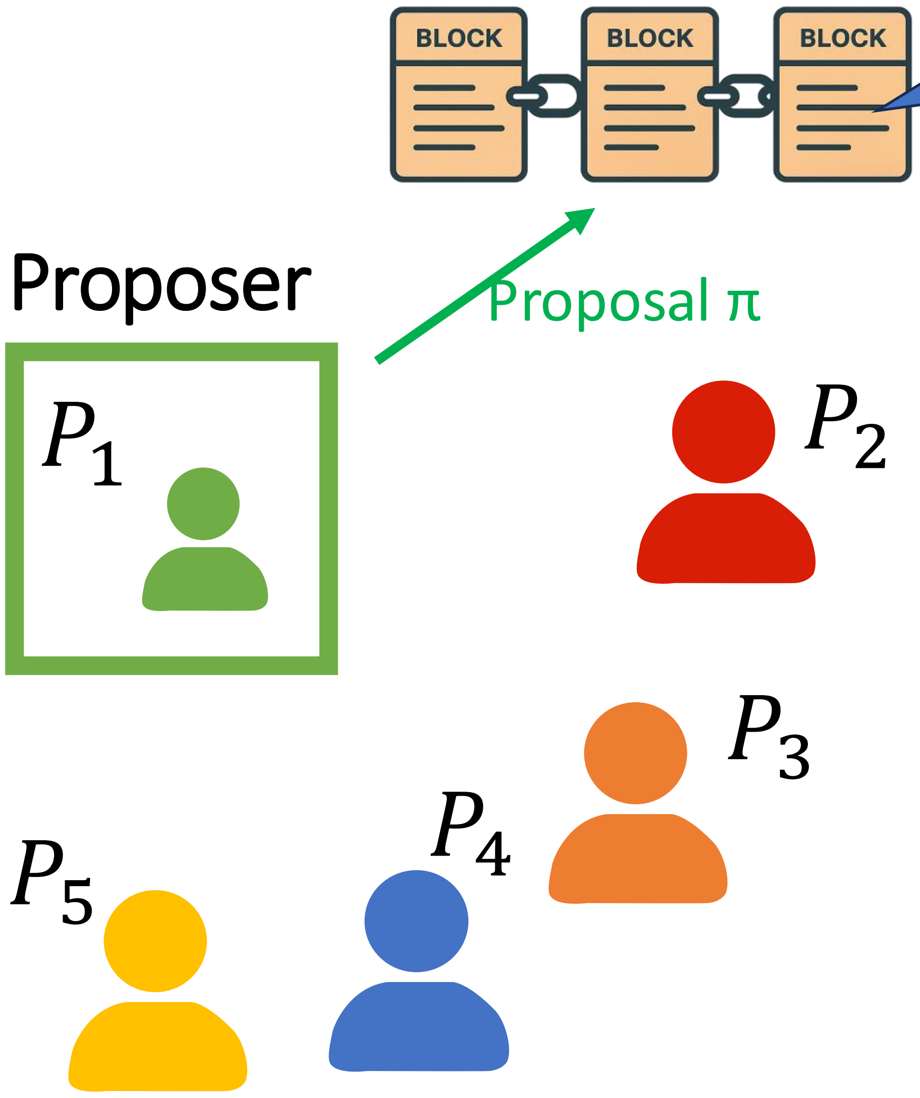


Our Motivating Application: Blockchain Governance



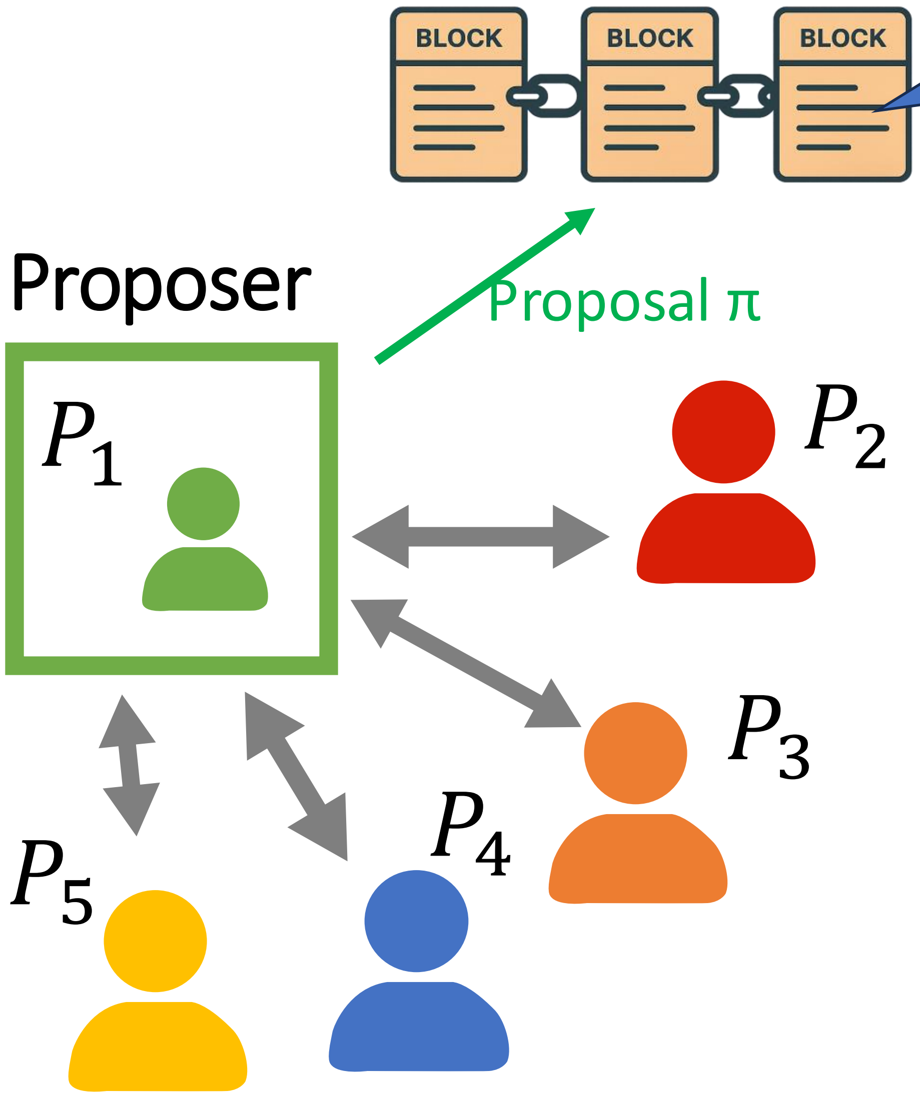
Our Motivating Application: Blockchain Governance

π was voted but t voters



Our Motivating Application: Blockchain Governance

π was voted but t voters



Desiderata/Security Properties

- **Trustless:** No ceremonies/trusted third parties/party honesty assumpt.
- **Round efficient:** Minimal interaction
- **Oblivious:** Voters should not learn information about other voters' intend before casting a vote
- **Post-quantum Untamperability:** Noone can change the number of votes, not even quantum attackers
- **Traceability:** No voter can undedectably vote more than once (for each proposal)
- **Unconditional Anonymity**: Noone should be able to learn what each party voted, even with unlimited computing power.
- ~~**Incoercibility**: Noone should be able to coerce a voter~~

Desiderata/Security Properties

- **Trustless:** No ceremonies/trusted third parties/party honesty assumpt.
- **Round efficient:** Minimal interaction
- **Oblivious:** Voters should not learn information about other voters' intend before casting a vote
- **Post-quantum Untamperability:** Noone can change the number of votes, not even quantum attackers
- **Traceability:** No voter can undedectably vote more than once (for each proposal)
- **Unconditional Anonymity**: Noone should be able to learn what each party voted, even with unlimited computing power.

Desiderata/Security Properties

- The Proposer might decrease (but not increase) votes he receives
- Noone else can change the number of votes

- **Oblivious:** Voters should not learn informa … voters intend before casting a vote
- **Post-quantum Untamperability:** Noone can change the number of votes, not even quantum attackers
- **Traceability:** No voter can undedectably vote more than once (for each proposal)
- **Unconditional Anonymity**: Noone should be able to learn what each party voted, even with unlimited computing power.

Desiderata/Security Properties

- The Proposer might decrease (but not increase) votes he receives
- Noone else can change the number of votes

- **Oblivious:** voters should not learn informa voters intend before casting a vote
- **Post-quantum Untamperability:** Noone can change the number of votes, not even quantum attackers
- **Traceability:** No voter can undedectably vote more than once (for each proposal)
- **Unconditional Anonymity**: Noone should be able to learn what each party voted, even with unlimited computing power.

Desiderata/Security Properties

- The Proposer might decrease (but not increase) votes he receives
- Noone else can change the number of votes

- **Oblivious:** voters should not learn informa voters intend before casting a vote
- **Post-quantum Untamperability:** Noone can change the number of votes, not even quantum attackers
- **Traceability:** No voter can ~~undedectably~~ vote more than once (for each proposal)
- **Unconditional Anonymity**: Noone should be able to learn what each party voted, even with unlimited computing power.

Desiderata/Security Properties

- The Proposer might decrease (but not increase) votes he receives
- Noone else can change the number of votes

- **Oblivious:** voters should not learn informa … voters intend before casting a vote
- **Post-quantum Untamperability:** Noone can change the number of votes, not even quantum attackers
- **Traceability:** No voter can ~~undedectably~~ vote more than once (for each proposal)
- **Unconditional Anonymity**: Noone should be able to learn what each party voted, even with unlimited computing power, except for the Proposer but he should no be able to prove it.

Our Contributions

- Sharp anonymous multisignatures (#AMS): A primitive that natively achieves all the above properties
- Relation to Threshold Ring Signatures (TRS)
- A template for building #AMS from (Lossy) Chameleon Hashing
  - Instantiations under different assumptions yield unconditional anonymity + postquantum security
  - A generic template abstracting several known TRS schemes
- Concrete instantiations of our governance goals using #AMS
  - Interactive-Voting on multiple proposals
  - Vote-and-go approach
  - Vote on only one proposal
- As a side-product: Relation between Lossy Identification and Lossy Chameleon Hashing.

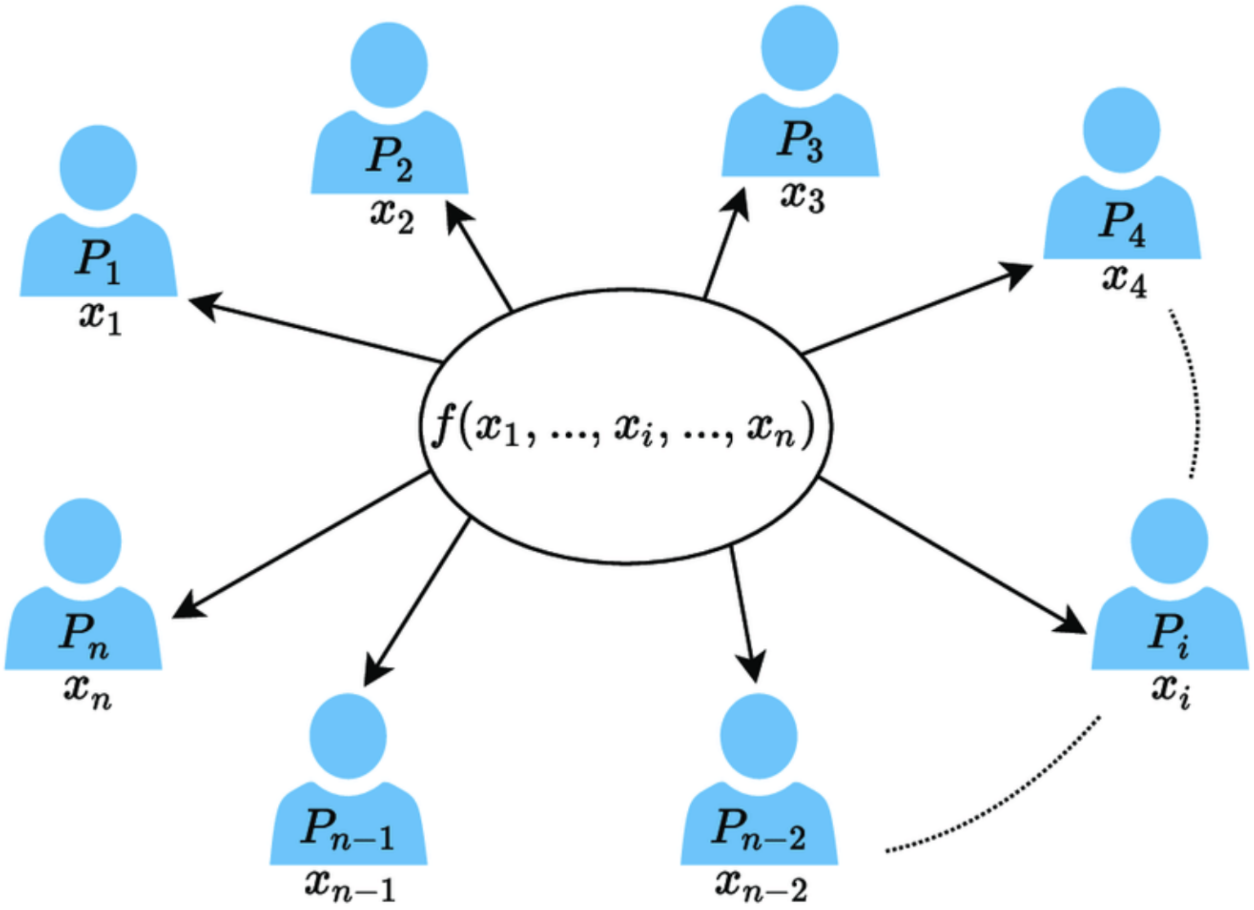
Our Contributions

- Sharp anonymous multisignatures (#AMS): A primitive that natively achieves all the above properties
- Relation to Threshold Ring Signatures (TRS)
- A template for building #AMS from (Lossy) Chameleon Hashing
  - Instantiations under different assumptions yield unconditional anonymity + postquantum security
  - A generic template abstracting several known TRS schemes
- Concrete instantiations of our governance goals using #AMS
  - Interactive-Voting on multiple proposals
  - Vote-and-go approach
  - Vote on only one proposal
- As a side-product: Relation between Lossy Identification and Lossy Chameleon Hashing.

Our Contributions

- Sharp anonymous multisignatures (#AMS): A primitive that natively achieves all the above properties
- Relation to Threshold Ring Signatures (TRS)
- A template for building #AMS from (Lossy) Chameleon Hashing
  - Instantiations under different assumptions yield unconditional anonymity + postquantum security
  - A generic template abstracting several known TRS schemes
- Concrete instantiations of our governance goals using #AMS
  - Interactive-Voting on multiple proposals
  - Vote-and-go approach
  - Vote on only one proposal
- As a side-product: Relation between Lossy Identification and Lossy Chameleon Hashing.

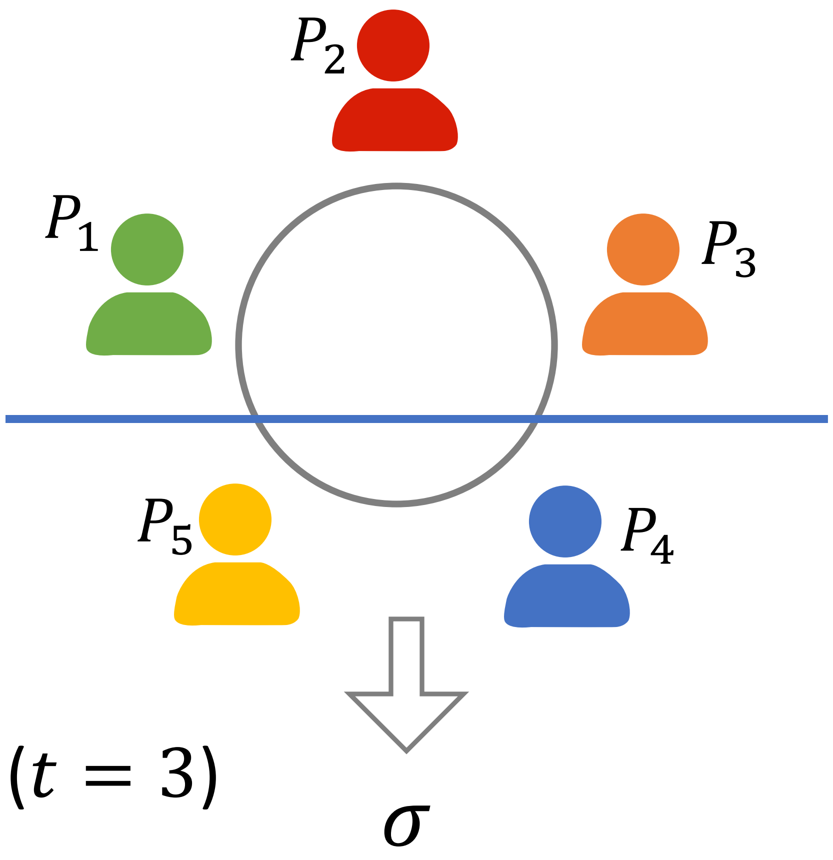
Attempt 1: Multi-party Computation



Yes, but:

- information-theoretic MPC needs an honest majority
- very costly

Attempt 2: Threshold Ring Signatures



Security

- **Correctness:** Any set of at least t parties can generate a signature
- **Unforgeability:** An adversary with less than t signing keys cannot forge
- **Anonymity:** The set of signatures hides the identity of the signers

Attempt 2: Threshold Ring Signatures

Close …

+ Can achieve unconditional anonymity
+ Trustless
+ Post-quantum (unforgeability) constructions exist, e.g., based on Latices (SIS, LWE).

… but not there

- A 0/1 definition (does not export the number t)
- Typically anonymity is for the final aggregated signature (adversary not a signer)
- Is t predefined/known to signers?
- Does anonymity hold among signers?

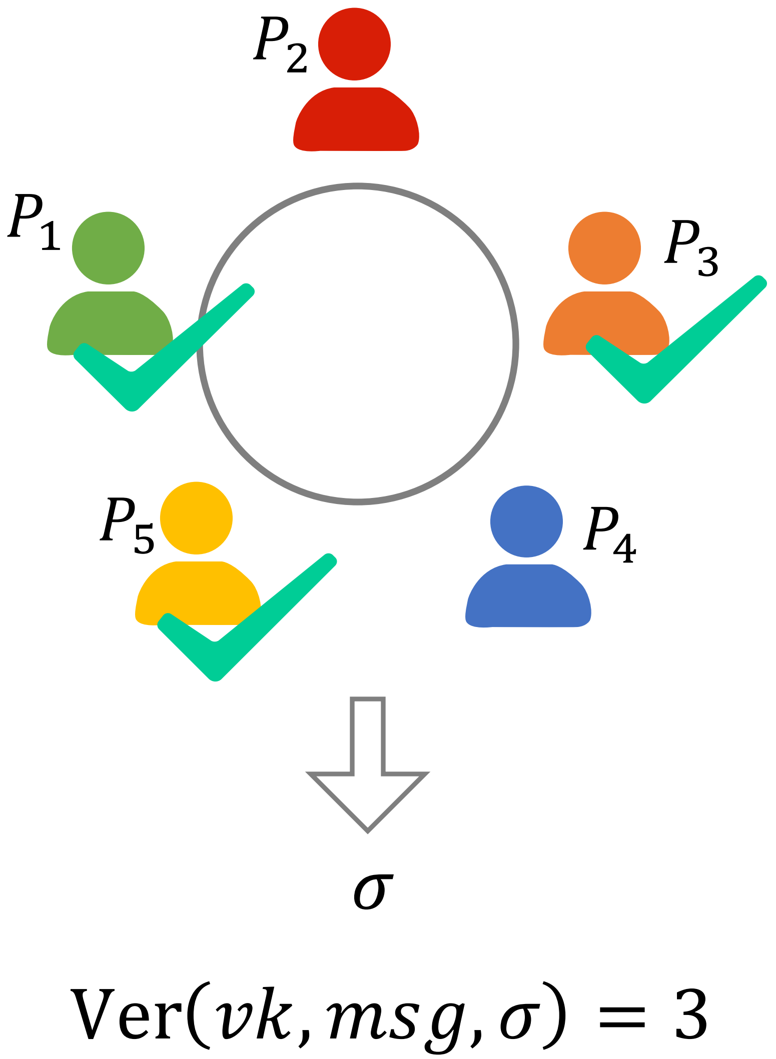
Due to the non-interactive definition

Our New Primitive: #AMS

#AMS: Sharp Anonymous MultiSignatures

$\mathrm{Ver}(vk, msg, \sigma)$ outputs the number of parties participating in the signing

- Correctness
- Unforgeability/Untamperability
- (unconditional) Anonymity (even against insiders)



Our New Primitive: #AMS

Non-interactive version similar issues as TRS

- Instead we define it as a protocol where partial signatures and t appear explicit

Our New Primitive: #AMS

Non-interactive version similar issues as TRS

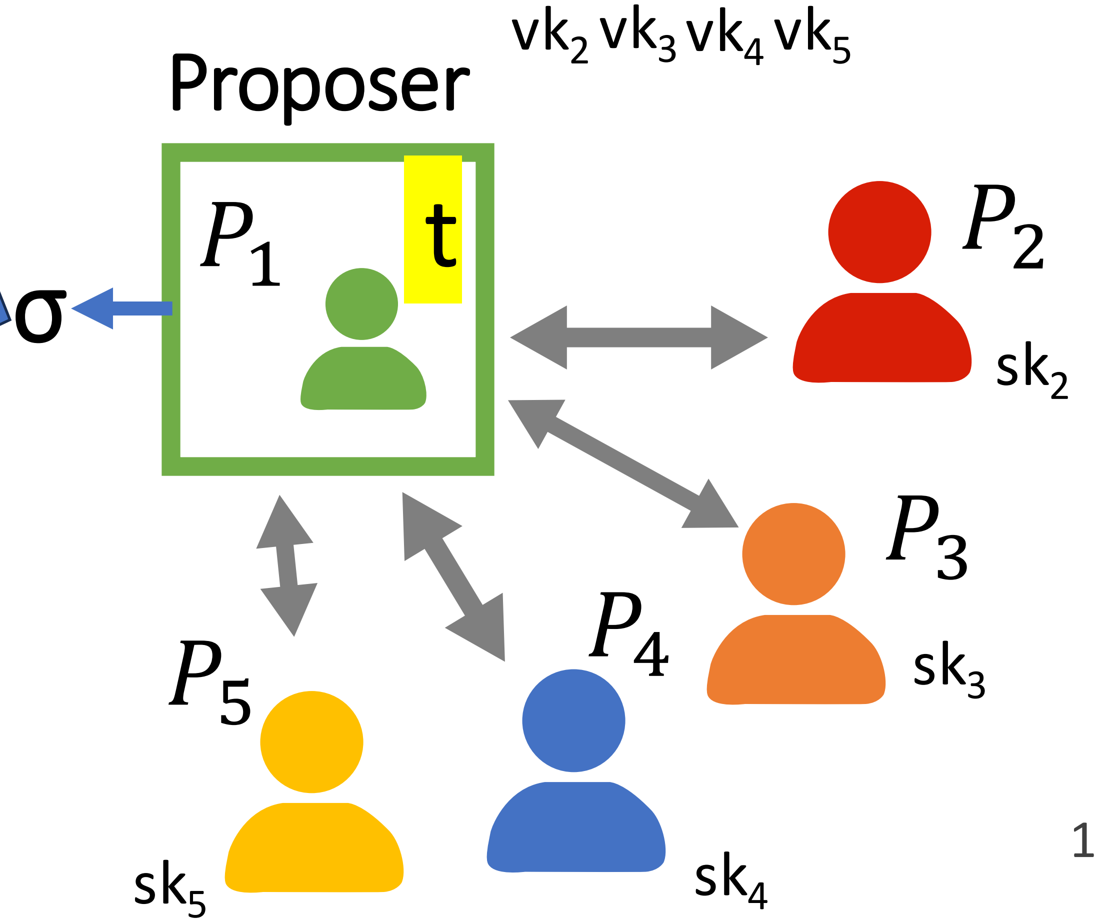
- Instead we define it as a protocol where partial signatures and t appear explicit

-**Correctness:** σ's verification outputs t

-**Unforgeability:** P_1 cannot generate a signature that verifies as t'>t

- **Anonymity:** Only P_1 learns the identities of the signers and he cannot publicly prove it

-**Obliviousness:** Parties (other than P_1) do not learn t during signature generation



Related Primitive: Graded Signatures [KOT15]

Also anonymous signatures aggreated by a moderator

But …

- Defintion requires trusted setup to generate and disrtibuted master keys
  - Similar in flavor to ID-based signature
- No unconditional anonymity
- No post-quantum secure instantiation

Our Contributions

- Sharp anonymous multisignatures (#AMS): A primitive that natively achieves all the above properties
- Relation to Threshold Ring Signatures (TRS)
- A template for building #AMS from (Lossy) Chameleon Hashing
  - Instantiations under different assumptions yield unconditional anonymity + postquantum security
  - A generic template abstracting several known TRS schemes
- Concrete instantiations of our governance goals using #AMS
  - Interactive-Voting on multiple proposals
  - Vote-and-go approach
  - Vote on only one proposal
- As a side-product: Relation between Lossy Identification and Lossy Chameleon Hashing.

Conditional Compiler from TRS … with caveats

TRS with **flexible threshold**

Generic compiler

Sign $(msg||t)$

"Weak" #AMS

sign both message and threshold

Issues:

- Desiderata do not follow from definition
- Not oblivious (voters learn t before they vote)

Our Contributions

- Sharp anonymous multisignatures (#AMS): A primitive that natively achieves all the above properties
- Relation to Threshold Ring Signatures (TRS)
- A template for building #AMS from (Lossy) Chameleon Hashing
  - Instantiations under different assumptions yield unconditional anonymity + postquantum security
  - A generic template abstracting several known TRS schemes
- Concrete instantiations of our governance goals using #AMS
  - Interactive-Voting on multiple proposals
  - Vote-and-go approach
  - Vote on only one proposal
- As a side-product: Relation between Lossy Identification and Lossy Chameleon Hashing.

Background: Chameleon Hashes

$$(hk, td) \leftarrow \mathrm{KGen}$$

$$\boxed{m}\boxed{r} \xrightarrow{H_{hk}(\cdot)} \boxed{h}$$

have td: **easy** to find collision

no td: **hard** to find collision

A Collision:

find $(m_1, r_1) \neq (m_2, r_2)$ s.t.

$$H_{hk}(m_1, r_1) = H_{hk}(m_2, r_2)$$

Background: Chameleon Hashes

Implementable from all standard cryptographic assumptions, including post quantum

$$(hk, td) \leftarrow \mathrm{KGen}$$

$$\boxed{m}\boxed{r} \xrightarrow{H_{hk}(\cdot)} \boxed{h}$$

have td: **easy** to find collision

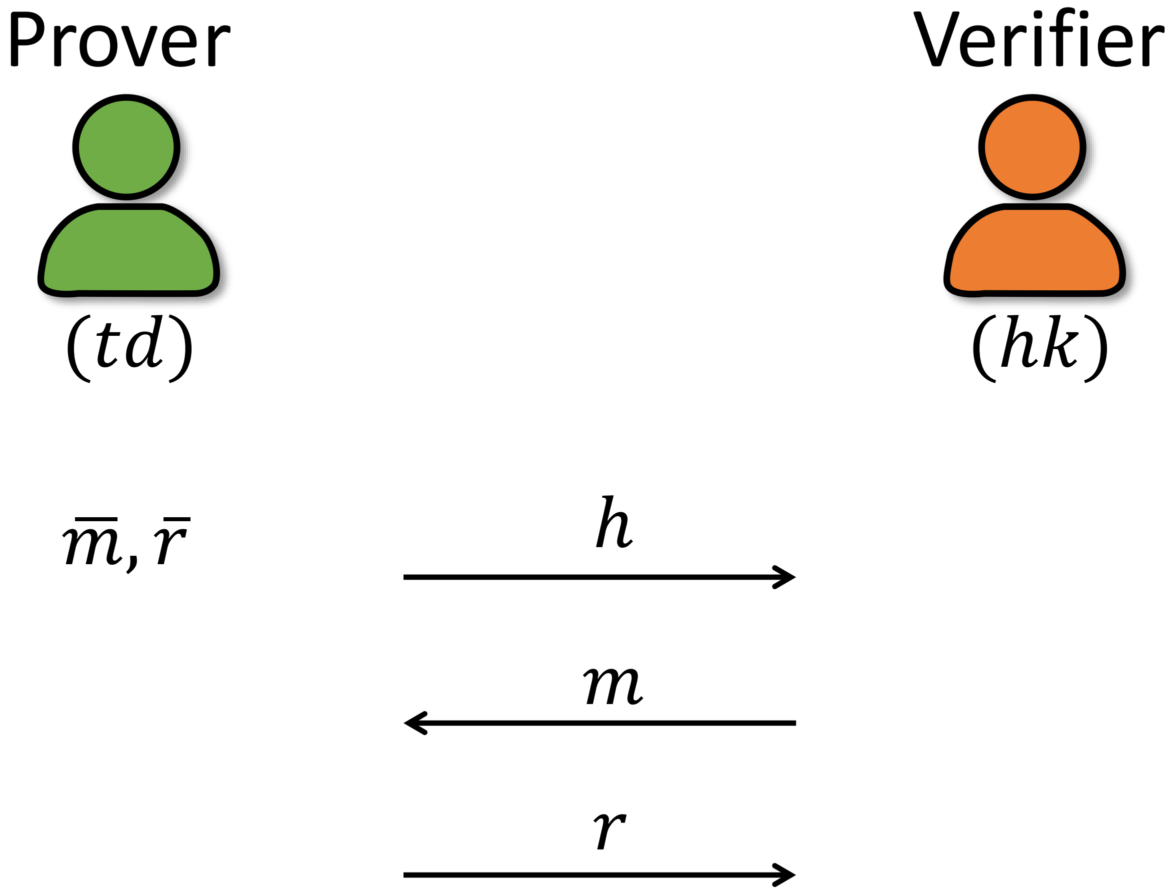
no td: **hard** to find collision

A Collision:

find $(m_1, r_1) \neq (m_2, r_2)$ s.t.

$H_{hk}(m_1, r_1) = H_{hk}(m_2, r_2)$

Chameleon Hashes → Σ-protocols → (PQ-)Signatures



Fiat-Shamir transform $(m = \widehat{H}(h, msg))$

- Turns proof into a signature
  - $Sign\_\{td\}(msg) = (h, m, r)$
  - $Ver_{hk}(h, m, r) = 1$ *iff*

    $m = \widehat{H}(h, msg)$ *and* $H_{hk}(m,r)=h$

#AMS from Chameleon Hashes

Idea: prove that among n users, there are t trapdoors (à la [CDS94])

$$\begin{cases} a_{1,1}m_1 + a_{1,2}m_2 + \ldots + a_{1,n}m_n = u_1, \\ a_{2,1}m_1 + a_{2,2}m_2 + \ldots + a_{2,n}m_n = u_2, \\ \ldots \\ a_{t,1}m_1 + a_{t,2}m_2 + \ldots + a_{t,n}m_n = u_t, \end{cases}$$

Signing Flow (yes)

P_i

$\bar{m}_i, \bar{r}_i$

$$\xrightarrow{\quad h_i \quad}$$

$$\xleftarrow{\quad m_i \quad}$$

$$\xrightarrow{\quad r_i \quad}$$

Simulated Flow (no)

$$\xrightarrow{\quad h_{-i} \quad}$$

$$\xleftarrow{\quad m_{-i} \leftarrow \$ \quad}$$

$$\xrightarrow{\quad r_{-i} \leftarrow \$ \quad}$$

$$(u_1, \ldots, u_t) \leftarrow \hat{\mathrm{H}}(msg, t, h_1, \ldots, h_n)$$

$$\sigma = (t, m_1, \ldots, m_n, r_1, \ldots, r_n)$$

Fault-Tolerant #AMS

P_1 P_2 P_3 P_4 P_5

participate in the voting

$\xleftarrow{\quad h_3 \quad}$

$\xrightarrow{\quad m_3 \quad}$

Flakers!

Cannot generate a signature normally!

$$\sigma = (t, m_1, \dots, m_n, r_1, \dots, r_n)$$

$$\sigma' = (t, F, m_1, \dots, m_n, \{r_i\}_{i \in [n] \setminus \mathrm{F}})$$

F: dropout group

Our Contributions

- Sharp anonymous multisignatures (#AMS): A primitive that natively achieves all the above properties
- Relation to Threshold Ring Signatures (TRS)
- A template for building #AMS from (Lossy) Chameleon Hashing
  - Instantiations under different assumptions yield unconditional anonymity + postquantum security
  - A generic template abstracting several known TRS schemes
- Concrete instantiations of our governance goals using #AMS
  - Interactive-Voting on multiple proposals
  - Vote-and-go approach
  - Vote on only one proposal
- As a side-product: Relation between Lossy Identification and Lossy Chameleon Hashing.

E-votin: Protocol V1

P_1 P_2 P_3 P_4 P_5

new proposal

Supporter:
send h_i

si
co
se

Problem: one more round after claiming the ballots!

compute r_i and send back

compute σ
publish on chain

posting period

declaration period

announcement period

Our Contributions

- Sharp anonymous multisignatures (#AMS): A primitive that natively achieves all the above properties
- Relation to Threshold Ring Signatures (TRS)
- A template for building #AMS from (Lossy) Chameleon Hashing
    - Instantiations under different assumptions yield unconditional anonymity + postquantum security
    - A generic template abstracting several known TRS schemes
- Concrete instantiations of our governance goals using #AMS
    - Interactive-Voting on multiple proposals
    - Vote-and-go approach
    - Vote on only one proposal
- As a side-product: Relation between Lossy Identification and Lossy Chameleon Hashing.

Protocol V2: Round Optimization

- **Goal**: Vote-and-go
- **Idea**: each voter generates a one-time (hk_i, td_i) for the voting

---

in favor: send (hk_i, td_i)

against/abstain: send hk_i only

- use (standard) signatures to ensure that hk_i was derived by user i
- use encryption to ensure that td_i is revealed to the Moderator only

Protocol V2: Round Optimization

new proposal

posting period

Problem: one voter is able to vote on many proposals!

compute σ

publish on chain

announcement period

Our Contributions

- Sharp anonymous multisignatures (#AMS): A primitive that natively achieves all the above properties
- Relation to Threshold Ring Signatures (TRS)
- A template for building #AMS from (Lossy) Chameleon Hashing
  - Instantiations under different assumptions yield unconditional anonymity + postquantum security
  - A generic template abstracting several known TRS schemes
- Concrete instantiations of our governance goals using #AMS
  - Interactive-Voting on multiple proposals
  - Vote-and-go approach
  - Vote on only one proposal
- As a side-product: Relation between Lossy Identification and Lossy Chameleon Hashing.

Protocol V3: Single Voting Setting

Single vote setting:

a voter can only cast one ballot among many candidates

Idea: user i generates different $hk_i^{(j)}$ for different proposals $IP^{(j)}$ and among $hk_i^{(1)}, \ldots, hk_i^{(j)}, \ldots$, only one trapdoor $hk_i^{(j)}$ is known to user i

$$\begin{cases} b_{1,1}hk_i^{(j_1)} + b_{1,2}hk_i^{(j_2)} + \ldots + b_{1,p}hk_i^{(j_p)} = \hat{hk}_1, \\ b_{2,1}hk_i^{(j_1)} + b_{2,2}hk_i^{(j_2)} + \ldots + b_{2,p}hk_i^{(j_p)} = \hat{hk}_2, \\ \ldots \\ b_{p-1,1}hk_i^{(j_1)} + b_{p-1,2}hk_i^{(j_2)} + \ldots + b_{p-1,p}hk_i^{(j_p)} = \hat{hk}_{p-1}, \end{cases}$$

(p the number of proposals)

$$\left(\widehat{hk}_1, \ldots, \widehat{hk}_{p-1}\right) \leftarrow \widehat{H}\left(IP^{(1)}, \ldots, IP^{(p)}, i\right)$$

Protocol V3: Single Voting Setting

new proposal

Supporter:
send (hk_i, td_i)

Non-supporter:
send hk_i

compute σ
publish on chain

posting period

declaration & signing period

announcement period

check the restriction of $\{hk_i^{(j)}\}$

Our Contributions

- Sharp anonymous multisignatures (#AMS): A primitive that natively achieves all the above properties
- Relation to Threshold Ring Signatures (TRS)
- A template for building #AMS from (Lossy) Chameleon Hashing
  - Instantiations under different assumptions yield unconditional anonymity + postquantum security
  - A generic template abstracting several known TRS schemes
- Concrete instantiations of our governance goals using #AMS
  - Interactive-Voting on multiple proposals
  - Vote-and-go approach
  - Vote on only one proposal
- As a side-product: Relation betwee
  Hashing.

Thank you!

https://eprint.iacr.org/2023/1881