How to Prove Statements Obliviously?

Sanjam Garg

UC Berkeley

Aarushi Goel

NTT Research –> Purdue

Mingyuan Wang

UC Berkeley –> NYU Shanghai

Aug. 2024 @ CRYPTO

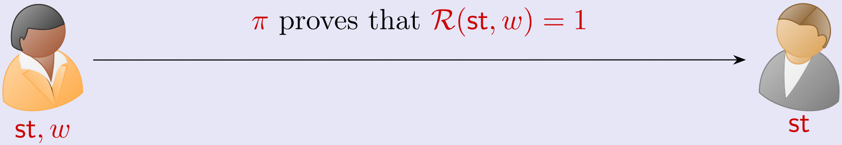
ia.cr/2023/1609

Succinct Non-interactive Arguments (SNARG)

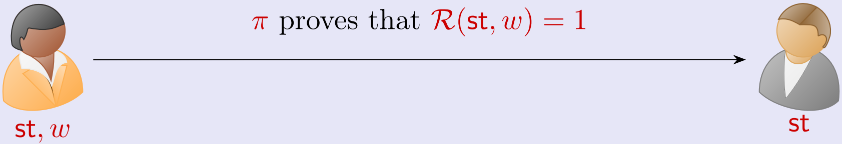
st, w

st

Succinct Non-interactive Arguments (SNARG)

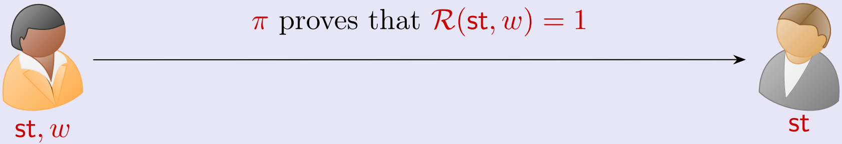


Succinct Non-interactive Arguments (SNARG)



- Completeness, Soundness.

Succinct Non-interactive Arguments (SNARG)



- Completeness, Soundness.
- Succinctness: Verification time and proof size $|\pi|$ are $\ll |w|$

Succinct Non-interactive Arguments (SNARG)

π proves that $\mathcal{R}(\mathsf{st}, w) = 1$

st, w

st

- Completeness, Soundness.
- Succinctness: Verification time and proof size $|\pi|$ are $\ll |w|$
- A large body of works — [Kil92, Mic94, Gro16, BCG+17, BCG+18, XZZ+19, GWC19, Set20, BCG20, CHM+20, Lee21, KMP20, ZLW+21, BCL22 ......]

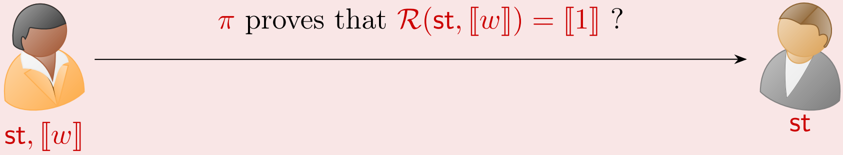
Oblivious Proof Generation

π proves that $\mathcal{R}(\mathsf{st}, [\![w]\!]) = [\![1]\!]$?

$\mathsf{st}, [\![w]\!]$ $\qquad\qquad$ st

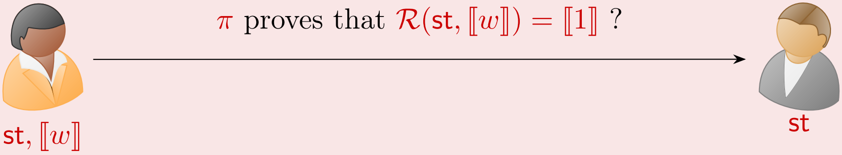
- What if the prover does not hold w in the clear? Only an encapsulation $[\![w]\!]$ of w.

Oblivious Proof Generation



- What if the prover does not hold w in the clear? Only an encapsulation $[\![w]\!]$ of w.
  - $[\![w]\!]$: encryption, commitment, g^w, ...

Oblivious Proof Generation



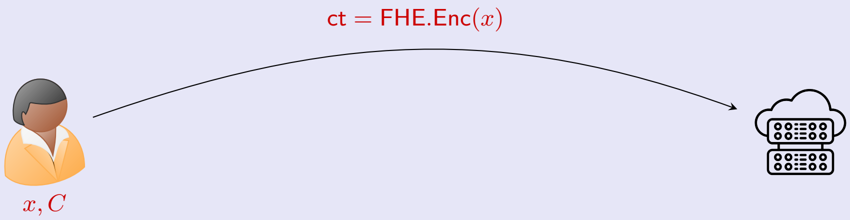
- What if the prover does not hold w in the clear? Only an encapsulation $[\![w]\!]$ of w.
  - $[\![w]\!]$: encryption, commitment, g^w, ...

How can the prover generate a proof obliviously?

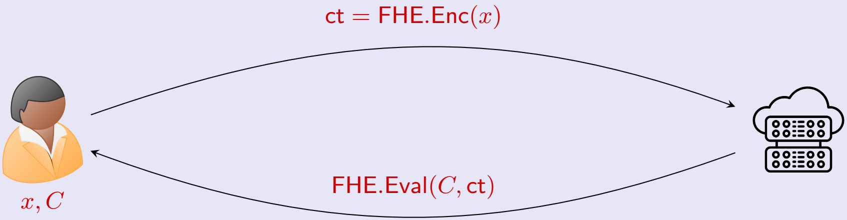
Example I: Verifiable Delegation of Computation

x, C

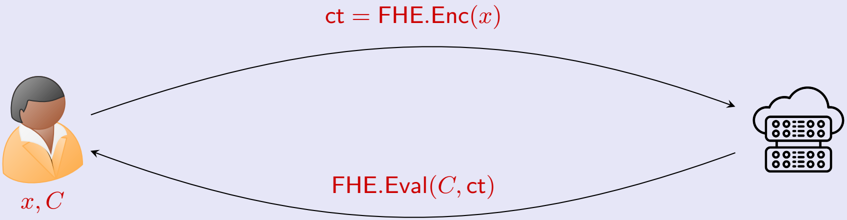
Example I: Verifiable Delegation of Computation



Example I: Verifiable Delegation of Computation

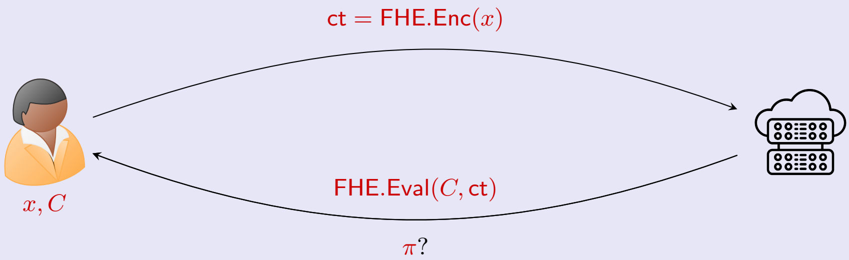


Example I: Verifiable Delegation of Computation



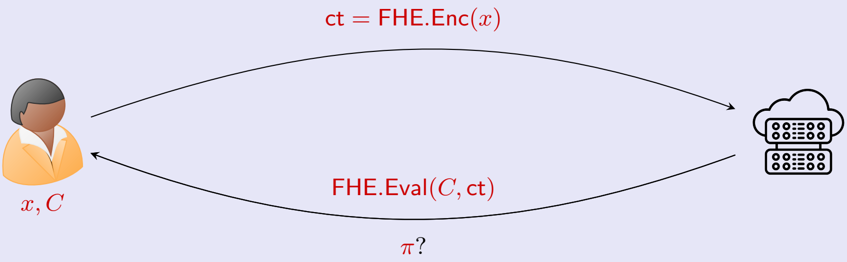
- FHE provides no integrity

Example I: Verifiable Delegation of Computation



- FHE provides no integrity
- Needs to prove the honest performance of $\mathsf{FHE.Eval}$

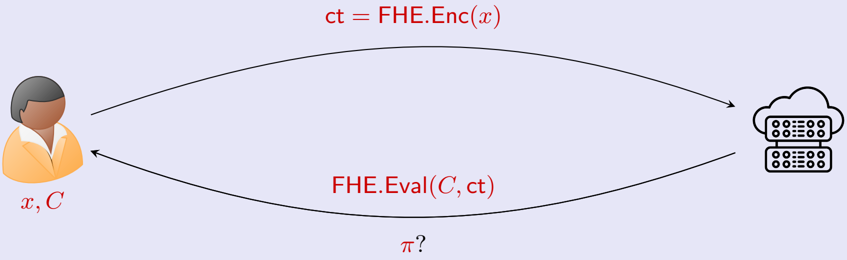
Example I: Verifiable Delegation of Computation



- FHE provides no integrity
- Needs to prove the honest performance of $\mathsf{FHE.Eval}$
- Oblivious proving

$$C(\llbracket x \rrbracket) = \llbracket C(x) \rrbracket$$

Example I: Verifiable Delegation of Computation



- FHE provides no integrity
- Needs to prove the honest performance of $\mathsf{FHE.Eval}$
- Oblivious proving

$$C(\llbracket x \rrbracket) = \llbracket C(x) \rrbracket$$

- Private verifiability is acceptable

Example II: Verifiable Public Keys Aggregation

$\mathsf{pk}_1, \ldots, \mathsf{pk}_n$

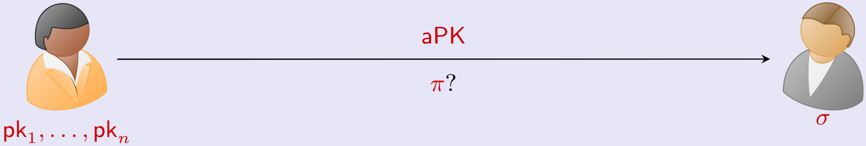
σ

- $\mathsf{pk}_i = g^{\mathsf{sk}_i}$
- σ succinct commitment of all pk_i

Example II: Verifiable Public Keys Aggregation

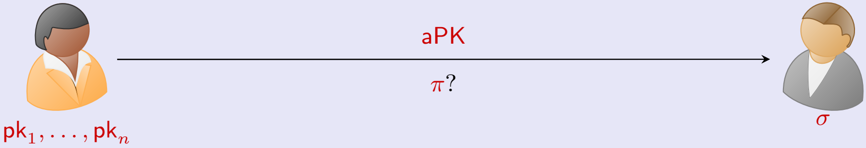
- $\mathsf{pk}_i = g^{\mathsf{sk}_i}$
- σ succinct commitment of all pk_i

Example II: Verifiable Public Keys Aggregation



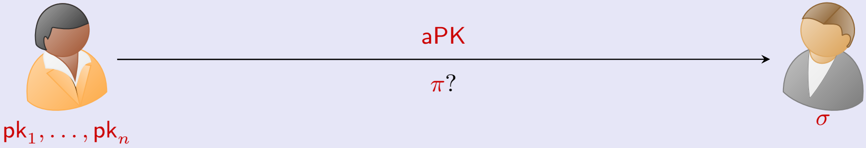
- $\mathsf{pk}_i = g^{\mathsf{sk}_i}$
- σ succinct commitment of all pk_i
- Prove aPK is an aggregation of $\geqslant T$ keys

Example II: Verifiable Public Keys Aggregation



- $\mathsf{pk}_i = g^{\mathsf{sk}_i}$
- σ succinct commitment of all pk_i
- Prove aPK is an aggregation of $\geqslant T$ keys
- Useful in threshold signatures setting [GJMSWZ'24, DCXNBR'23]

Example II: Verifiable Public Keys Aggregation

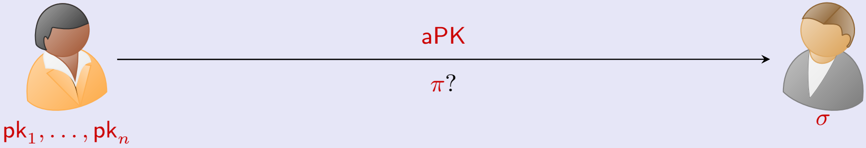


- $\mathsf{pk}_i = g^{\mathsf{sk}_i}$
- σ succinct commitment of all pk_i
- Prove aPK is an aggregation of $\geqslant T$ keys
- Useful in threshold signatures setting [GJMSWZ'24, DCXNBR'23]
- Oblivious proof in disguise!

$$\mathsf{pk}_i = [\![\mathsf{sk}_i]\!]$$

$$\textstyle\prod_i \mathsf{pk}_i = \mathsf{aPK} \qquad \Longrightarrow \qquad \sum_i [\![\mathsf{sk}_i]\!] = [\![\mathsf{aSK}]\!]$$

Example II: Verifiable Public Keys Aggregation



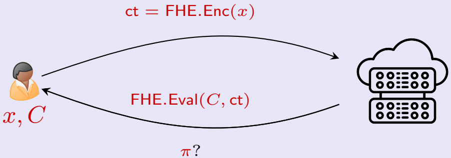
- $\mathsf{pk}_i = g^{\mathsf{sk}_i}$
- σ succinct commitment of all pk_i
- Prove aPK is an aggregation of $\geqslant T$ keys
- Useful in threshold signatures setting [GJMSWZ'24, DCXNBR'23]
- Oblivious proof in disguise!

$$\mathsf{pk}_i = [\![\mathsf{sk}_i]\!]$$

$$\prod_i \mathsf{pk}_i = \mathsf{aPK} \qquad \Longrightarrow \qquad \sum_i [\![\mathsf{sk}_i]\!] = [\![\mathsf{aSK}]\!]$$

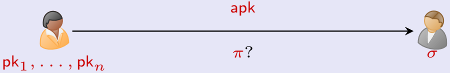
- Need Public verifiability

Example I: Verifiable Delegation of Computation



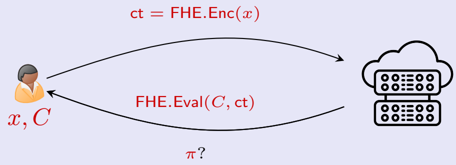
- $C([\![x]\!]) = [\![C(x)]\!]$
- Private verifiability

Example II: Verifiable Public Keys Aggregation



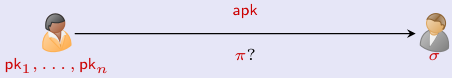
- $\sum_i [\![\mathsf{sk}_i]\!] = [\![\mathsf{aSK}]\!]$
- Public verifiability

Example I: Verifiable Delegation of Computation



- $C([\![x]\!]) = [\![C(x)]\!]$
- Private verifiability

Example II: Verifiable Public Keys Aggregation

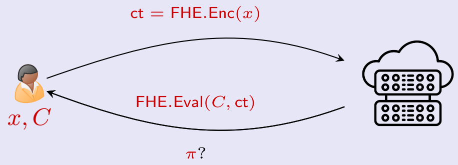


- $\sum_i [\![\mathsf{sk}_i]\!] = [\![\mathsf{aSK}]\!]$
- Public verifiability

A Trivial Solution

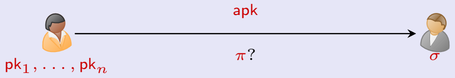
- Translate homomorphic operations (FHE.Eval, group operation) as circuits and generically apply SNARKs

Example I: Verifiable Delegation of Computation



- $C(\llbracket x \rrbracket) = \llbracket C(x) \rrbracket$
- Private verifiability

Example II: Verifiable Public Keys Aggregation

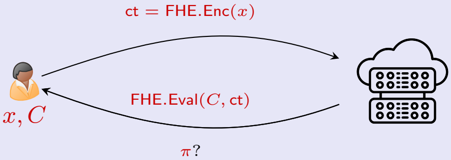


- $\sum_i \llbracket \mathsf{sk}_i \rrbracket = \llbracket \mathsf{aSK} \rrbracket$
- Public verifiability

A Trivial Solution

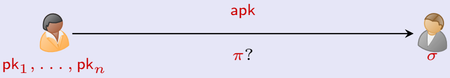
- Translate homomorphic operations (FHE.Eval, group operation) as circuits and generically apply SNARKs
- Non-black-box, highly inefficient

Example I: Verifiable Delegation of Computation



- $C(\llbracket x \rrbracket) = \llbracket C(x) \rrbracket$
- Private verifiability

Example II: Verifiable Public Keys Aggregation



- $\sum_i \llbracket \mathsf{sk}_i \rrbracket = \llbracket \mathsf{aSK} \rrbracket$
- Public verifiability

A Trivial Solution

- Translate homomorphic operations (FHE.Eval, group operation) as circuits and generically apply SNARKs
- Non-black-box, highly inefficient

How can the prover generate a proof obliviously with only black-box use *of the encapsulation scheme?*

Our Results

A General Technique

FRI on hidden values enables an oblivious polynomial commitment scheme:

Our Results

A General Technique

FRI on hidden values enables an oblivious polynomial commitment scheme:

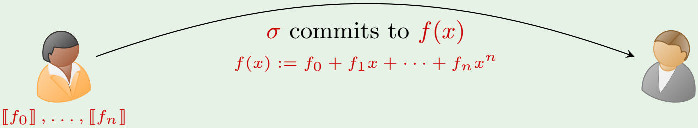
$[\![f_0]\!], \ldots, [\![f_n]\!]$

- Assuming $[\![\cdot]\!]$ supports linear homomorphism

Our Results

A General Technique

FRI on hidden values enables an oblivious polynomial commitment scheme:

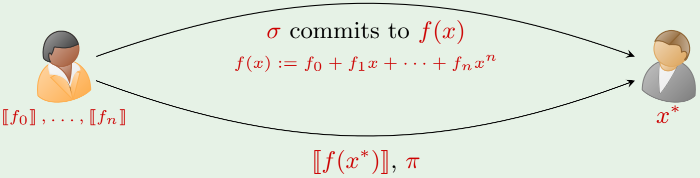


- Assuming $[\![\cdot]\!]$ supports linear homomorphism

Our Results

A General Technique

FRI on hidden values enables an oblivious polynomial commitment scheme:

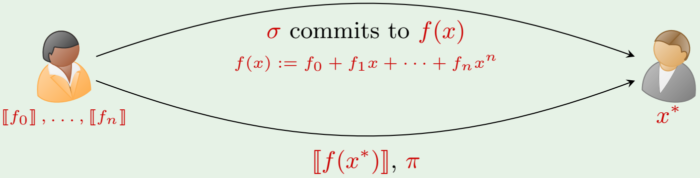


- Assuming $[\![\cdot]\!]$ supports linear homomorphism

Our Results

A General Technique

FRI on hidden values enables an oblivious polynomial commitment scheme:

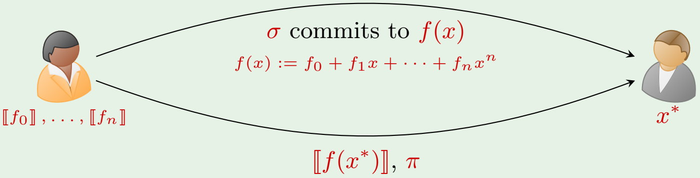


- Assuming $[\![\cdot]\!]$ supports linear homomorphism
- Private Verifiable if $[\![\cdot]\!]$ is decryptable (e.g., FHE)

Our Results

A General Technique

FRI on hidden values enables an oblivious polynomial commitment scheme:

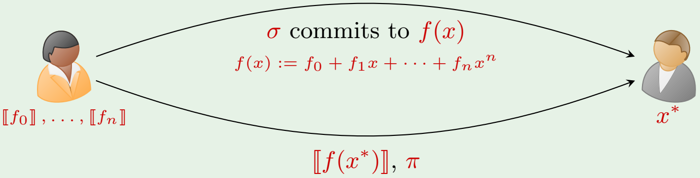


- Assuming $[\![\cdot]\!]$ supports linear homomorphism
- Private Verifiable if $[\![\cdot]\!]$ is decryptable (e.g., FHE)
- Public verifiable if $[\![\cdot]\!]$ is linear homomorphic in randomness ($[\![x; r_1]\!] + [\![y; r_2]\!] = [\![x + y; r_1 + r_2]\!]$)
    - E.g., group exponentiation, ElGamal, ...

Our Results

A General Technique

FRI on hidden values enables an oblivious polynomial commitment scheme:



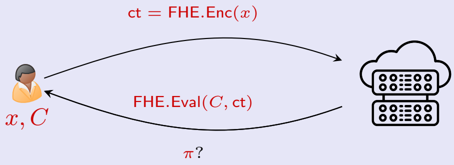
Prover $O(n \log^2 n)$

$|\sigma| = O(1)$

$|\pi| = O(\log^2 n)$

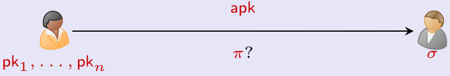
- Assuming $[\![\cdot]\!]$ supports linear homomorphism
- Private Verifiable if $[\![\cdot]\!]$ is decryptable (e.g., FHE)
- Public verifiable if $[\![\cdot]\!]$ is linear homomorphic in randomness ($[\![x; r_1]\!] + [\![y; r_2]\!] = [\![x + y; r_1 + r_2]\!]$)
  - E.g., group exponentiation, ElGamal, ...
- An adaptation of the celebrated FRI proof system [Ben-Sasson-Bentov-Horesh-Riabzev'18]
  Black-box in $[\![\cdot]\!]$ and achieve the same efficiency as FRI

Example I: Verifiable Delegation of Computation



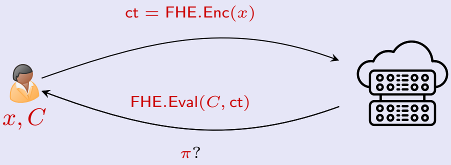
- $C(\llbracket x \rrbracket) = \llbracket C(x) \rrbracket$
- Private verifiability

Example II: Verifiable Public Keys Aggregation



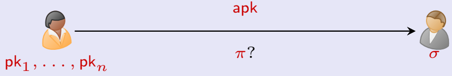
- $\sum_i \llbracket \mathsf{sk}_i \rrbracket = \llbracket \mathsf{aSK} \rrbracket$
- Public verifiability

Example I: Verifiable Delegation of Computation



- $C(\llbracket x \rrbracket) = \llbracket C(x) \rrbracket$
- Private verifiability

Example II: Verifiable Public Keys Aggregation

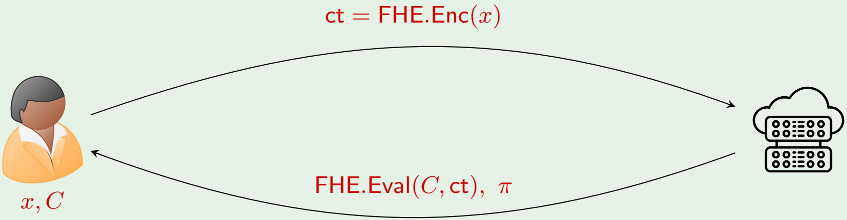


- $\sum_i \llbracket \mathsf{sk}_i \rrbracket = \llbracket \mathsf{aSK} \rrbracket$
- Public verifiability

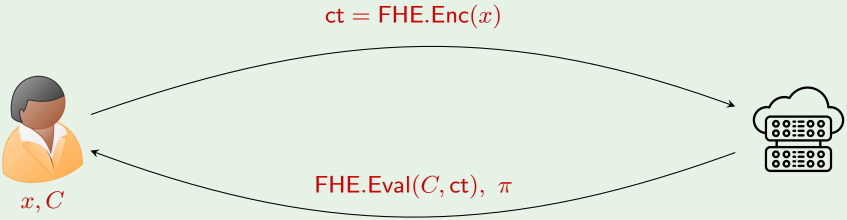
Our Results

- Existing techniques: polynomial commitment $\Longrightarrow$ SNARKs
- We show: FRI on hidden values $\Longrightarrow$ oblivious (black-box) proof generation

Application: Verifiable Delegation of Computation

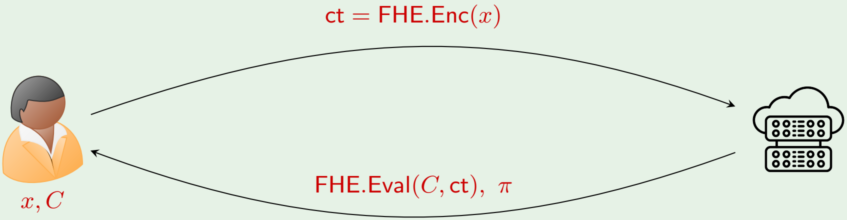


Application: Verifiable Delegation of Computation



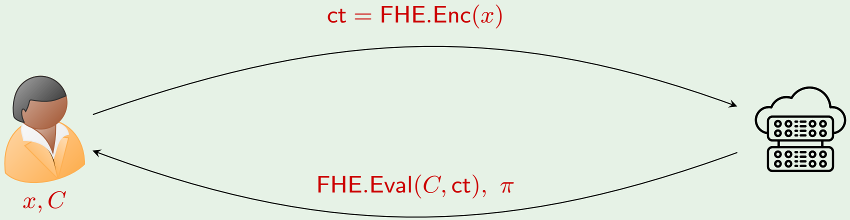
- Prover Efficiency: $|C| \log |C|$ (black-box) FHE.Eval Operations
- Proof Size: $|\pi| = O(\log^2 |C|)$

Application: Verifiable Delegation of Computation



- Prover Efficiency: $|C| \log |C|$ (black-box) FHE.Eval Operations
- Proof Size: $|\pi| = O(\log^2 |C|)$
- Private Verification: $O(\log^2 |C|)$ FHE operations

Application: Verifiable Delegation of Computation

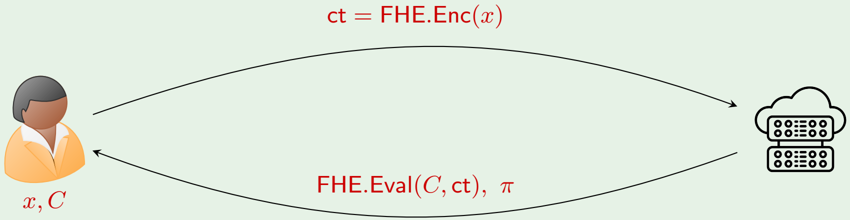


- Prover Efficiency: $|C| \log |C|$ (black-box) FHE.Eval Operations
- Proof Size: $|\pi| = O(\log^2 |C|)$
- Private Verification: $O(\log^2 |C|)$ FHE operations

Prior Works

- Require the client to perform $|C|$ FHE operations [Gennaro-Gentry-Parno'10, Applebaum-Ishai-Kushilevitz'10, Chung-Kalai-Vadhan'10, Benabbas-Gennaro-Vahlis'11, ...]

Application: Verifiable Delegation of Computation

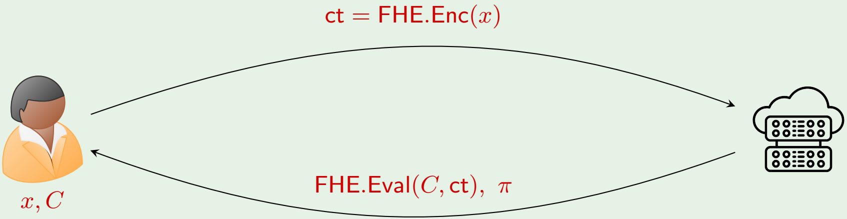


- Prover Efficiency: $|C| \log |C|$ (black-box) FHE.Eval Operations
- Proof Size: $|\pi| = O(\log^2 |C|)$
- Private Verification: $O(\log^2 |C|)$ FHE operations

Prior Works

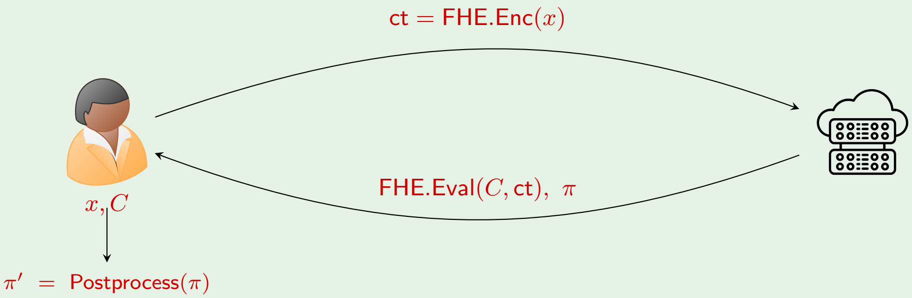
- Require the client to perform $|C|$ FHE operations [Gennaro-Gentry-Parno'10, Applebaum-Ishai-Kushilevitz'10, Chung-Kalai-Vadhan'10, Benabbas-Gennaro-Vahlis'11, ...]
- Makes non-black-box use of FHE [Fiore-Gennaro-Pastr'14, Fiore-Nitulescu-Pointcheval'20, Bois-Cascudo-Fiore-Kim'21, ...]

Application: Verifiable Delegation of Computation



- Server Efficiency: $|C| \log |C|$ (black-box) FHE.Eval Operations
- Proof Size: $|\pi| = O(\log^2 |C|)$

Application: Verifiable Delegation of Computation

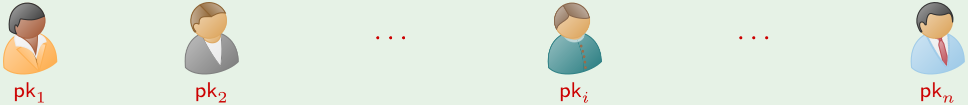


- Server Efficiency: $|C| \log |C|$ (black-box) FHE.Eval Operations
- Proof Size: $|\pi| = O(\log^2 |C|)$

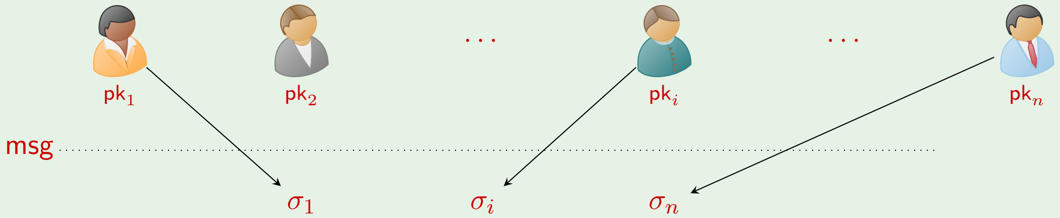
Public Verifiability

- Client Postprocessing: $O(\log^2 |C|)$
- Application to the delegation of zkSNARKs to untrusted server (see paper!)

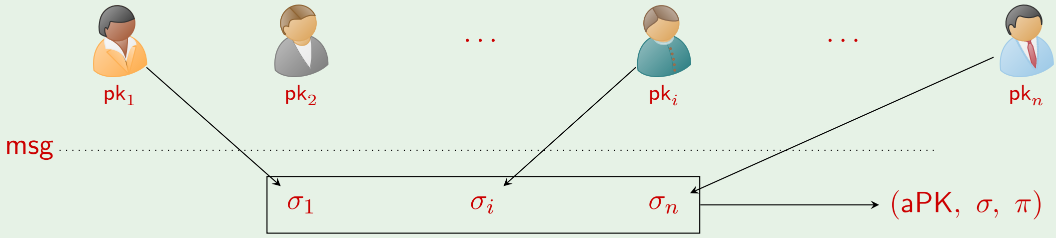
Application: Weighted Threshold Signature without DKG



Application: Weighted Threshold Signature without DKG

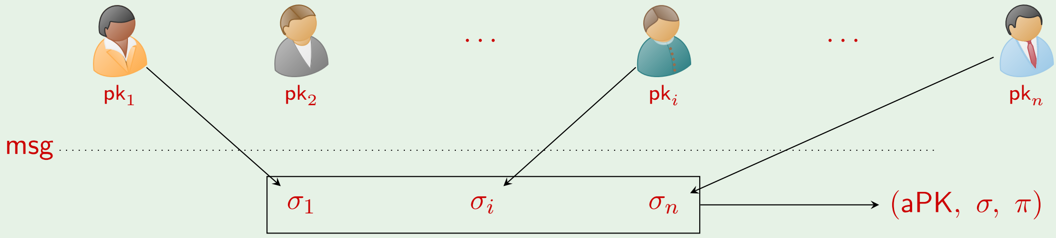


Application: Weighted Threshold Signature without DKG



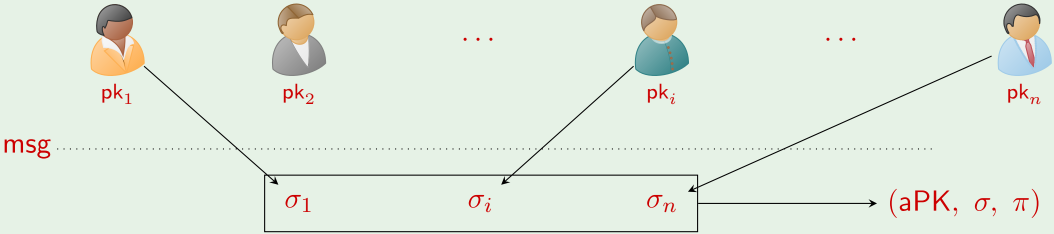
- Works for signatures with linearly-aggregatable public keys (BLS, Schnorr)

Application: Weighted Threshold Signature without DKG



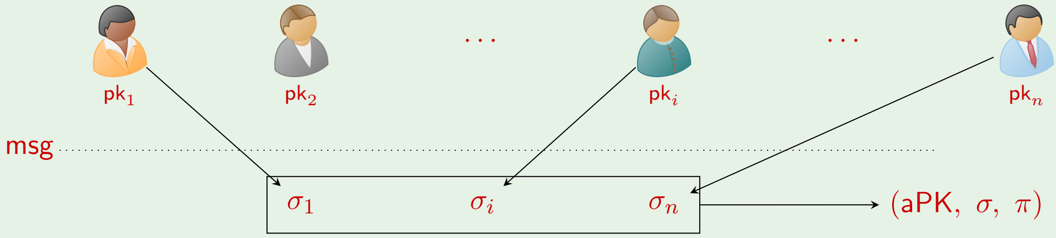
- Works for signatures with linearly-aggregatable public keys (BLS, Schnorr)
- π proves aPK is aggregation of $\geqslant T$ public keys

Application: Weighted Threshold Signature without DKG



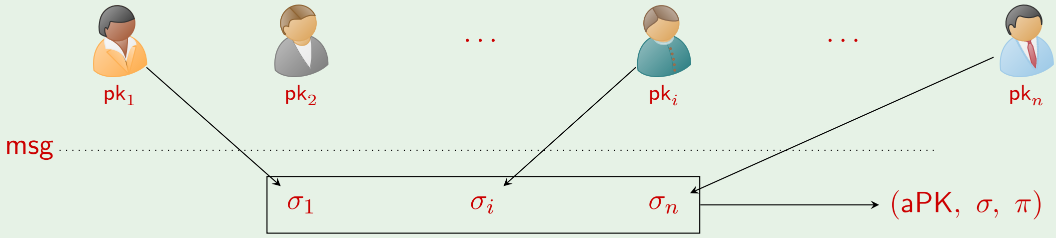
- Works for signatures with linearly-aggregatable public keys (BLS, Schnorr)
- π proves aPK is aggregation of $\geqslant T$ public keys
- Extends to weighted setting <u>without</u> efficiency degradation

Application: Weighted Threshold Signature without DKG



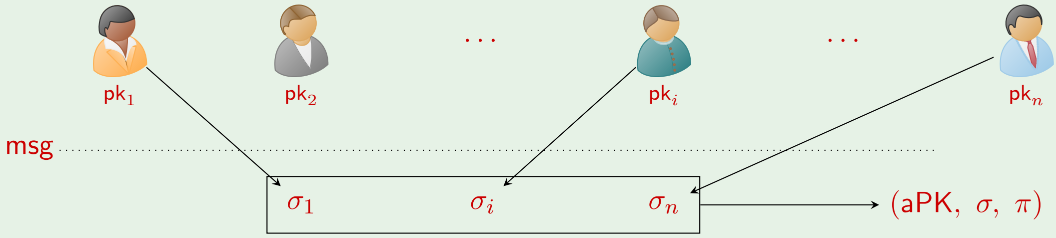
- Works for signatures with linearly-aggregatable public keys (BLS, Schnorr)
- π proves aPK is aggregation of $\geqslant T$ public keys
- Extends to weighted setting without efficiency degradation
- Can be extended to arbitrary access structure $C : 2^{[n]} \to \{0, 1\}$.

Application: Weighted Threshold Signature without DKG



- Works for signatures with linearly-aggregatable public keys (BLS, Schnorr)
- π proves aPK is aggregation of $\geqslant T$ public keys
- Extends to weighted setting without efficiency degradation
- Can be extended to arbitrary access structure $C : 2^{[n]} \to \{0,1\}$.
- No distributed key generation (DKG)

Application: Weighted Threshold Signature without DKG

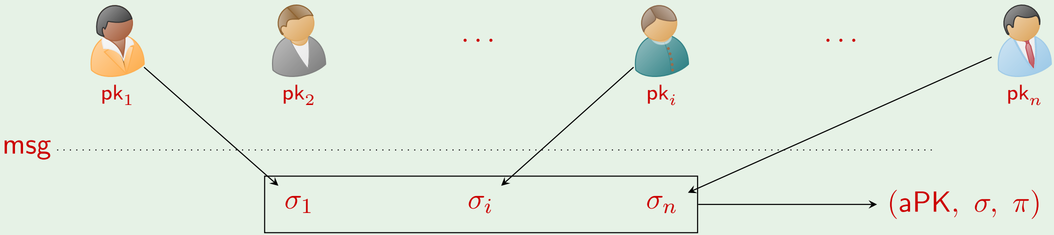


- Works for signatures with linearly-aggregatable public keys (BLS, Schnorr)
- π proves aPK is aggregation of $\geqslant T$ public keys
- Extends to weighted setting without efficiency degradation
- Can be extended to arbitrary access structure $C : 2^{[n]} \to \{0,1\}$.
- No distributed key generation (DKG)

Prior Works

- Pairing-based SNARKs: $O(n)$-size SRS, $O(n)$-size pk_i [Garg-Jain-Mukherjee-Sinha-W-Zhang'24, Das-Camacho-Xiang-Nieto-Bunz-Ren'23]

Application: Weighted Threshold Signature without DKG



- Works for signatures with linearly-aggregatable public keys (BLS, Schnorr)
- π proves aPK is aggregation of $\geqslant T$ public keys
- Extends to weighted setting without efficiency degradation
- Can be extended to arbitrary access structure $C : 2^{[n]} \to \{0,1\}$.
- No distributed key generation (DKG)

Prior Works

- Pairing-based SNARKs: $O(n)$-size SRS, $O(n)$-size pk$_i$ [Garg-Jain-Mukherjee-Sinha-W-Zhang'24, Das-Camacho-Xiang-Nieto-Bunz-Ren'23]
- Require Ramp: [Micali-Reyzin-Vlachos-Wahby-Zeldovich'21]

Technical Highlights

Prover wants to prove $C(\llbracket x \rrbracket) = \llbracket y \rrbracket$, where $\llbracket \cdot \rrbracket$ supports some homomorphism.

SNARK on top of Homomorphism

Homomorphism on top of SNARK

Prover wants to prove $C(\llbracket x \rrbracket) = \llbracket y \rrbracket$, where $\llbracket \cdot \rrbracket$ supports some homomorphism.

SNARK on top of Homomorphism

$C(\llbracket x \rrbracket) = \llbracket y \rrbracket$ is a public relation

Homomorphism on top of SNARK

$C(x) = y$ is a private relation. (Even the verifier may not know this statement)

Prover wants to prove $C(\llbracket x \rrbracket) = \llbracket y \rrbracket$, where $\llbracket \cdot \rrbracket$ supports some homomorphism.

SNARK on top of Homomorphism

$C(\llbracket x \rrbracket) = \llbracket y \rrbracket$ is a public relation

Homomorphic Evaluation
on $\llbracket x \rrbracket$ to get $\llbracket y \rrbracket$

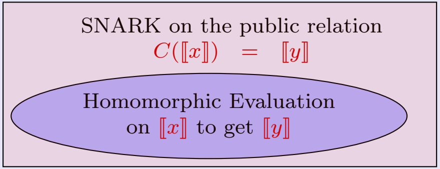
Homomorphism on top of SNARK

$C(x) = y$ is a private relation. (Even the verifier may not know this statement)

Prover wants to prove $C(\llbracket x \rrbracket) = \llbracket y \rrbracket$, where $\llbracket \cdot \rrbracket$ supports some homomorphism.

SNARK on top of Homomorphism

$C(\llbracket x \rrbracket) = \llbracket y \rrbracket$ is a public relation



Homomorphism on top of SNARK

$C(x) = y$ is a private relation. (Even the verifier may not know this statement)

Prover wants to prove $C(\llbracket x \rrbracket) = \llbracket y \rrbracket$, where $\llbracket \cdot \rrbracket$ supports some homomorphism.

SNARK on top of Homomorphism

$C(\llbracket x \rrbracket) = \llbracket y \rrbracket$ is a public relation

SNARK on the public relation
$C(\llbracket x \rrbracket) \quad = \quad \llbracket y \rrbracket$

Homomorphic Evaluation
on $\llbracket x \rrbracket$ to get $\llbracket y \rrbracket$

- Non-black-box

Homomorphism on top of SNARK

$C(x) = y$ is a private relation. (Even the verifier may not know this statement)

Prover wants to prove $C(\llbracket x \rrbracket) = \llbracket y \rrbracket$, where $\llbracket \cdot \rrbracket$ supports some homomorphism.

SNARK on top of Homomorphism

$C(\llbracket x \rrbracket) = \llbracket y \rrbracket$ is a public relation

SNARK on the public relation
$C(\llbracket x \rrbracket) \quad = \quad \llbracket y \rrbracket$

Homomorphic Evaluation
on $\llbracket x \rrbracket$ to get $\llbracket y \rrbracket$

- Non-black-box

Homomorphism on top of SNARK

$C(x) = y$ is a private relation. (Even the verifier may not know this statement)

SNARK on the private relation
$C(x) \quad = \quad y$

Prover wants to prove $C(\llbracket x \rrbracket) = \llbracket y \rrbracket$, where $\llbracket \cdot \rrbracket$ supports some homomorphism.

SNARK on top of Homomorphism

$C(\llbracket x \rrbracket) = \llbracket y \rrbracket$ is a public relation

SNARK on the public relation
$C(\llbracket x \rrbracket) \quad = \quad \llbracket y \rrbracket$

Homomorphic Evaluation
on $\llbracket x \rrbracket$ to get $\llbracket y \rrbracket$

- Non-black-box

Homomorphism on top of SNARK

$C(x) = y$ is a private relation. (Even the verifier may not know this statement)

Homomorphic Evaluation on $\llbracket x \rrbracket$
to generate the SNARK proof

SNARK on the private relation
$C(x) \quad = \quad y$

Prover wants to prove $C(\llbracket x \rrbracket) = \llbracket y \rrbracket$, where $\llbracket \cdot \rrbracket$ supports some homomorphism.

SNARK on top of Homomorphism

$C(\llbracket x \rrbracket) = \llbracket y \rrbracket$ is a public relation

SNARK on the public relation
$C(\llbracket x \rrbracket) \quad = \quad \llbracket y \rrbracket$

Homomorphic Evaluation
on $\llbracket x \rrbracket$ to get $\llbracket y \rrbracket$

- Non-black-box

Homomorphism on top of SNARK

$C(x) = y$ is a private relation. (Even the verifier may not know this statement)

Homomorphic Evaluation on $\llbracket x \rrbracket$
to generate the SNARK proof

SNARK on the private relation
$C(x) \quad = \quad y$

- Black-box

Prover wants to prove $C(\llbracket x \rrbracket) = \llbracket y \rrbracket$, where $\llbracket \cdot \rrbracket$ supports some homomorphism.

SNARK on top of Homomorphism

$C(\llbracket x \rrbracket) = \llbracket y \rrbracket$ is a public relation

SNARK on the public relation
$C(\llbracket x \rrbracket) \quad = \quad \llbracket y \rrbracket$

Homomorphic Evaluation
on $\llbracket x \rrbracket$ to get $\llbracket y \rrbracket$

- Non-black-box

Homomorphism on top of SNARK

$C(x) = y$ is a private relation. (Even the verifier may not know this statement)

Homomorphic Evaluation on $\llbracket x \rrbracket$
to generate the SNARK proof

SNARK on the private relation
$C(x) \quad = \quad y$

- Black-box
- Verifiability?

Prover wants to prove $C(\llbracket x \rrbracket) = \llbracket y \rrbracket$, where $\llbracket \cdot \rrbracket$ supports some homomorphism.

SNARK on top of Homomorphism

$C(\llbracket x \rrbracket) = \llbracket y \rrbracket$ is a public relation

SNARK on the public relation
$C(\llbracket x \rrbracket) \quad = \quad \llbracket y \rrbracket$

Homomorphic Evaluation
on $\llbracket x \rrbracket$ to get $\llbracket y \rrbracket$

- Non-black-box

Homomorphism on top of SNARK

$C(x) = y$ is a private relation. (Even the verifier may not know this statement)

Homomorphic Evaluation on $\llbracket x \rrbracket$
to generate the SNARK proof

SNARK on the private relation
$C(x) \quad = \quad y$

- Black-box
- Verifiability?
- What Homomorphism needed beyond C?

Prover wants to prove $C(\llbracket x \rrbracket) = \llbracket y \rrbracket$, where $\llbracket \cdot \rrbracket$ supports some homomorphism.

SNARK on top of Homomorphism

$C(\llbracket x \rrbracket) = \llbracket y \rrbracket$ is a public relation

SNARK on the public relation
$C(\llbracket x \rrbracket) \quad = \quad \llbracket y \rrbracket$

Homomorphic Evaluation
on $\llbracket x \rrbracket$ to get $\llbracket y \rrbracket$

- Non-black-box

Homomorphism on top of SNARK

$C(x) = y$ is a private relation. (Even the verifier may not know this statement)

Homomorphic Evaluation on $\llbracket x \rrbracket$
to generate the SNARK proof

SNARK on the private relation
$C(x) \quad = \quad y$

- Black-box
- Verifiability?
- What Homomorphism needed beyond C?

What kind of operations does a prover need to perform to prove C?

- Feasibility: $\llbracket x \rrbracket$ may only support linear homomorphism: g^x

Prover wants to prove $C(\llbracket x \rrbracket) = \llbracket y \rrbracket$, where $\llbracket \cdot \rrbracket$ supports some homomorphism.

SNARK on top of Homomorphism

$C(\llbracket x \rrbracket) = \llbracket y \rrbracket$ is a public relation

SNARK on the public relation
$C(\llbracket x \rrbracket) \quad = \quad \llbracket y \rrbracket$

Homomorphic Evaluation
on $\llbracket x \rrbracket$ to get $\llbracket y \rrbracket$

- Non-black-box

Homomorphism on top of SNARK

$C(x) = y$ is a private relation. (Even the verifier may not know this statement)

Homomorphic Evaluation on $\llbracket x \rrbracket$
to generate the SNARK proof

SNARK on the private relation
$C(x) \quad = \quad y$

- Black-box
- Verifiability?
- What Homomorphism needed beyond C?

What kind of operations does a prover need to perform to prove C?

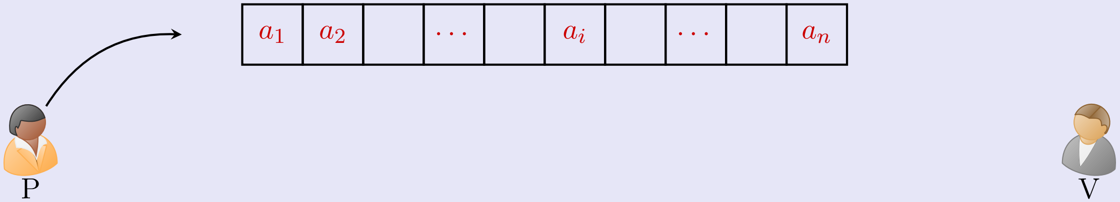
- Feasibility: $\llbracket x \rrbracket$ may only support linear homomorphism: g^x
- Efficiency: The efficiency of FHE depends on the homomorphism supported.

Interactive Oracle Proof (IOP) [Ben-Sasson-Chiesa-Spooner'16]

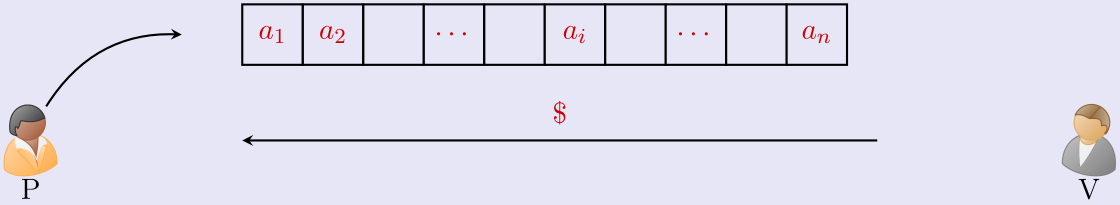
P

V

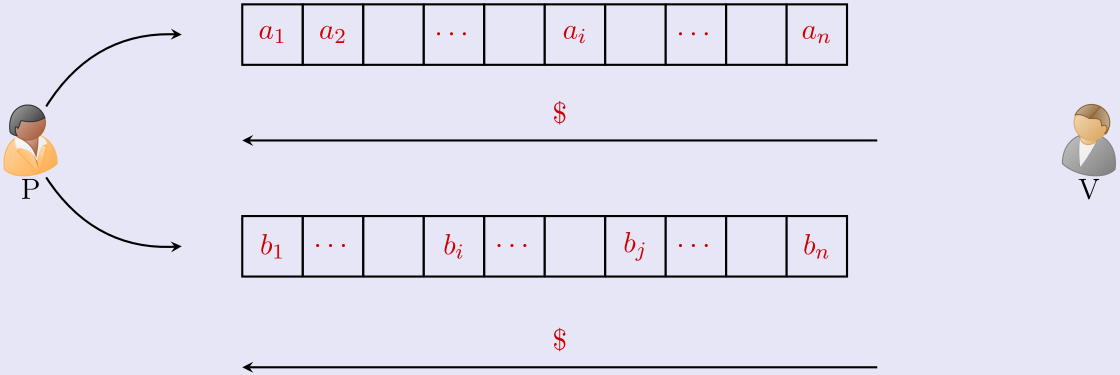
Interactive Oracle Proof (IOP) [Ben-Sasson-Chiesa-Spooner'16]



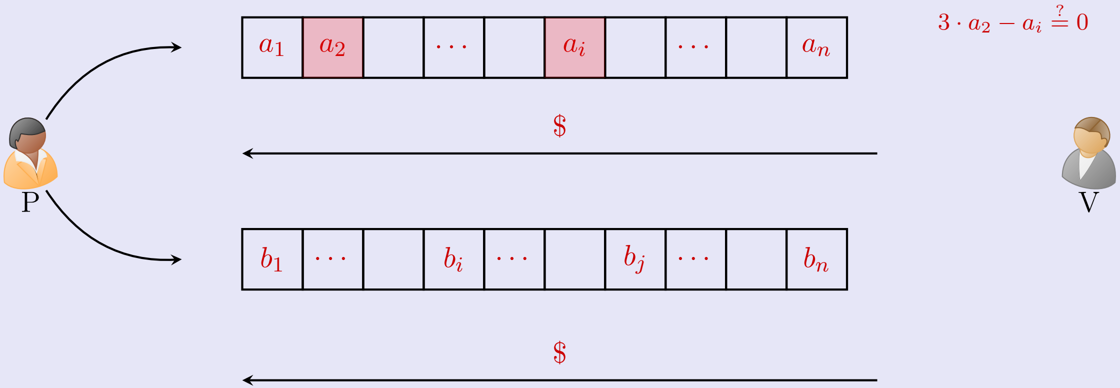
Interactive Oracle Proof (IOP) [Ben-Sasson-Chiesa-Spooner'16]



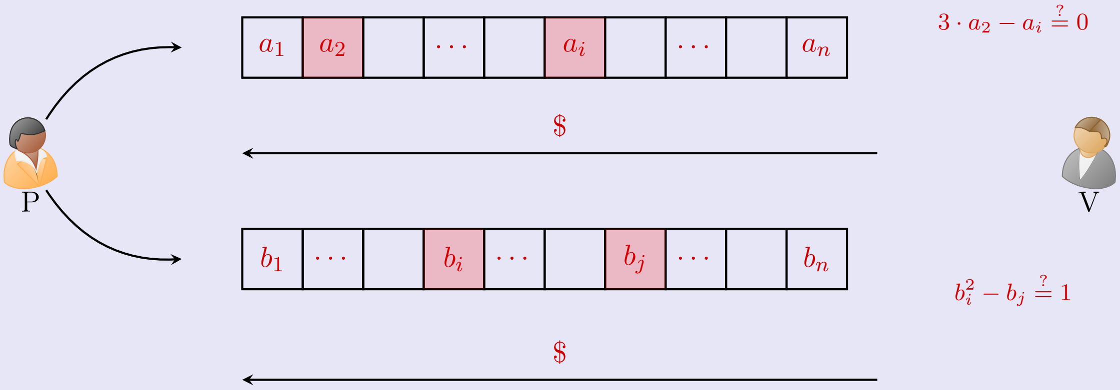
Interactive Oracle Proof (IOP) [Ben-Sasson-Chiesa-Spooner'16]



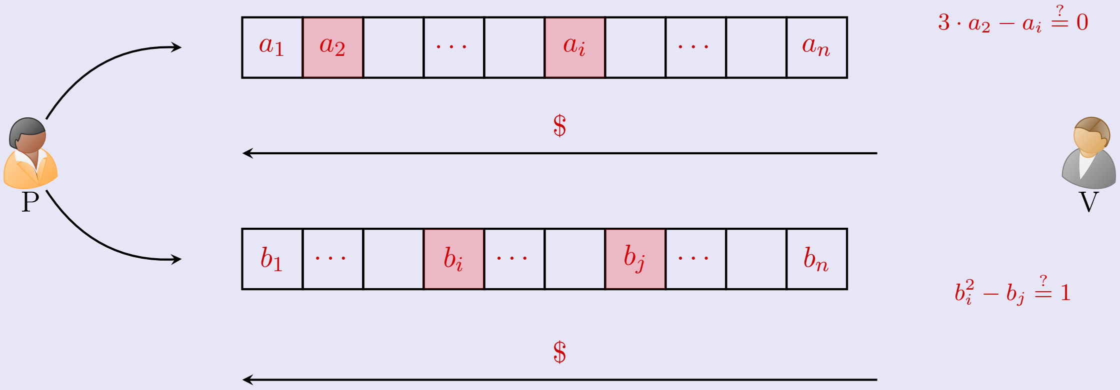
Interactive Oracle Proof (IOP) [Ben-Sasson-Chiesa-Spooner'16]



Interactive Oracle Proof (IOP) [Ben-Sasson-Chiesa-Spooner'16]

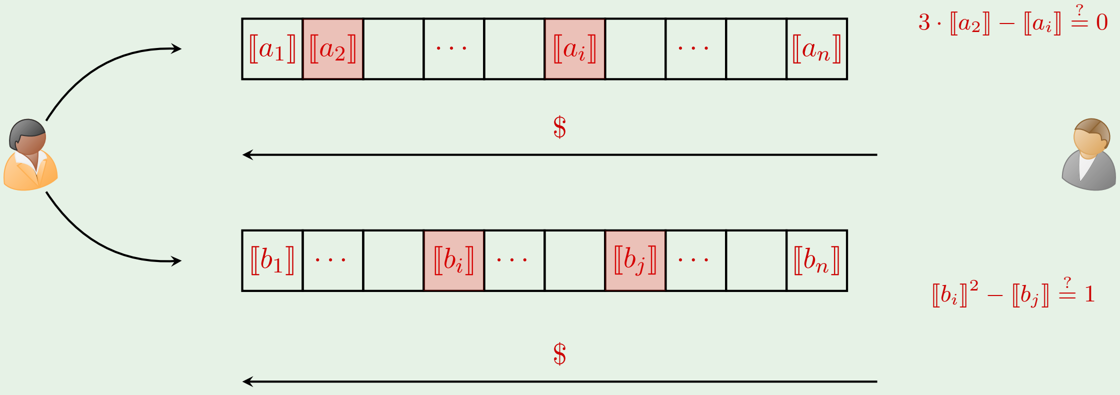


Interactive Oracle Proof (IOP) [Ben-Sasson-Chiesa-Spooner'16]

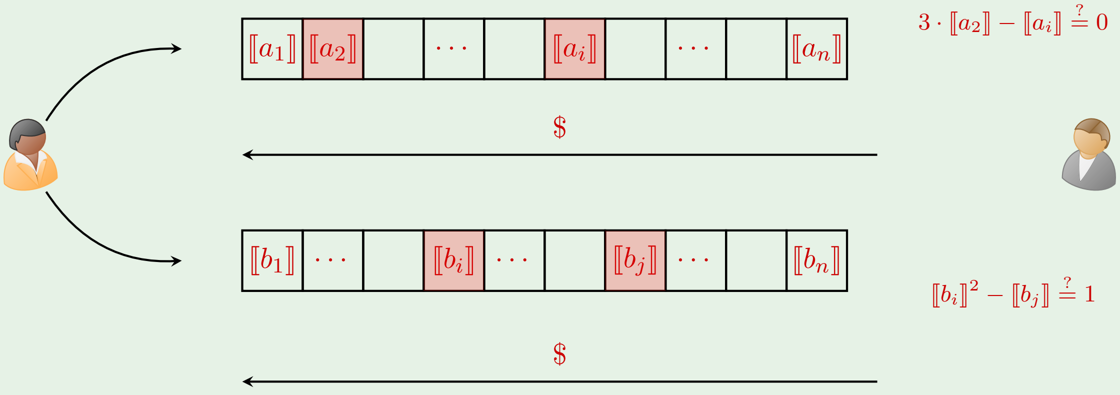


- Can be compiled into SNARKs using Merkle's commitment and Fiat-Shamir.
  - Proof size grows with # queries

IOP on Hidden Values

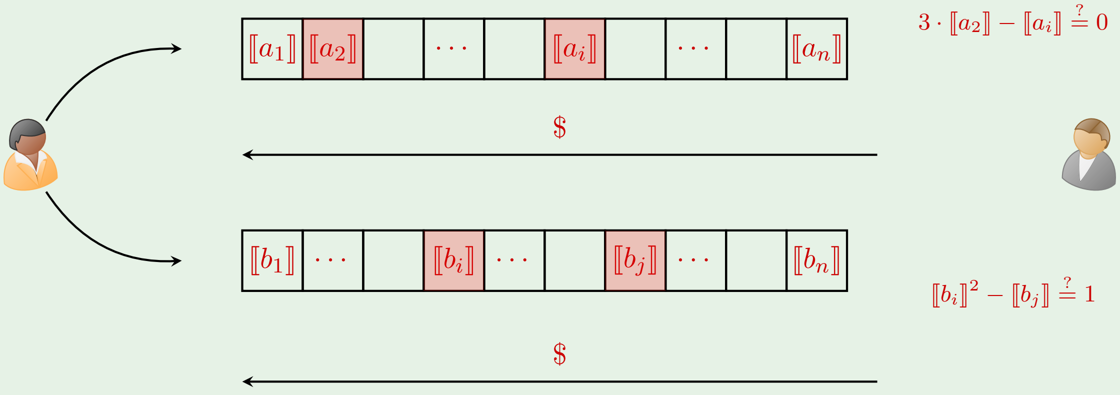


IOP on Hidden Values



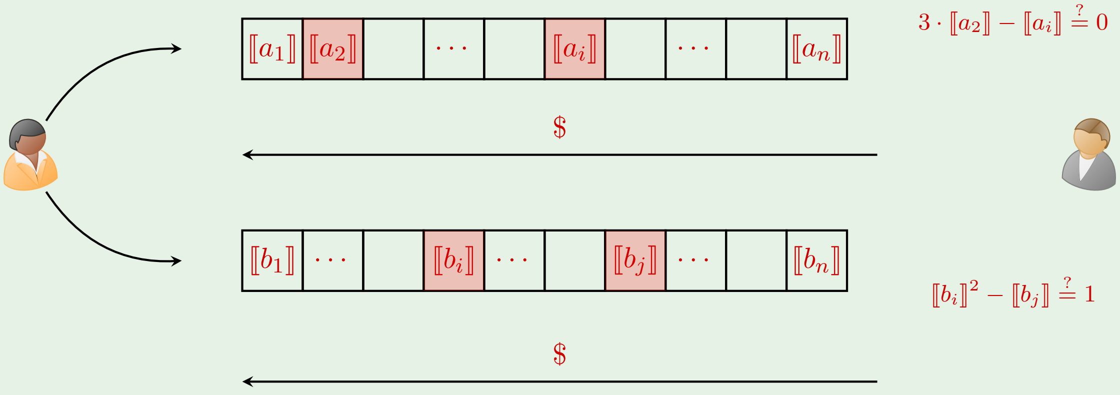
- Privately verifiable if decryptable

IOP on Hidden Values



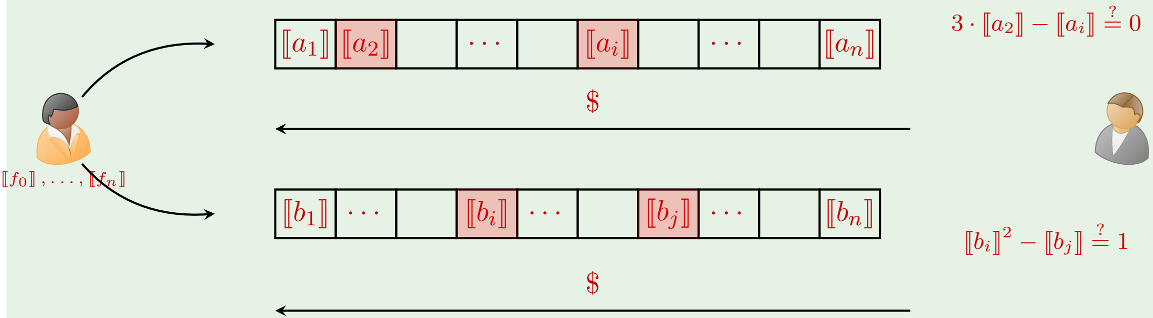
- Privately verifiable if decryptable
- How to compile IOP to SNARKs?

IOP on Hidden Values

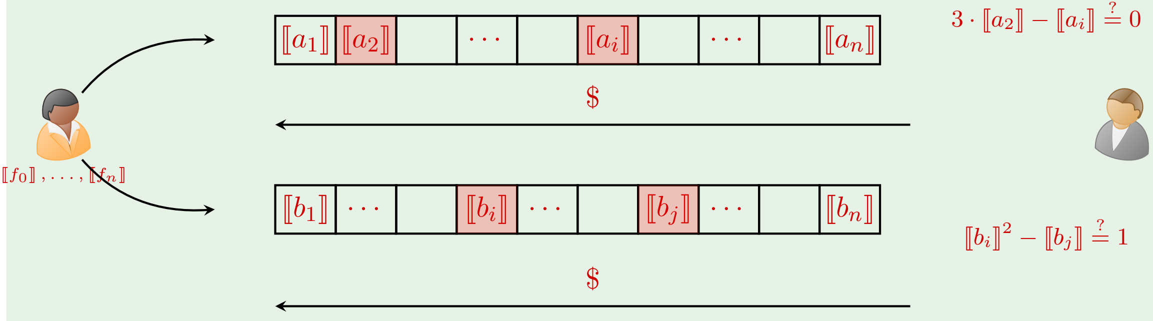


- Privately verifiable if decryptable
- How to compile IOP to SNARKs?
  - directly apply Merkle's commitment and Fiat-Shamir on $[\![x]\!]$
  - No need for leveraging homomorphism on random oracle computation

FRI on Hidden Values

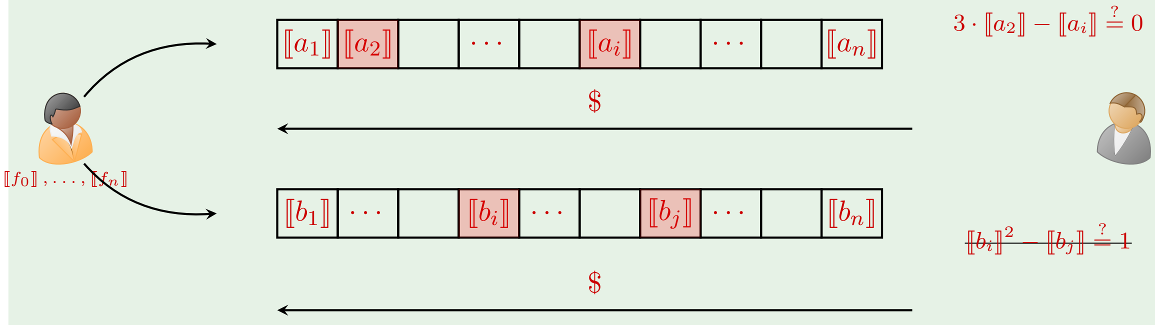


FRI on Hidden Values



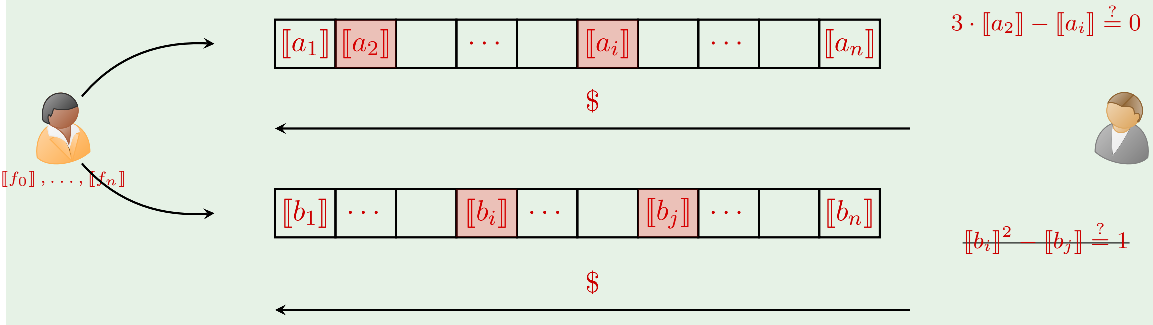
- Polynomial Commitment for hidden polynomial. $f(x) := f_0 + f_1 \cdot x + \cdots f_n \cdot x^n$

FRI on Hidden Values



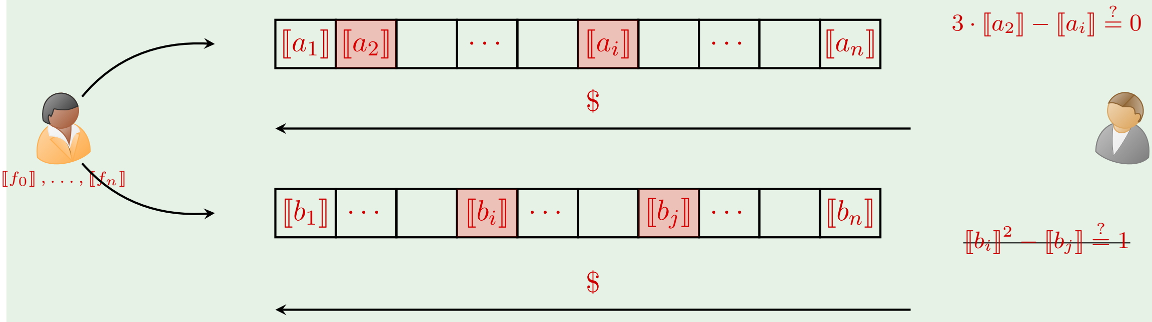
- Polynomial Commitment for hidden polynomial. $f(x) := f_0 + f_1 \cdot x + \cdots f_n \cdot x^n$
- Only linear operations (prover & verifier) $\Longrightarrow$ linear homomorphism suffices

FRI on Hidden Values



- Polynomial Commitment for hidden polynomial. $f(x) := f_0 + f_1 \cdot x + \cdots f_n \cdot x^n$
- Only linear operations (prover & verifier) $\Longrightarrow$ linear homomorphism suffices
- Public Verifiability?

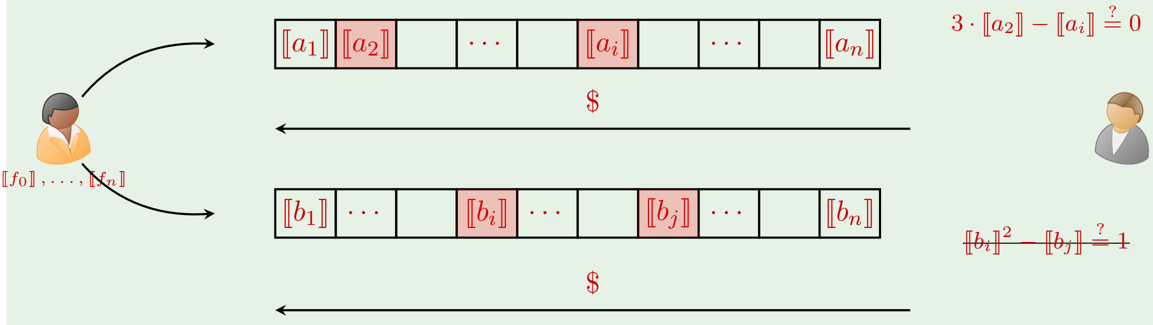
FRI on Hidden Values



- Polynomial Commitment for hidden polynomial. $f(x) := f_0 + f_1 \cdot x + \cdots f_n \cdot x^n$
- Only linear operations (prover & verifier) $\Longrightarrow$ linear homomorphism suffices
- Public Verifiability?
  - Homomorphic in randomness $\Longrightarrow$ Check relation at encapsulation level

$$(g^{a_2})^3 \; / \; g^{a_i} \stackrel{?}{=} 1$$

FRI on Hidden Values



- Polynomial Commitment for hidden polynomial. $f(x) := f_0 + f_1 \cdot x + \cdots f_n \cdot x^n$
- Only linear operations (prover & verifier) $\Longrightarrow$ linear homomorphism suffices
- Public Verifiability?
  - Homomorphic in randomness $\Longrightarrow$ Check relation at encapsulation level

$$(g^{a_2})^3 \;/\; g^{a_i} \stackrel{?}{=} 1$$

FRI + Polynomial IOP: $|C| \cdot \log |C|$ operations with multiplication depth $\mathsf{depth}(C) + O(1)$

Summary

FRI on hidden values enables oblivious proof generation:

- Verifiable Delegation of Computation
- Delegation of the generation of zkSNARKs to untrusted server
- (Weighted) Threshold Signature without DKG

Future direction: more applications?

Thanks!

Questions?

ia.cr/2023/1609