A Framework for Group Action-Based Multi-Signatures and Applications to LESS, MEDS, and ALTEQ

Edoardo Signorini
Joint work with Giuseppe D'Alconzo, Andrea Flamini, and Alessio Meneghetti

May 12, 2025

Background

- New approach for digital signatures among NIST on-ramp candidates based on cryptographic group actions:
  - Code equivalence: LESS, MEDS.
  - Alternating Trilinear Form: ALTEQ.
- Many multi-signatures have been proposed for Schnorr's and lattice-based signatures.
  - Near-optimal schemes like MuSig2[1] and MuSig-L.[2]

[1] *Nick, Ruffing, and Seurin. "MuSig2: Simple Two-Round Schnorr Multi-signatures". CRYPTO 2021, Part I.*

[2] *Boschini, Takahashi, and Tibouchi. "MuSig-L: Lattice-Based Multi-signature with Single-Round Online Phase". CRYPTO 2022, Part II.*

Background

- New approach for digital signatures among NIST on-ramp candidates based on cryptographic group actions:
  - Code equivalence: LESS, MEDS.
  - Alternating Trilinear Form: ALTEQ.
- Many multi-signatures have been proposed for Schnorr's and lattice-based signatures.
  - Near-optimal schemes like MuSig2[1] and MuSig-L.[2]
- Group action-based signatures share Fiat-Shamir construction but are less structured.

Can we build (interactive) multi-signatures from cryptographic group actions?

[1] *Nick, Ruffing, and Seurin. "MuSig2: Simple Two-Round Schnorr Multi-signatures". CRYPTO 2021, Part I.*

[2] *Boschini, Takahashi, and Tibouchi. "MuSig-L: Lattice-Based Multi-signature with Single-Round Online Phase". CRYPTO 2022, Part II.*

Cryptographic Group Action (CGA)

Let G be a group, X be a set and $\star : G \times X \to X$.

$(G, X, \star)$ is a group action if $\star$ is compatible with the group operation:

- $e \star x = x$;
- $g \star (h \star x) = (gh) \star x$;

for all $g, h \in G$ and $x \in X$.

Cryptographic Group Action (CGA)

Let G be a group, X be a set and $\star : G \times X \to X$.

$(G, X, \star)$ is a group action if $\star$ is compatible with the group operation:

- $e \star x = x$;
- $g \star (h \star x) = (gh) \star x$;

for all $g, h \in G$ and $x \in X$.

Cryptographic group action means that it has interesting properties for cryptographic applications.

Effective

Polynomial time algorithms for the following:

- Operations on G.
- Computing $\star$ on almost all G, X.
- Uniformly sampling from G and X

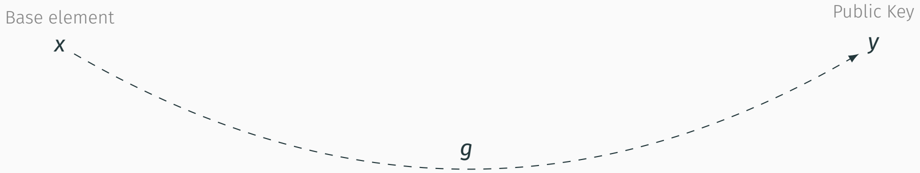
One-way (GAIP)

Given $x, y \in X$, find, if exists, $g \in G$ such that $y = g \star x$.

$$x \xrightarrow{\cancel{g}} y$$

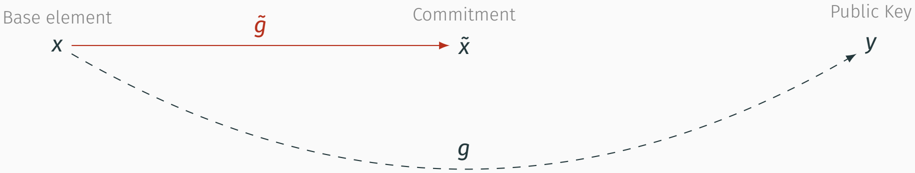
Sigma Protocol for Group Actions

Consider a cryptographic group action $(G, X, \star)$ and $x \in X$. Let $g \in G$ be the witness for the statement y with $y = g \star x$.



Sigma Protocol for Group Actions

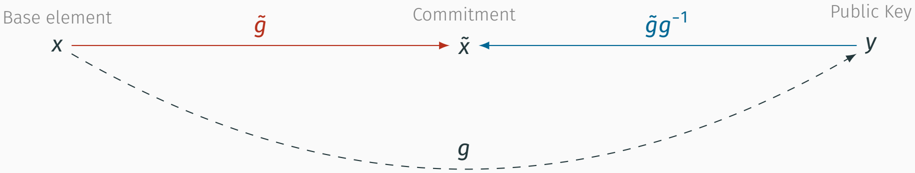
Consider a cryptographic group action $(G, X, \star)$ and $x \in X$. Let $g \in G$ be the witness for the statement y with $y = g \star x$.



- The commitment is $\tilde{g} \star x$, where $\tilde{g} \leftarrow_{\$} G$.

Sigma Protocol for Group Actions

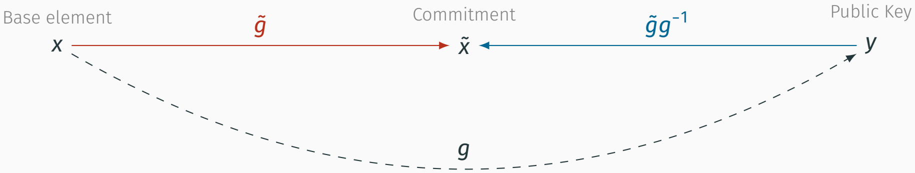
Consider a cryptographic group action $(G, X, \star)$ and $x \in X$. Let $g \in G$ be the witness for the statement y with $y = g \star x$.



- The commitment is $\tilde{g} \star x$, where $\tilde{g} \leftarrow_\$ G$.
- If $\mathsf{ch} = 0$, reveal $\mathsf{rsp} = \tilde{g}$.
- If $\mathsf{ch} = 1$, reveal $\mathsf{rsp} = \tilde{g}g^{-1}$.

Sigma Protocol for Group Actions

Consider a cryptographic group action $(G, X, \star)$ and $x \in X$. Let $g \in G$ be the witness for the statement y with $y = g \star x$.



- The commitment is $\tilde{g} \star x$, where $\tilde{g} \leftarrow_{\$} G$.
- If $\mathsf{ch} = 0$, reveal $\mathsf{rsp} = \tilde{g}$.
- If $\mathsf{ch} = 1$, reveal $\mathsf{rsp} = \tilde{g}g^{-1}$.

Digital Signature

Apply Fiat-Shamir and send = (ch, rsp).

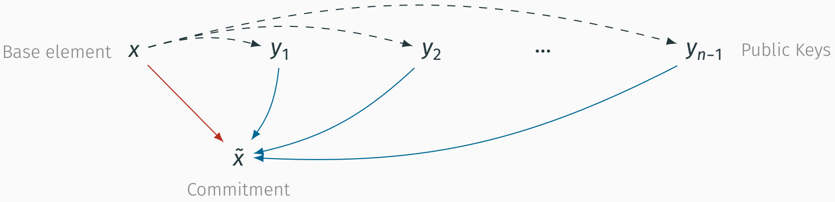
The Σ-protocol is correct, 2-special sound and HVZK if $(G, X, \star)$ is a one-way CGA.

Requires λ parallel repetitions before applying Fiat-Shamir.

A Useful Technique: Multiple Keys Optimization

The Σ-protocol from CGA is 2-special-sound

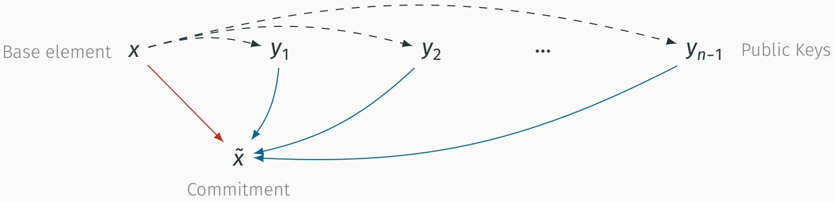
- **Base protocol**: the challenge space is $\{0, 1\} \implies$ soundness-error is $1/2$
- **Multiple public keys**: Use multiple public keys $y_1, \ldots, y_{n-1}$ and enlarge the challenge space to $\{0, \ldots, n-1\} \implies$ soundness-error is $1/n$



A Useful Technique: Multiple Keys Optimization

The Σ-protocol from CGA is 2-special-sound

- **Base protocol**: the challenge space is $\{0, 1\} \implies$ soundness-error is $1/2$
- **Multiple public keys**: Use multiple public keys $y_1, \ldots, y_{n-1}$ and enlarge the challenge space to $\{0, \ldots, n-1\} \implies$ soundness-error is $1/n$

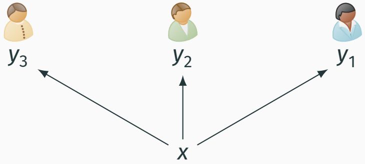


Multi-Signature Idea

Adapt the multi-public keys optimization to an interactive protocol.

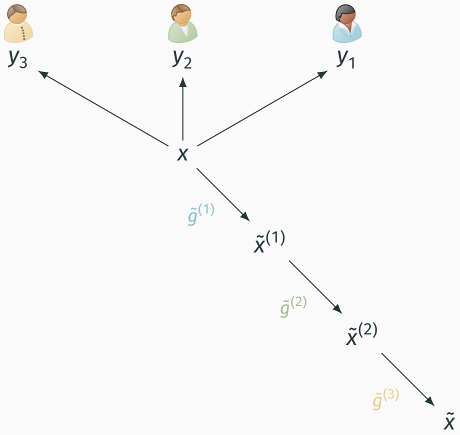
Multi-Signature from Cryptographic Group Action

- Each party P_i holds a public key $y_i = g_i \star x$.



Multi-Signature from Cryptographic Group Action

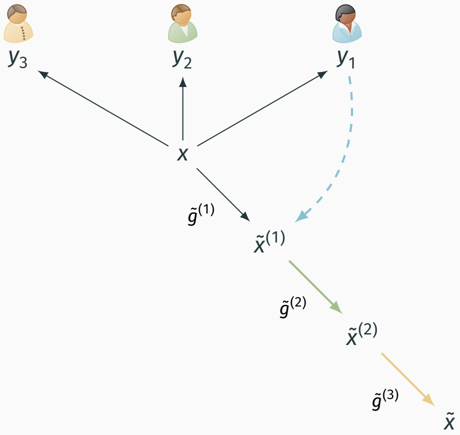
- Each party P_i holds a public key $y_i = g_i \star x$.
- Parties collaborate in a round-robin protocol to generate a common commitment $\tilde{x}$.
- Parties commit to a random salt to generate shared randomness.



Multi-Signature from Cryptographic Group Action

- Each party P_i holds a public key $y_i = g_i \star x$.
- Parties collaborate in a round-robin protocol to generate a common commitment $\tilde{x}$.
- Parties commit to a random salt to generate shared randomness.
- On challenge $\mathsf{ch} = i$, each party $P_k, k \neq i$ reveals its ephemeral group element $\tilde{g}^{(k)}$, while P_i reveals the map from y_i to $\tilde{x}^{(i)}$.
- P_i computes the response as

$$\mathsf{rsp} = \left(\prod_{k=0}^{n-1} \tilde{g}^{n-k}\right) g_i^{-1}.$$

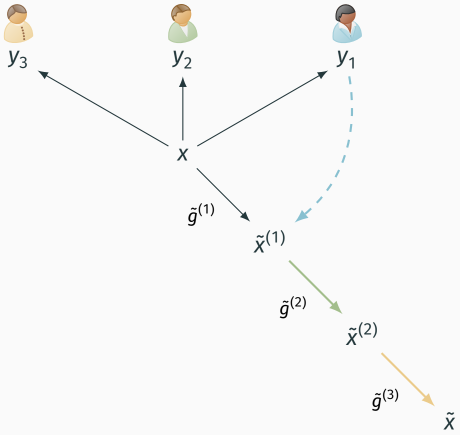


Multi-Signature from Cryptographic Group Action

- Each party P_i holds a public key $y_i = g_i \star x$.
- Parties collaborate in a round-robin protocol to generate a common commitment $\tilde{x}$.
- Parties commit to a random salt to generate shared randomness.
- On challenge $\mathsf{ch} = i$, each party $P_k, k \neq i$ reveals its ephemeral group element $\tilde{g}^{(k)}$, while P_i reveals the map from y_i to $\tilde{x}^{(i)}$.
- P_i computes the response as

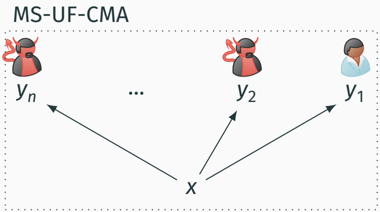
$$\mathsf{rsp} = \left(\prod_{k=0}^{n-1} \tilde{g}^{n-k}\right) g_i^{-1}.$$

- Signature = $(\mathsf{ch}, \mathsf{rsp})$ verification is identical to the underlying scheme (with different parameters).



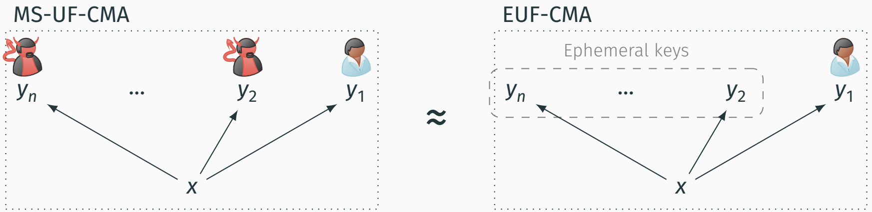
Security Proof

The adversary must forge a multi-signature involving a target user, with all other users potentially corrupted (MS-UF-CMA). The adversary can execute concurrent signing sessions.



Security Proof

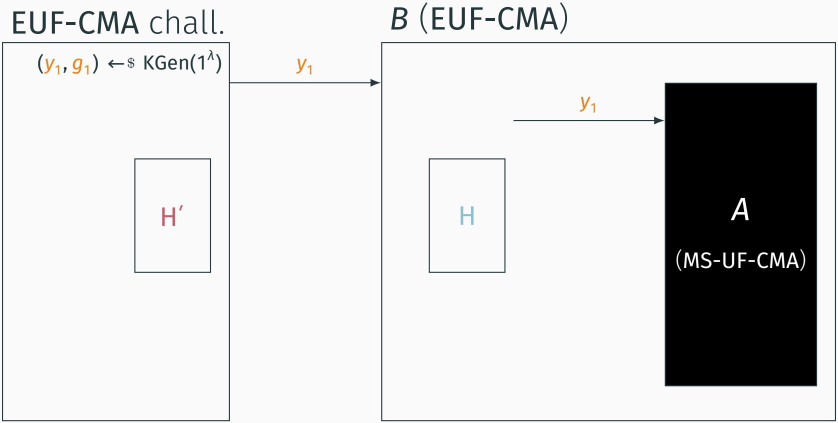
The adversary must forge a multi-signature involving a target user, with all other users potentially corrupted (**MS-UF-CMA**). The adversary can execute concurrent signing sessions.



1. **MS-UF-CMA** tightly reduces to **EUF-CMA** for a variant of the centralized signature scheme in the ROM.
2. The Σ-protocol Π' underlying the signature variant is a proof of knowledge.
3. The Fiat-Shamir transform can be applied to Π'.

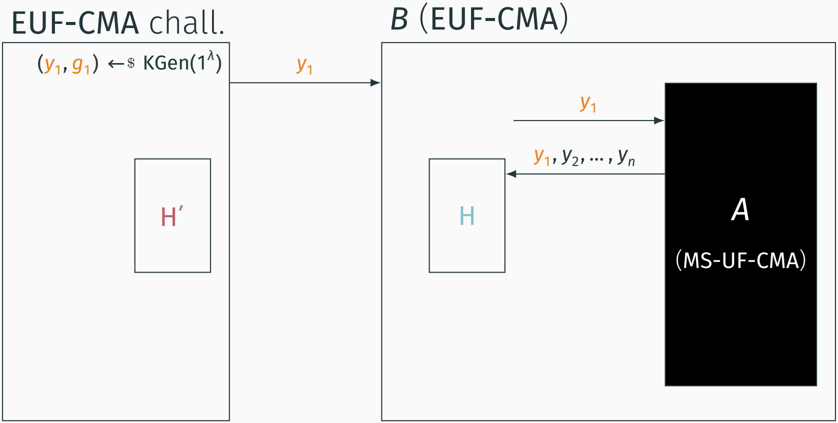
Reduction Sketch

The **EUF-CMA** adversary B is given pk = y_1 with oracle access to H' and **OSign**. Then, B forwards y_1 to the **MS-UF-CMA** adversary A and simulates H and **OMuSign**.



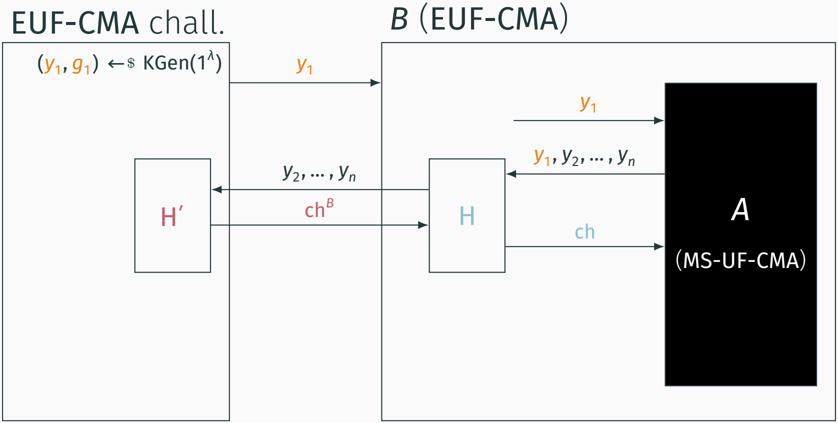
Reduction Sketch

The **EUF-CMA** adversary B is given pk = y_1 with oracle access to H′ and **OSign**. Then, B forwards y_1 to the **MS-UF-CMA** adversary A and simulates H and **OMuSign**.



Reduction Sketch

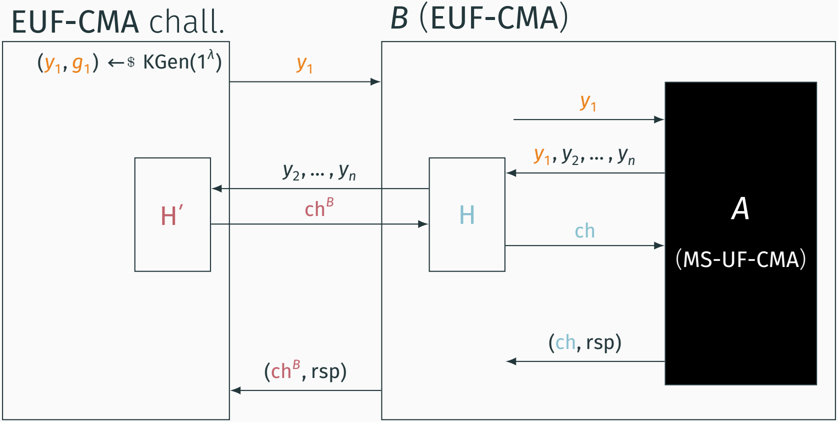
The **EUF-CMA** adversary B is given pk = y_1 with oracle access to H' and **OSign**. Then, B forwards y_1 to the **MS-UF-CMA** adversary A and simulates H and **OMuSign**.



B can program the random oracle H so that a valid response to ch can be adapted to produce a forgery for the signature with ephemeral keys.

Reduction Sketch

The **EUF-CMA** adversary B is given pk = y_1 with oracle access to H' and **OSign**. Then, B forwards y_1 to the **MS-UF-CMA** adversary A and simulates H and **OMuSign**.

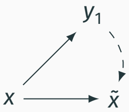


B can program the random oracle H so that a valid response to ch can be adapted to produce a forgery for the signature with ephemeral keys.

Σ-Protocol Variant with Ephemeral Keys

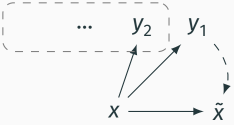
- We show that the Σ-protocol Π' is a proof of knowledge.
- Correctness and HVZK are easy, we focus on knowledge soundness.

Π: (base protocol)



2-special-sound

Π': (variable ephem. keys)

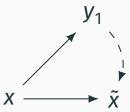


?

Σ-Protocol Variant with Ephemeral Keys

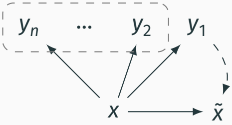
- We show that the Σ-protocol Π' is a proof of knowledge.
- Correctness and HVZK are easy, we focus on knowledge soundness.

Π: (base protocol)



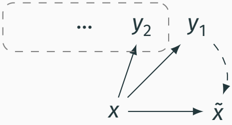
2-special-sound

$\Pi[n]$: ($n-1$ ephem. keys)



$(n+1)$-special-sound

Π': (variable ephem. keys)

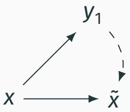


?

Σ-Protocol Variant with Ephemeral Keys

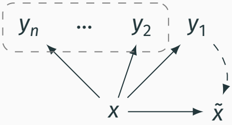
- We show that the Σ-protocol Π' is a proof of knowledge.
- Correctness and HVZK are easy, we focus on knowledge soundness.

Π: (base protocol)



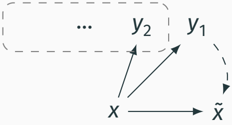
2-special-sound

$\Pi[n]$: ($n - 1$ ephem. keys)



$(n + 1)$-special-sound

Π': (variable ephem. keys)



?

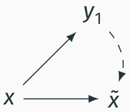
Custom Extractor

- Each dishonest (deterministic) prover P^* attacking Π' can be used to build a (probabilistic) prover P_n against $\Pi[n]$.
- The success probability of P_n is the same as for P^*.
- Use the extractor for $\Pi[n]$ to extract a witness from P_n.

Σ-Protocol Variant with Ephemeral Keys

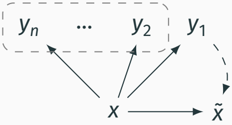
- We show that the Σ-protocol Π' is a proof of knowledge.
- Correctness and HVZK are easy, we focus on knowledge soundness.

Π: (base protocol)



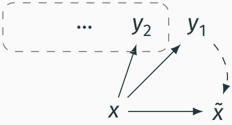
2-special-sound

$\Pi[n]$: ($n - 1$ ephem. keys)



($n + 1$)-special-sound

Π': (variable ephem. keys)



?

Custom Extractor

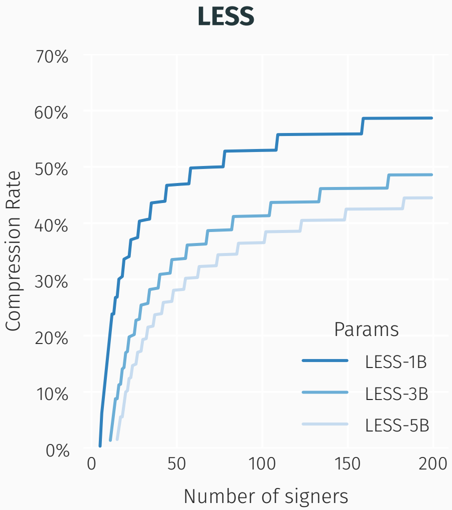
- Each dishonest (deterministic) prover P^* attacking Π' can be used to build a (probabilistic) prover P_n against $\Pi[n]$.
- The success probability of P_n is the same as for P^*.
- Use the extractor for $\Pi[n]$ to extract a witness from P_n.

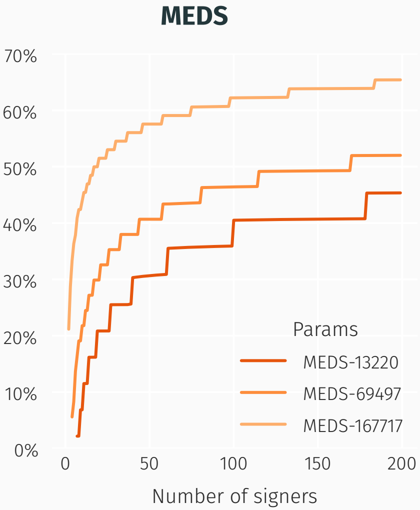
Fiat-Shamir can be applied by employing multiple random oracles via Random Oracle Cloning.[3]

[3] *Bellare, Davis, and Günther. "Separate Your Domains: NIST PQC KEMs, Oracle Cloning and Read-Only Indifferentiability". EUROCRYPT 2020, Part II.*

Benchmarking

Applicable to group action-based signature schemes (e.g., LESS, MEDS, ALTEQ)





Conclusions

Feasibility of multi-signature scheme for unstructured group-action signatures.

- Three round complexity (two round-robin and one broadcast).
- Secure in the plain public-key model (no custom key generation required).
- Reduce to the Group Action Inverse Problem in the classical ROM.

Open Questions:

- Reduce round complexity by removing the initial commitment round.
- Key Aggregation and constant size signature.
- Proof in the QROM.

Thank You!