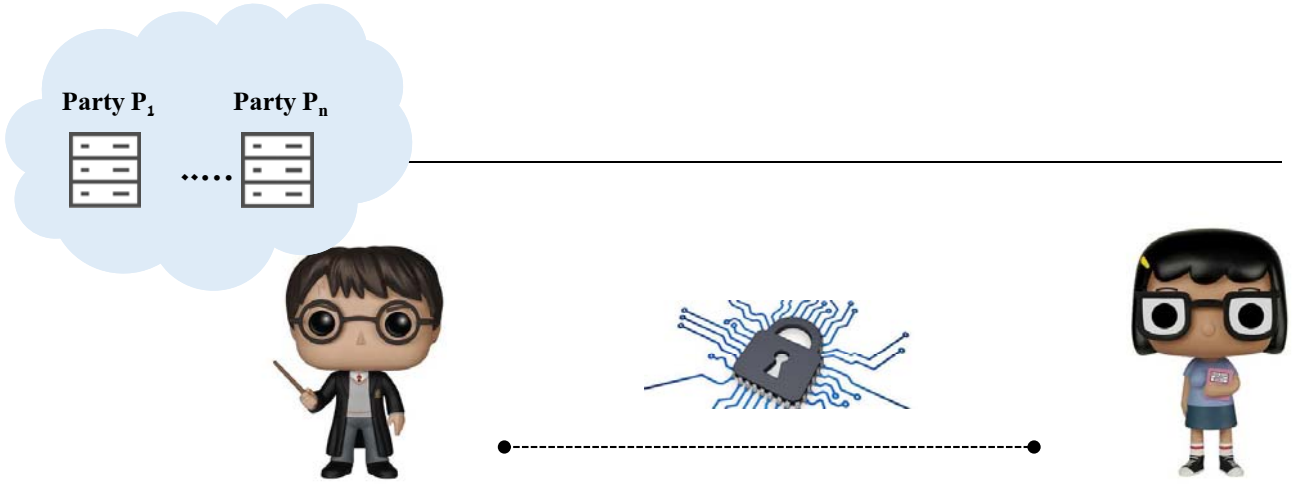
Zero-Knowledge from MPC-in-the-Head: Constructions and Applications



Carmit Hazay
Faculty of Engineering,
Bar-Ilan University

Taxonomy of Proofs

1. P vs NP
2. Interactive vs Non-interactive
3. Trusted setup vs No setup (transparent)
4. ZK vs (only) Soundness
5. Succinct vs Non-succinct
6. Public-Key Crypto vs (only) Symmetric-Key Crypto

Taxonomy of Proofs

1. P vs NP
2. Interactive vs Non-interactive
3. Trusted setup vs No setup (transparent)
4. ZK vs (only) Soundness
5. Succinct vs Non-succinct
6. Public-Key Crypto vs (only) Symmetric-Key Crypto

Taxonomy of Proofs

1. P vs NP
2. Interactive vs Non-interactive
3. Trusted setup vs No setup (transparent)
4. ZK vs (only) Soundness
5. Succinct vs Non-succinct
6. Public-Key Crypto vs (only) Symmetric-Key Crypto

Taxonomy of Proofs

1. P vs NP
2. Interactive vs Non-interactive
3. Trusted setup vs No setup (transparent)
4. ZK vs (only) Soundness
5. Succinct vs Non-succinct
6. Public-Key Crypto vs (only) Symmetric-Key Crypto

Taxonomy of Proofs

1. P vs NP
2. Interactive vs Non-interactive
3. Trusted setup vs No setup (transparent)
4. ZK vs (only) Soundness
5. Succinct vs Non-succinct
6. Public-Key Crypto vs (only) Symmetric-Key Crypto

Taxonomy of Proofs

1. P vs NP
2. Interactive vs Non-interactive
3. Trusted setup vs No setup (transparent)
4. ZK vs (only) Soundness
5. Succinct vs Non-succinct
6. Public-Key Crypto vs (only) Symmetric-Key Crypto

Taxonomy of Proofs

1. P vs NP
2. Interactive vs Non-interactive
3. Trusted setup vs No setup (transparent)
4. ZK vs (only) Soundness
5. Succinct vs Non-succinct
6. Public-Key Crypto vs (only) Symmetric-Key Crypto

Prior Approaches to "Practical" ZK

1. **Probabilistically Checkable Proofs (PCPs)** [BFLS91, Kil92, Mic94, ALMSS98, AS98, DL08, GLR11, CMT12, BC12, DFH12, BCCT12, IMS12, Tha13, VSBW13], Interactive PCPs [KR08], Interactive Oracle PCPs [BCGT13, BCS16, RRR16, BCGRS16, BBCGGHPRSTV17,BBHR17]

   No setup
   High prover's complexity

2. **Linear PCPs** [IKO07, Gro10, GGPR13, BCIOP13, Gro10, Lip12, SMBW12, Lip13, PGHR13, BCGTV13, FLZ13, SBBPW13, Lip14, DFGK14, KPPSST14, ZPK14, CFHKKNPZ15, WSRBW15, BCTV14, BBFR15, Groth16, FFGKOP16, BFS16, BISW17, GM17,BBBPWM18]

   Short Proofs
   Fast Verification
   Heavy Public-Key Crypto
   Trusted Setup
   Quantum Insecure

3. **Interactive Proofs (IP)** [GKR08, ZGKPP17-18, WTSTW18]

   No setup
   Moderate Public-Key Crypto

4. **Multiparty Computation (MPC)** [IKOS07, GMO16, CDGORRSZ17, AHIV17,KKW18]

Zero-Knowledge from MPC [IKOS07]

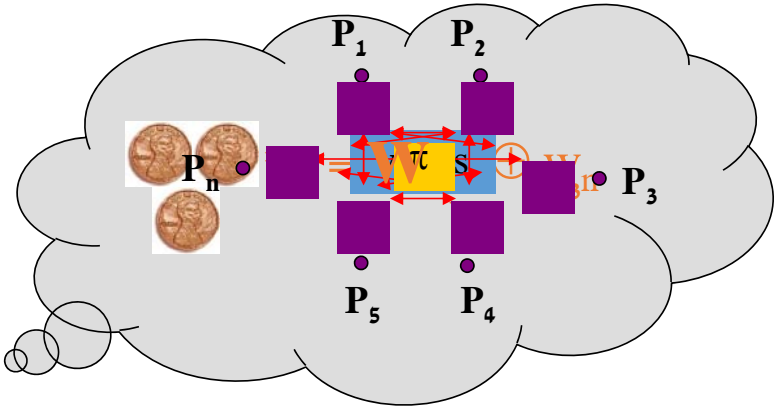
- Goal: ZK proof for an NP-relation $R(x,w)$

- Towards using MPC:
  - Define n-party functionality

    $g(x;\ w_1,\ldots,w_n) = R(x,\ w_1 \oplus \ldots \oplus w_n)$

- Use OT-based MPC
  - Security in semi-honest model

Zero-Knowledge from MPC [IKOS07]



Given MPC protocol π for
$g(x; w_1,\ldots,w_n) = R(x, w_1 \oplus \ldots \oplus w_n)$

Prover

Verifier

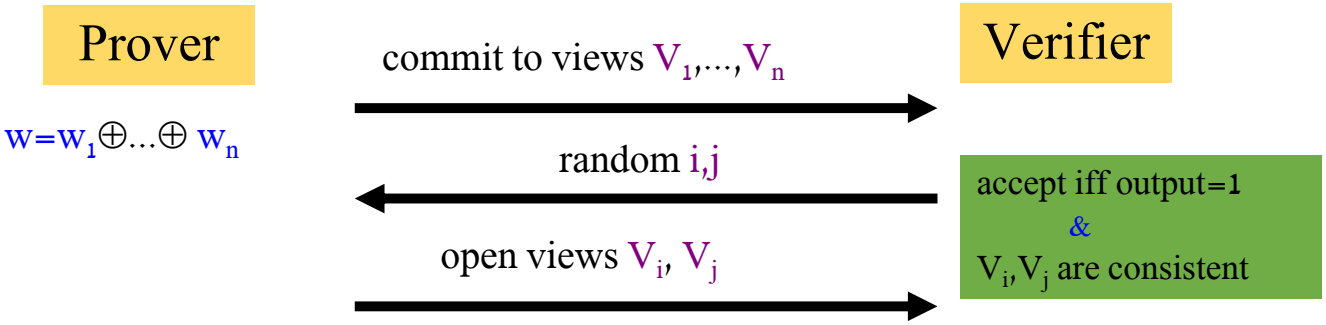
commit to views $V_1,\ldots,V_n$

random i,j

open views V_i, V_j

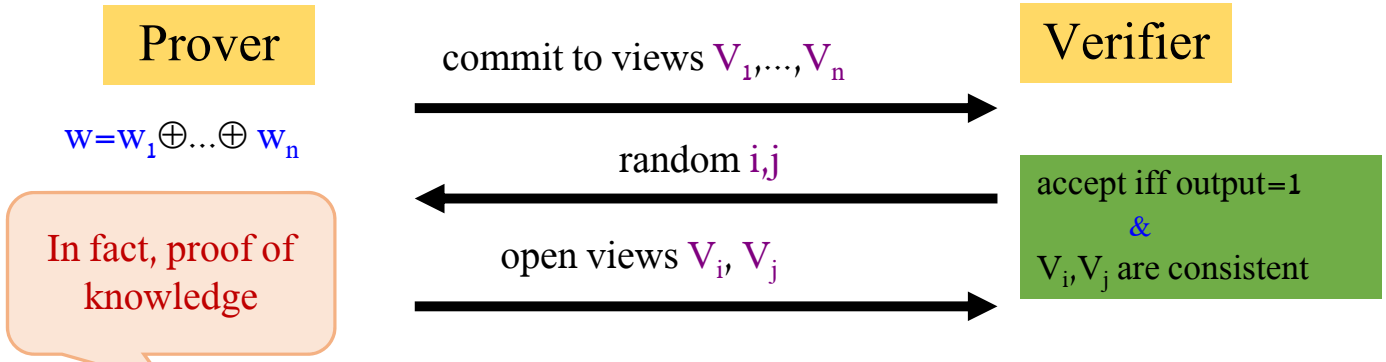
accept iff output=1
&
Vi,Vj are consistent

Analysis



- Completeness: √
- Zero-knowledge: by 2-security of π and randomness of w_i, w_j

Analysis

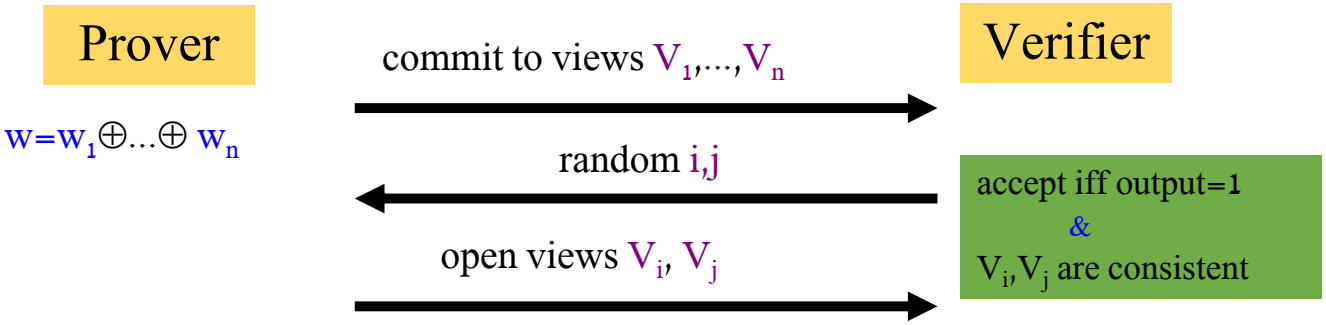


- Soundness: Suppose R(x,w)=0 for all w
  either (1) $V_1,\ldots,V_n$ consistent with protocol π
  or (2) $V_1,\ldots,V_n$ not consistent with π

  (1) outputs=0 (perfect correctness)
  verifier rejects

  (2) for some (i,j), V_i,V_j are inconsistent
  verifier rejects with prob. $\geq \binom{n}{2}$

Analysis



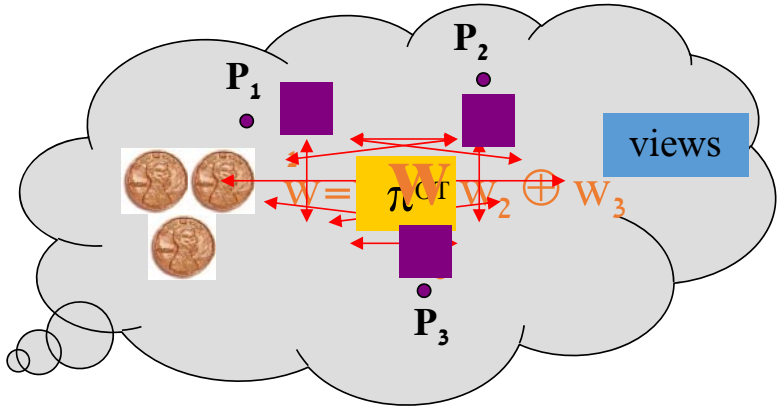
Communication complexity:

$\approx$ (comm. complexity + rand. complexity + input size) of π

ZKBoo: Faster Zero-Knowledge for Boolean Circuits [GMO16]

Post-Quantum Zero-Knowledge and Signatures from Symmetric-Key Primitives (ZKB++) [CDGORRSZ17]

Zero-Knowledge from 3-Party GMW [IKOS07,GMO16]



Use 3-party GMW protocol π^{OT} for
$g(x;\ w_1,w_2,w_3) = R(x,\ w_1 \oplus w_2 \oplus w_3)$

Prover

Verifier

commit to views V_1,V_2,V_3

random i,j

open views V_i, V_j

accept iff output=1
&
V_i,V_j are consistent
soundness error $\leq 2/3$

Extensions

- Variant 1: Use 1-secure MPC
  - Commit to views of parties + channels
  - Open one view and incident channels
- Variant 2: Directly get 2^{-k} soundness error via security in malicious model
  - n=O(k) parties
  - Ω(n)-security with abort
  - Broadcast is “free”

- Handle MPC with error via coin-flipping

Prior Approaches to "Practical" ZK

1. **Probabilistically Checkable Proofs (PCPs)** [BFLS91, Kil92, Mic94, ALMSS98, AS98, DL08, GLR11, CMT12, BC12, DFH12, BCCT12, IMS12, Tha13, VSBW13], Interactive PCPs [KR08], Interactive Oracle PCPs [BCGT13, BCS16, RRR16, BCGRS16, BBCGGHPRSTV17,BBHR17]

   No setup
   High prover's complexity

2. **Linear PCPs** [IKO07, Gro10, GGPR13, BCIOP13, Gro10, Lip12, SMBW12, Lip13, PGHR13, BCGTV13, FLZ13, SBBPW13, Lip14, DFGK14, KPPSST14, ZPK14, CFHKKNPZ15, WSRBW15, BCTV14, BBFR15, Groth16, FFGKOP16, BFS16, BISW17, GM17,BBBPWM18]

   Short Proofs
   Fast Verification
   Heavy Public-Key Crypto
   Trusted Setup
   Quantum Insecure

3. **Interactive Proofs (IP)** [GKR08, ZGKPP17-18, WTSTW18]

   No setup
   Moderate Public-Key Crypto

4. **Multiparty Computation (MPC)** [IKOS07, GMO16, CDGORRSZ17, AHIV17,KKW18]

   No Setup
   Fast Prover
   Post Quantum Secure
   Everything Linear

Ligero: Lightweight Sublinear Arguments Without a Trusted Setup [AHIV17]

High-Level Overview

High level approach: use **MPC in the head** [IKOS07]

- Transform Honest-majority MPC to ZK
- Optimized and implemented in [GMO16,CDGORRSZ17]

Can the communication be sublinear?
Communication complexity of (i.t.) MPC > circuit size

Key insight: Communication per party can be sublinear [DI06,IPS09]

High-Level Overview

High level approach: use **MPC in the head** [IKOS07]

- Transform Honest-majority MPC to ZK
- Optimized and implemented in [GMO16,CDGORRSZ17]

MPC $\longrightarrow$ Interactive PCP[KR08] $\xrightarrow{\text{[BCS16]}}$ ZK

it size

Key insight: Communication per party can be sublinear [DI06,IPS09]

Main Result

Sublinear ZK arguments without trusted setup

- Simple, concretely efficient
- Symmetric-crypto only (eg, SHA256)
- Post-quantum secure

First "sublinear" arguments for NP that avoid both complex PCP machinery and public-key crypto

Main Result

Sublinear ZK arguments without trusted setup

Concretely:

- **40-bit security:** comm. is $0.5\sqrt{|C|}$ kb in the Boolean case
- Can be made non-interactive via Fiat-Shamir
- Can handle Boolean or arithmetic circuits
- Prover computation: Merkle Tree ($O(\sqrt{|C|})$ leaves) + $O(\sqrt{|C|})$ FFT's of $O(\sqrt{|C|})$ evaluations

Eg, SHA256 certification with 40-bit security:

i.e. For statement y, prover proves knowledge of x such that SHA256(x) = y

	Linear PCP [Pinocchio]	ZKBoo/++ [CDGORRSZ17]	**Ligero**
Communication	~ bytes	200 KB	**34 KB**
Prover time	mins	~33ms	**140ms**
Verifier time	<10ms	~38ms	**60ms**
Asymptotic Communication	~ bytes	O(|C|)	**$O(\sqrt{|C|})$**
Trusted Setup	YES	NO	**NO**
Amortization	NA	NO	**YES**

Proof Schematic

Prover

Commitment

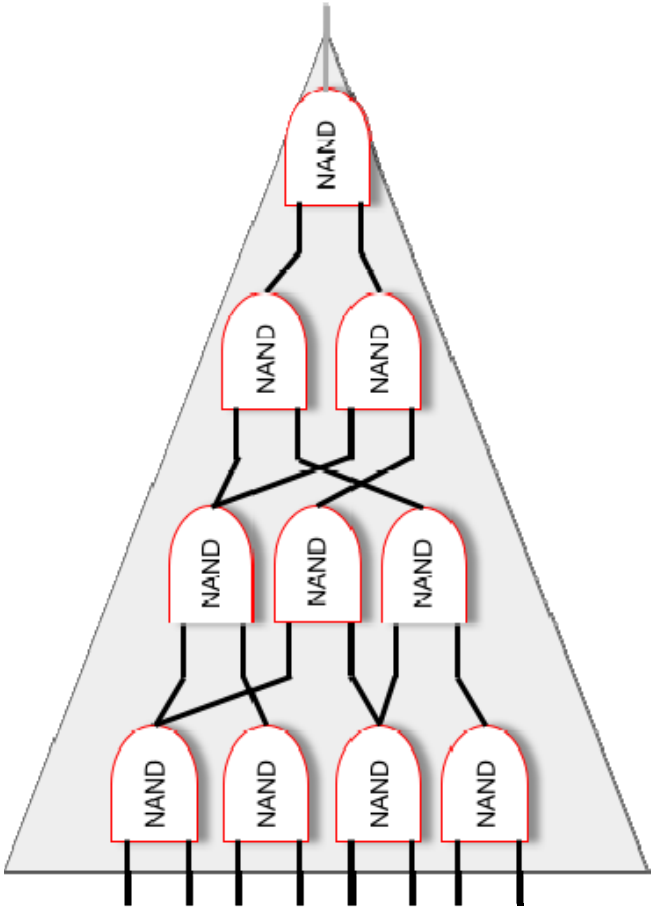
Challenge

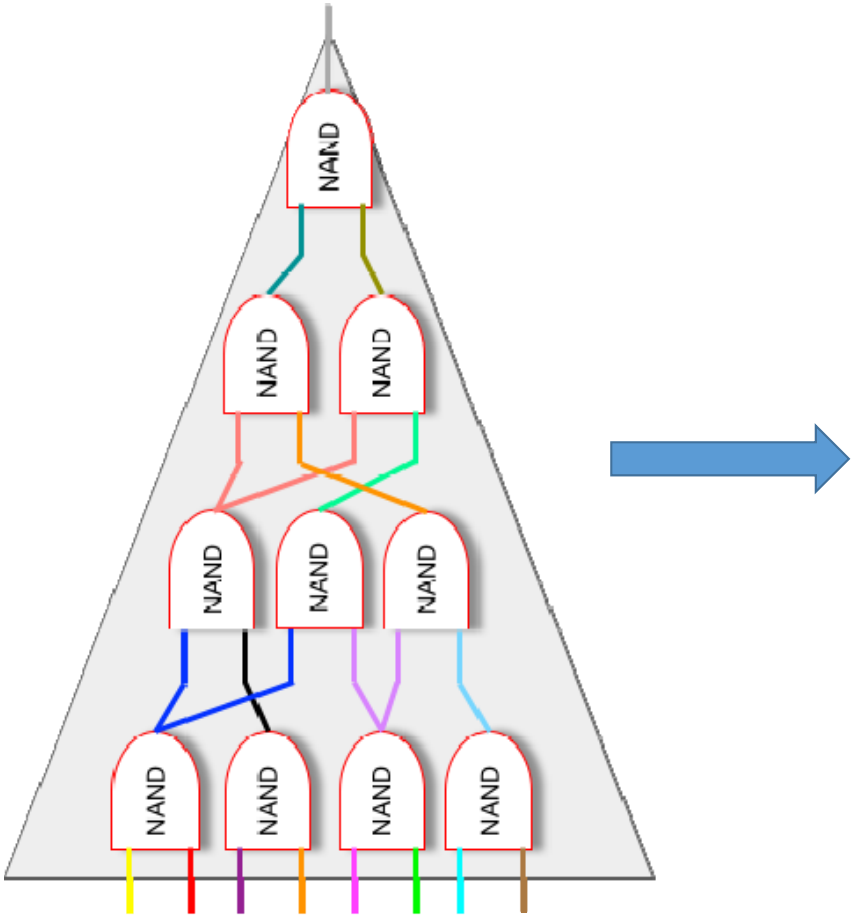
Response

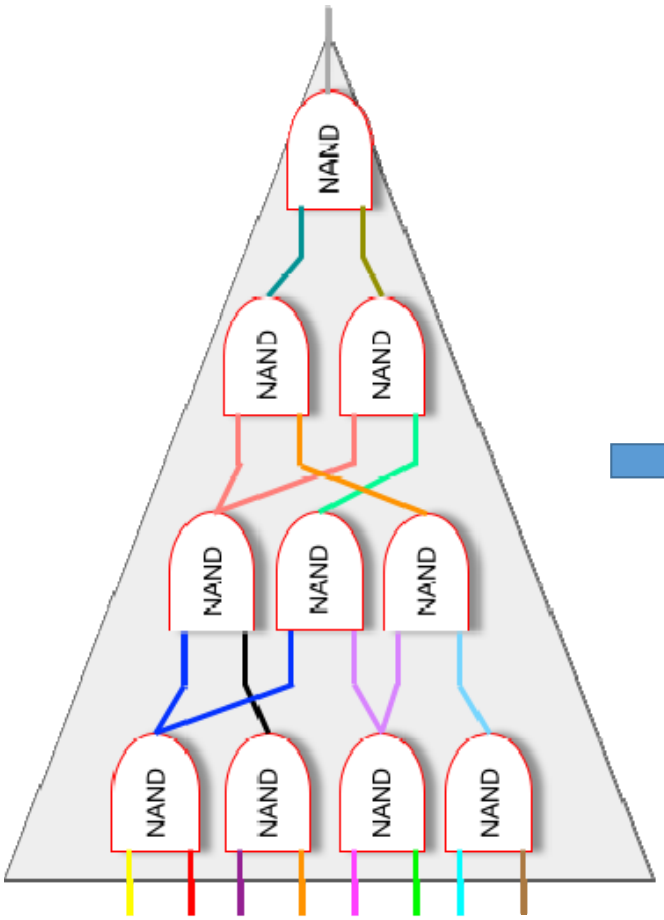
Challenge

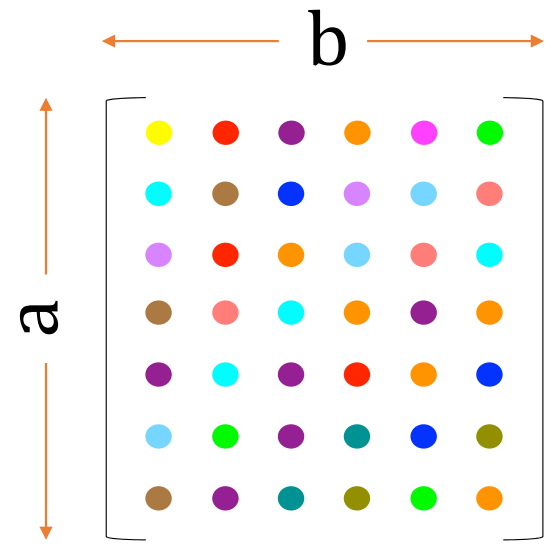
Response

Verifier


NAND
NAND
NAND
NAND
NAND
NAND
NAND
NAND
NAND
NAND

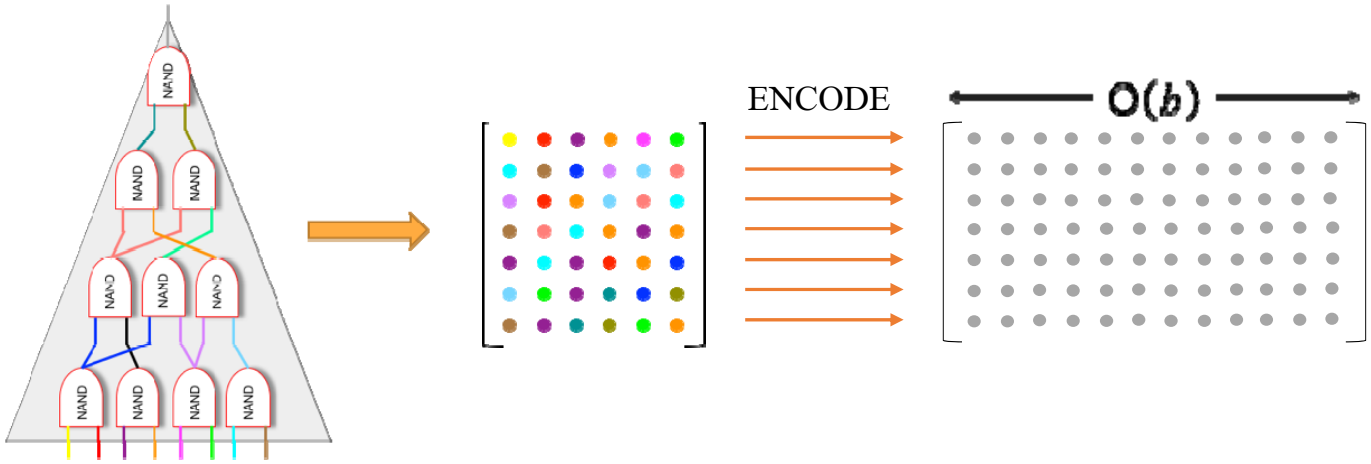

NAND
NAND
NAND
NAND
NAND
NAND
NAND
NAND
NAND





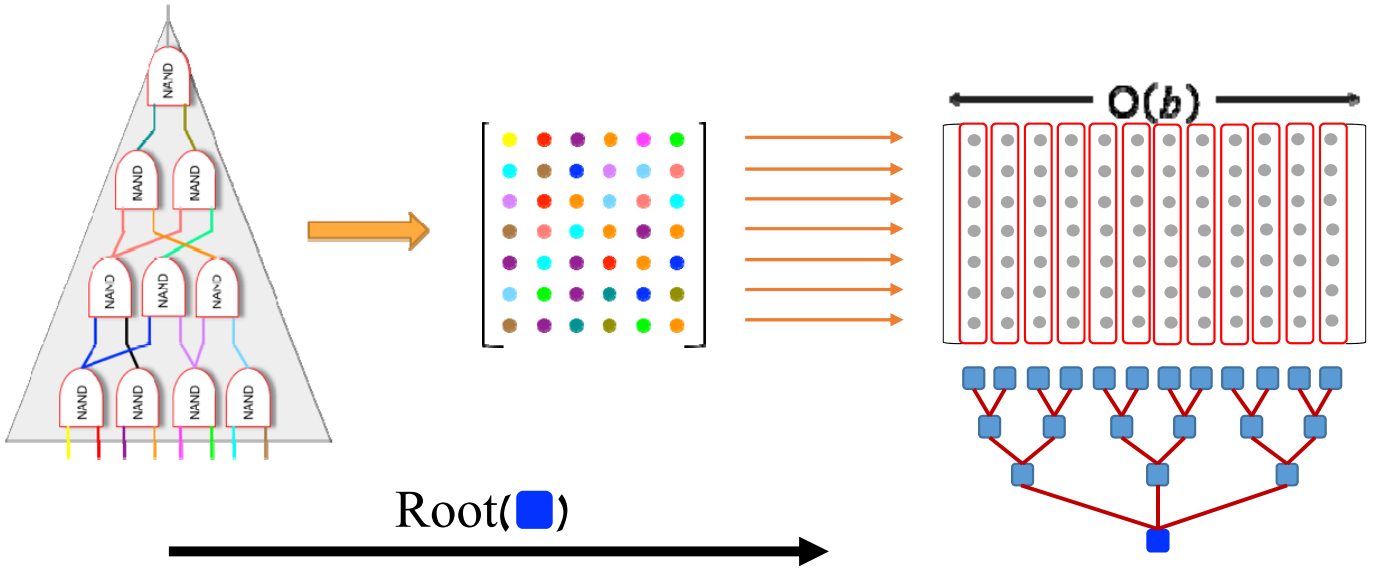
$$a \cdot b \geq X \cdot \#gates$$

Boolean: X = 2, AND/XOR
Arithmetic: X = 3, AND



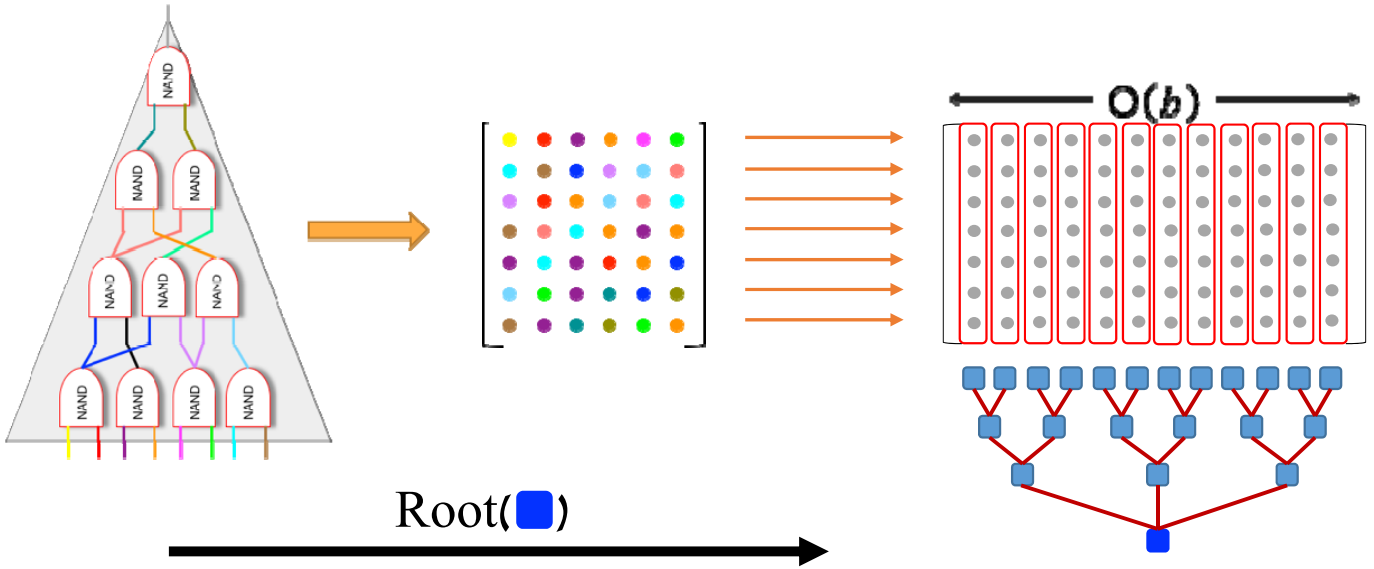
Prover

Verifier



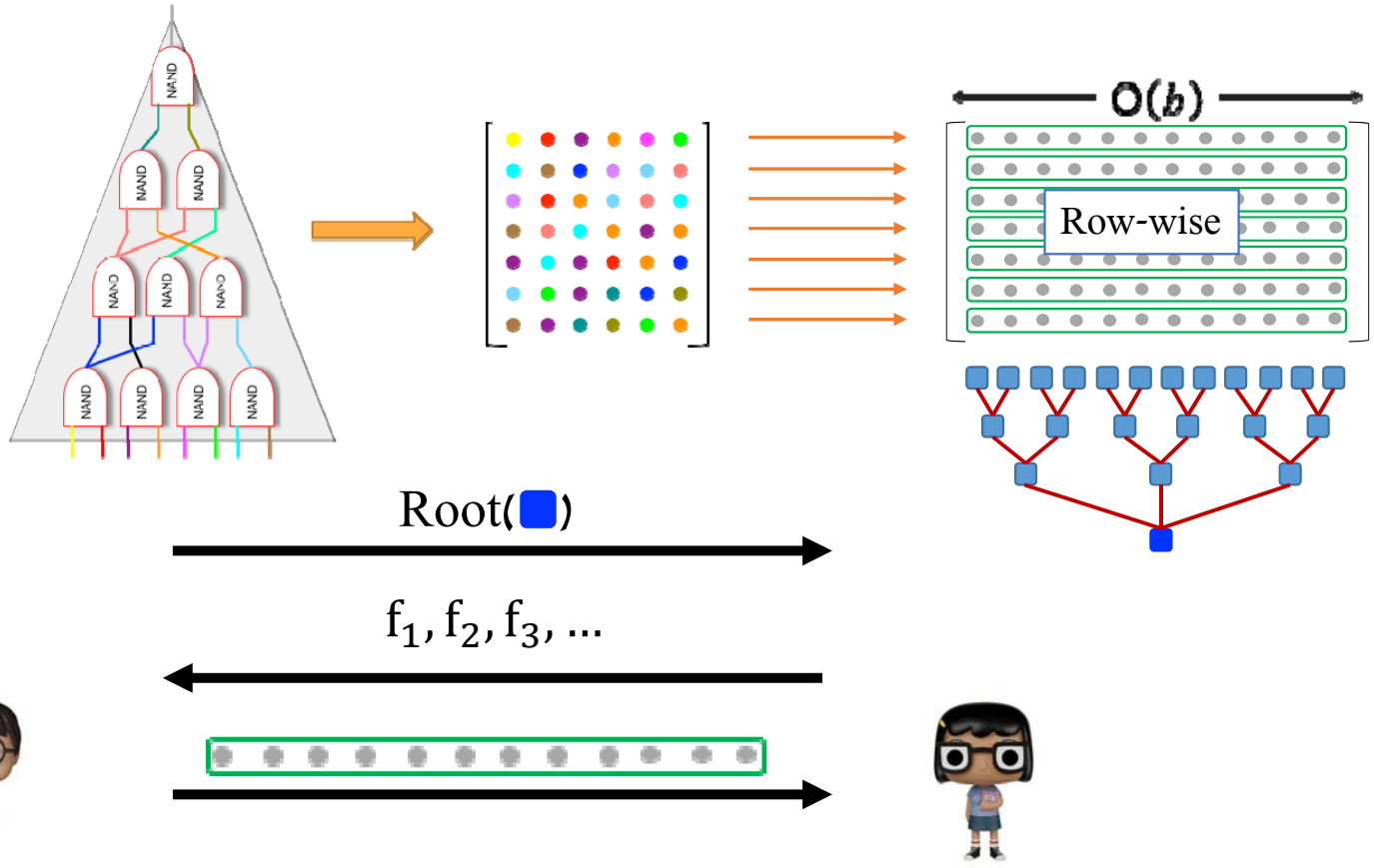
Prover

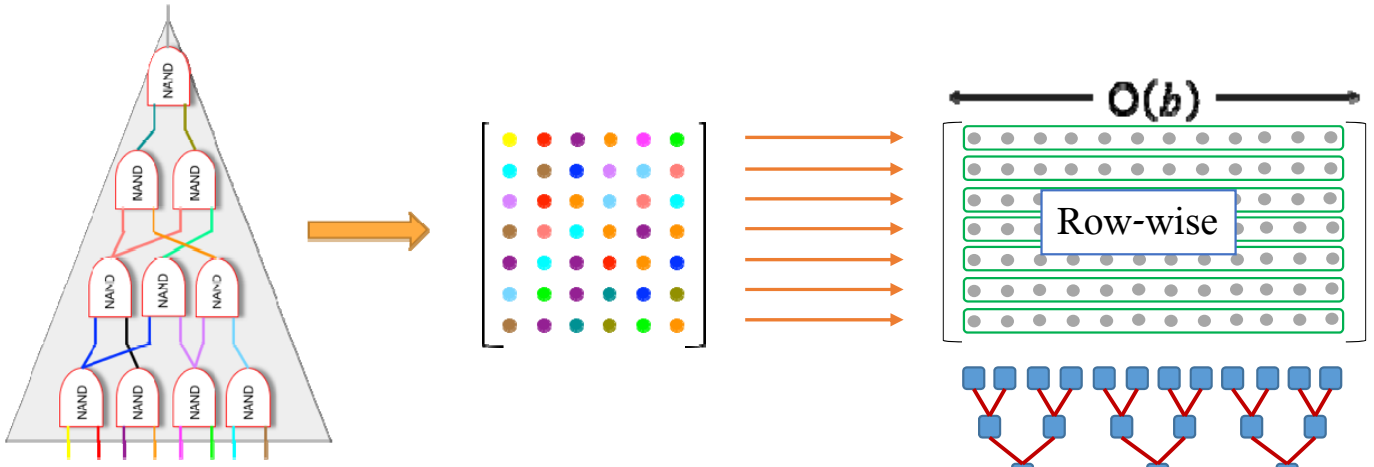
Verifier



Prover

Verifier


NAND
O(b)
Row-wise
Root(■)
$f_1, f_2, f_3, \ldots$
Prover
Verifier



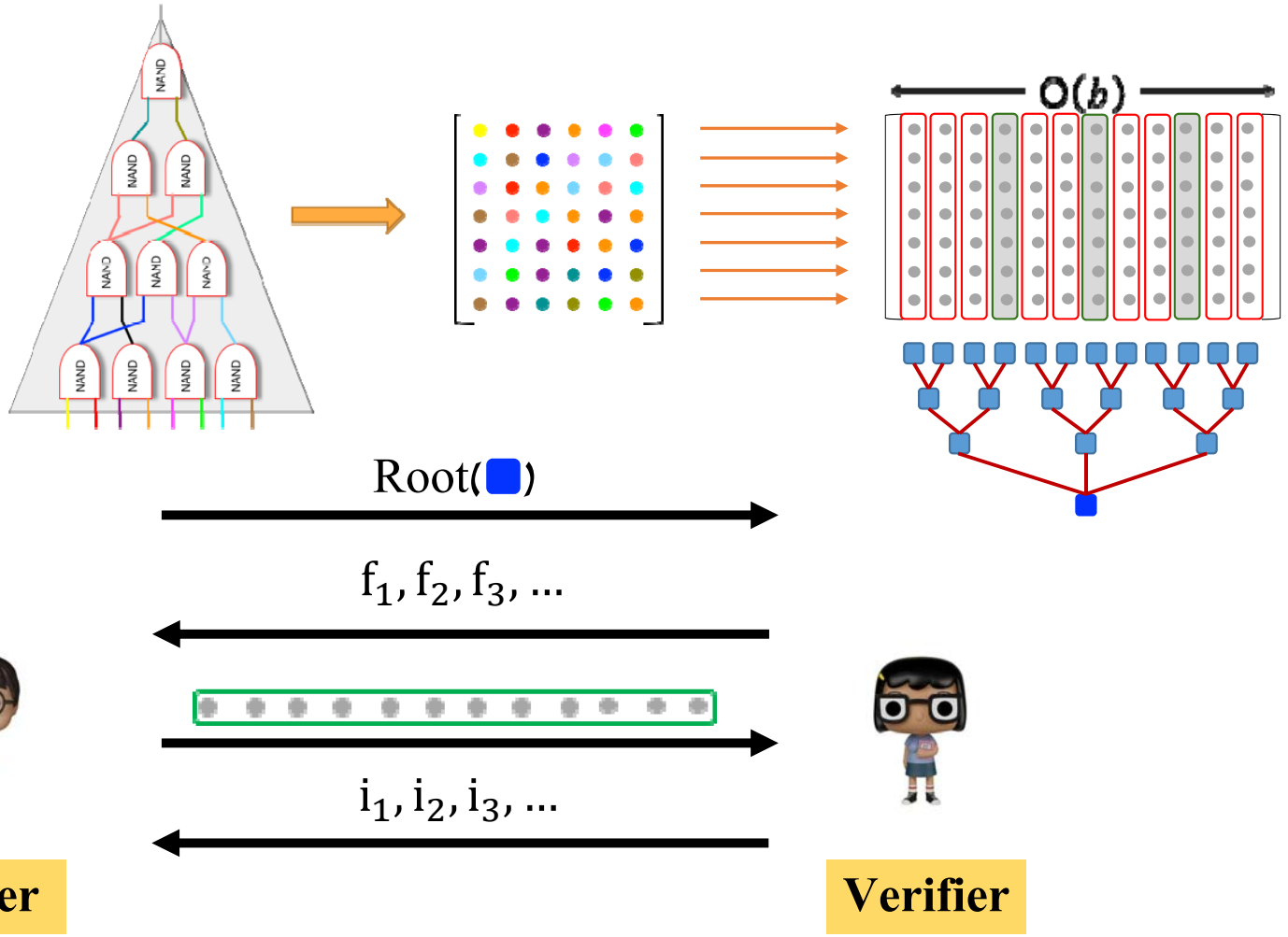
Root(■)

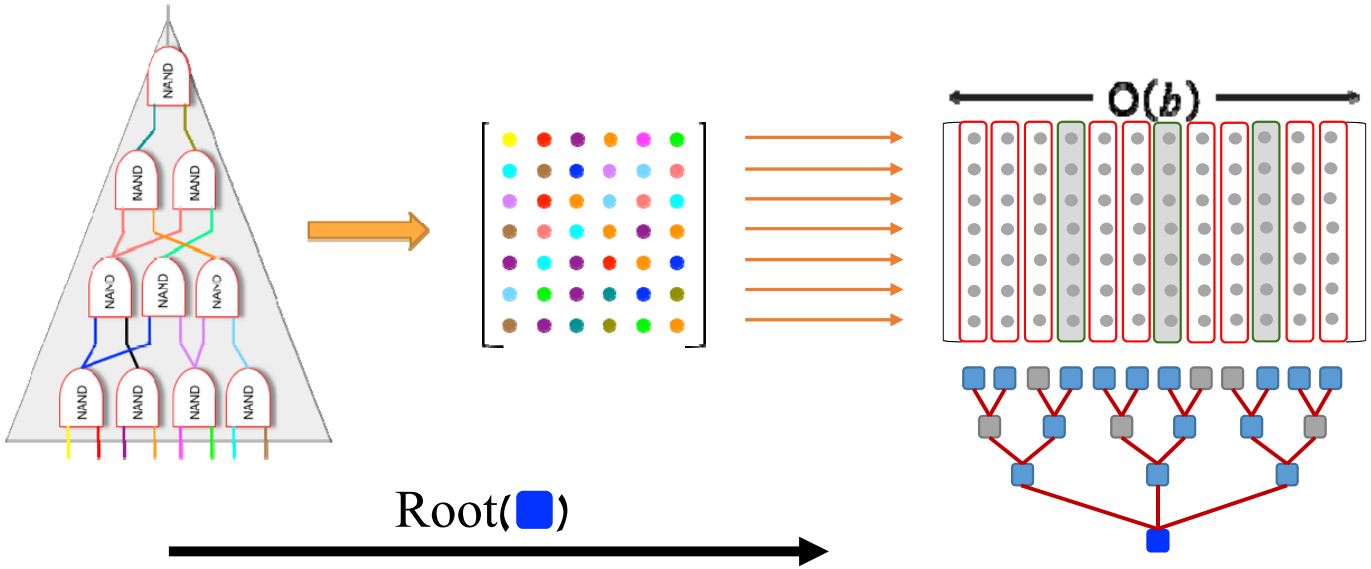
$f_1, f_2, f_3, \ldots$

$i_1, i_2, i_3, \ldots$

Prover

Verifier


NAND
O(b)
Root(■)
$f_1, f_2, f_3, \ldots$
$i_1, i_2, i_3, \ldots$
Prover
Verifier



Root(■)

$f_1, f_2, f_3, \ldots$

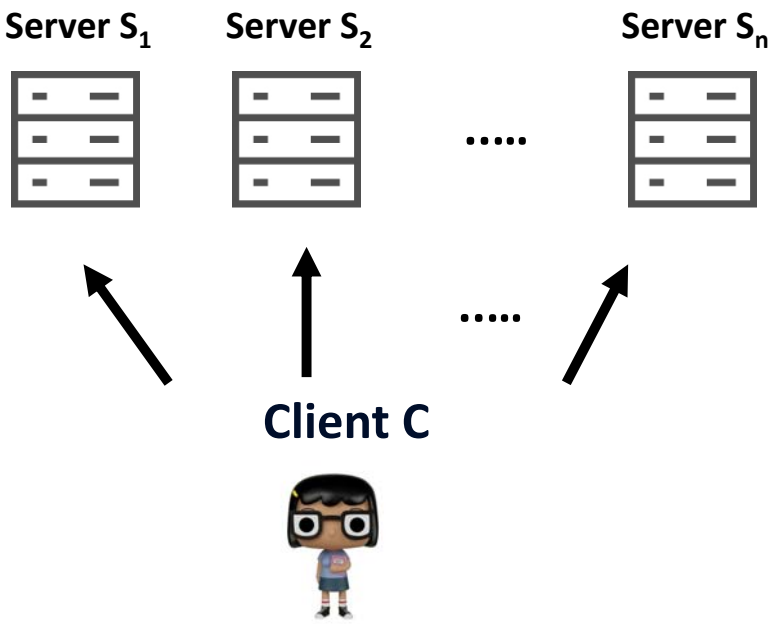
$i_1, i_2, i_3, \ldots$

Prover

Verifier

Proof Length:
$O(\mathbf{b} + \boldsymbol{\kappa} \cdot \mathbf{a})$
Computation:
$O(\mathbf{a})$ FFTs of $O(\mathbf{b})$

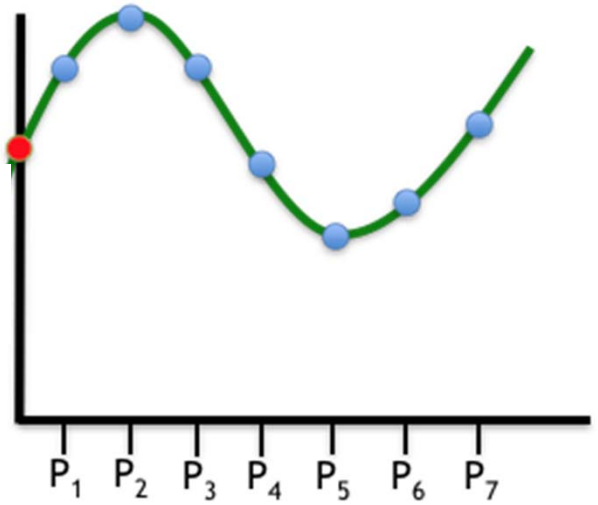
The Underlying MPC Protocol



1. Input sharing phase
   - Sharing of extended witness
   - Server's view is a matrix column
2. Local computation
   - Proofs of correctness

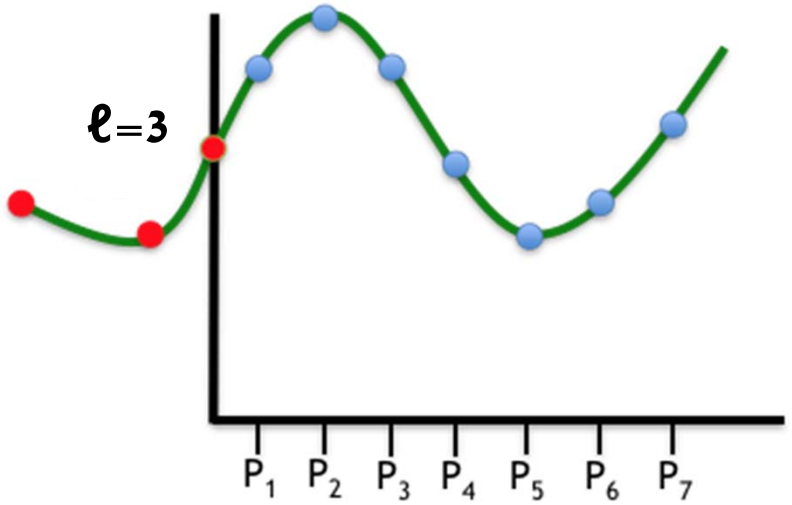
Idea 1: Shamir Secret Sharing [S79]

Pick a random **t**-degree polynomial p such that
p(0) is secret
Distribute p(1), …, p(n)
t shares do not reveal the secrets
n-t/2 modified shares do not affect correctness



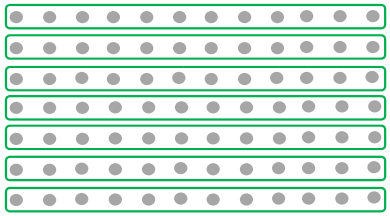
Idea 1: Packed Secret Sharing [FY92]

Pick a random **t+ℓ**-degree polynomial p such that
p(0), p(-1), …, p(-ℓ) are secrets
Distribute p(1), …, p(n)
t+ℓ shares do not reveal the secrets



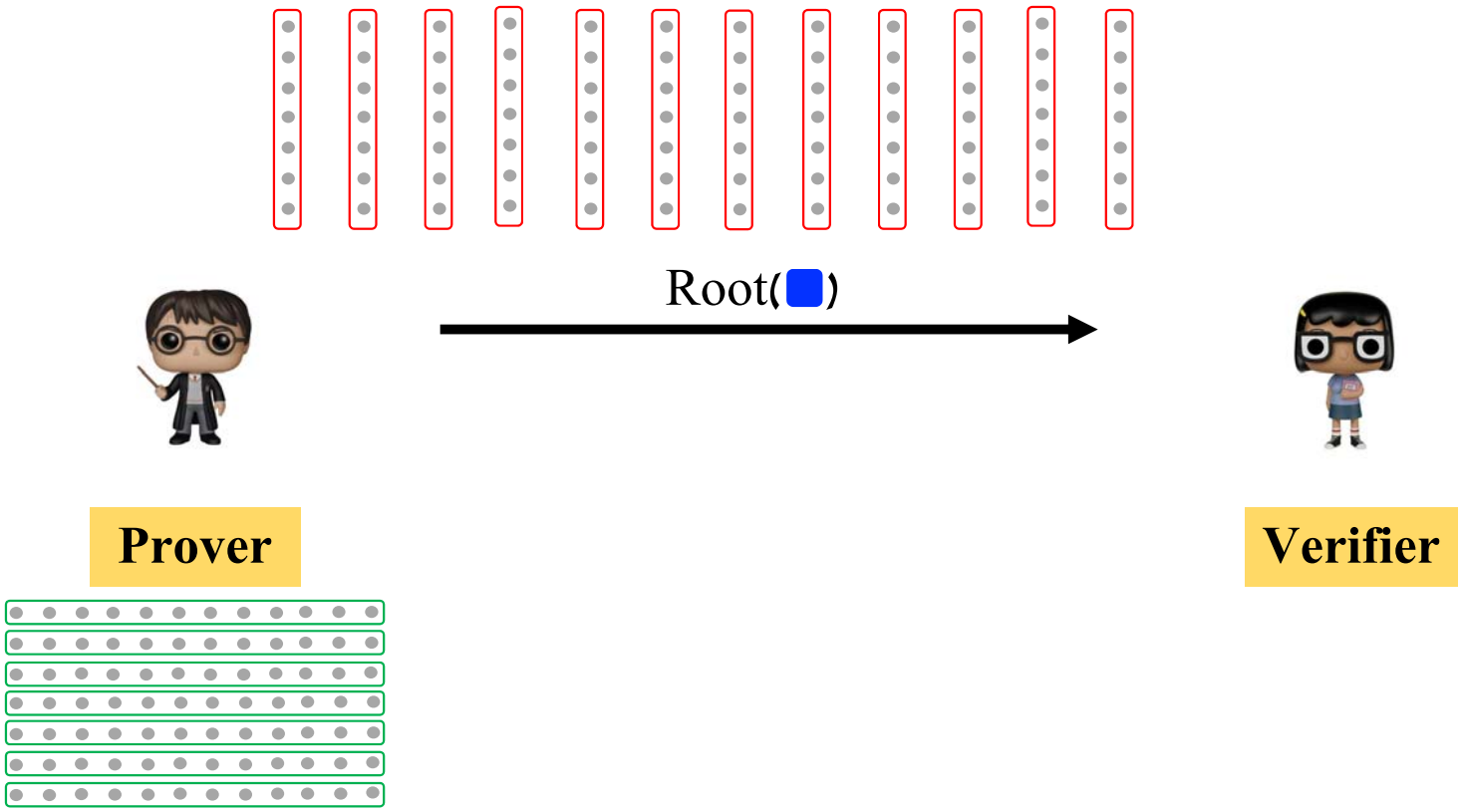
Idea 2: Testing Interleaved RS Codes

Prover

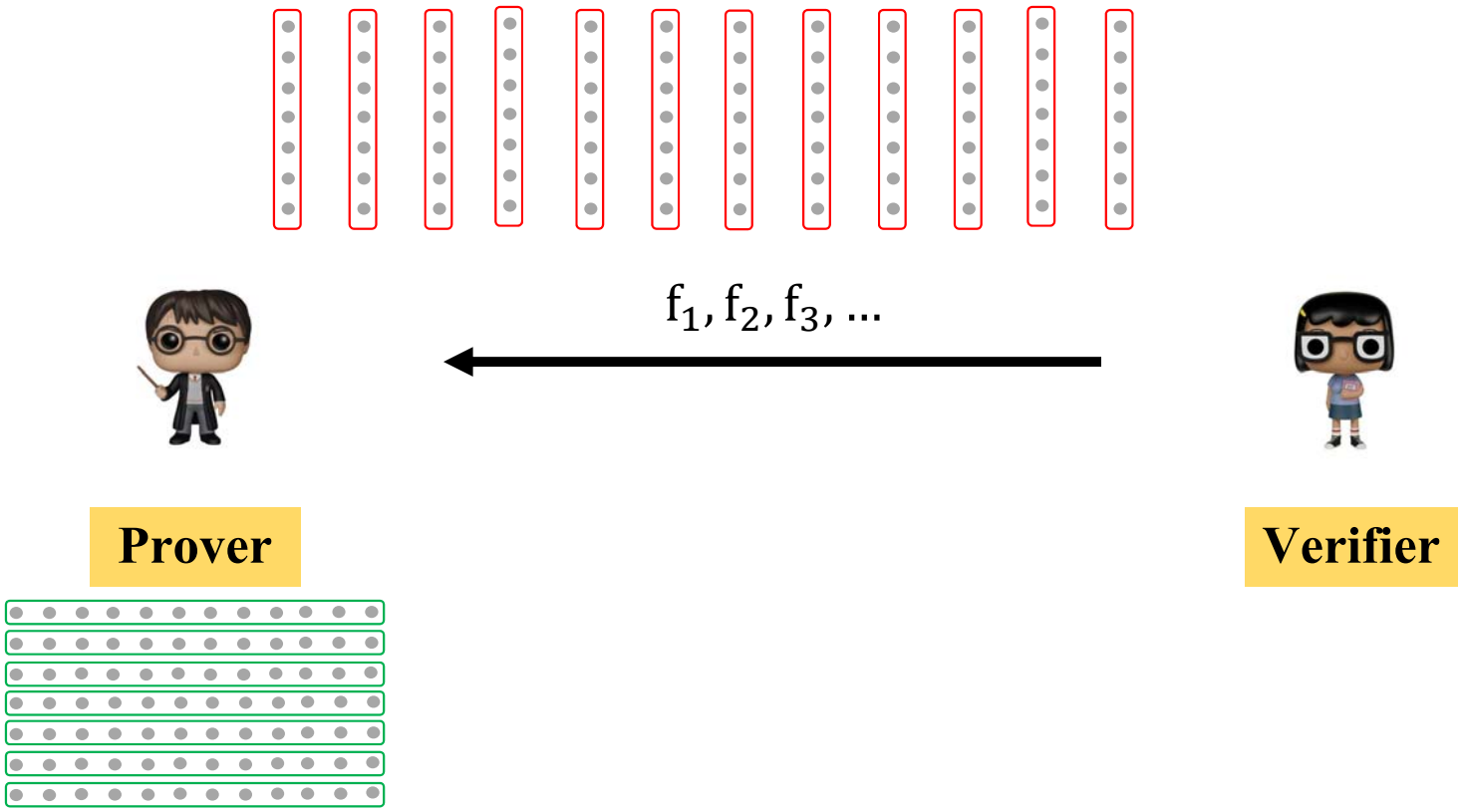


Verifier

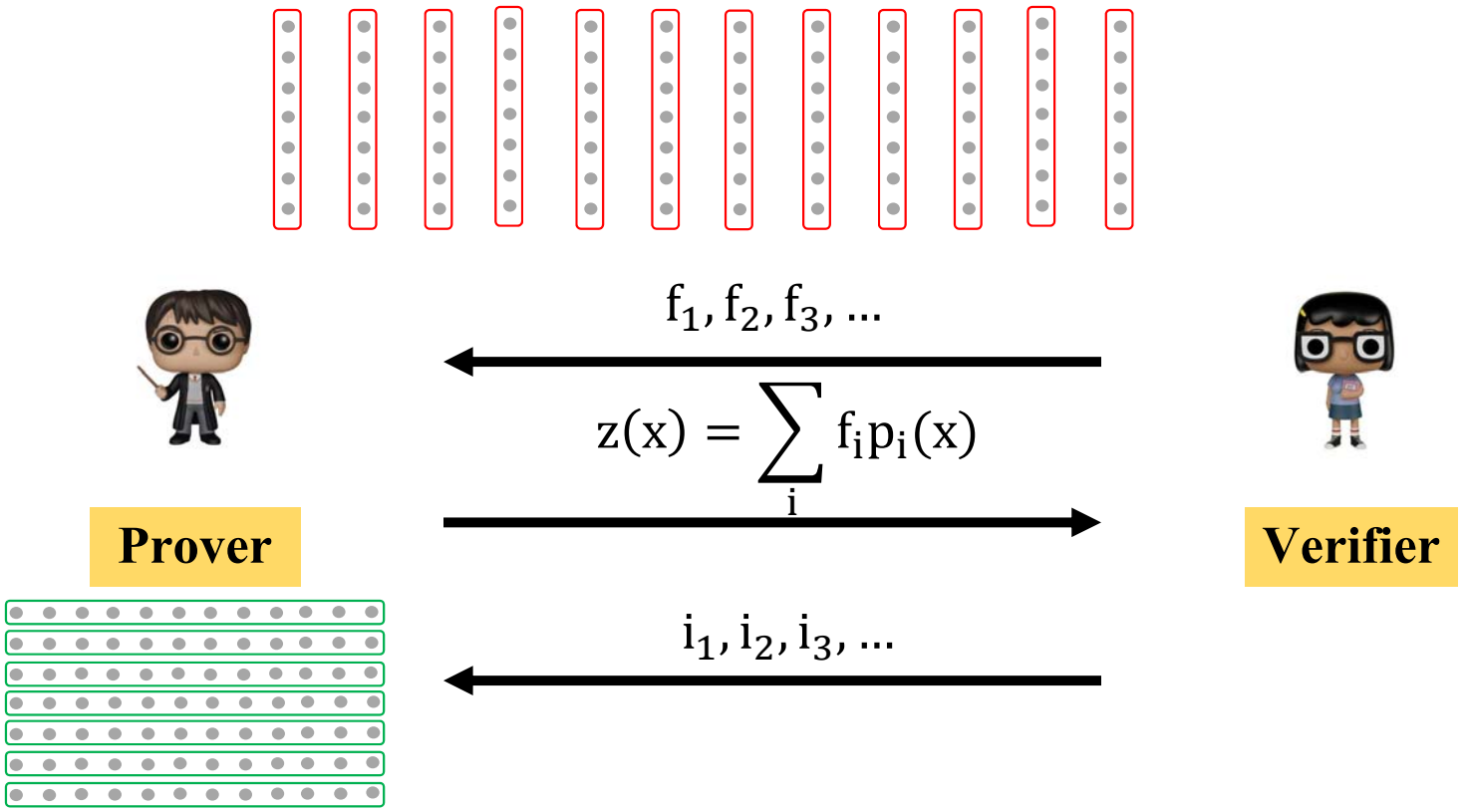
Idea 2: Testing Interleaved RS Codes



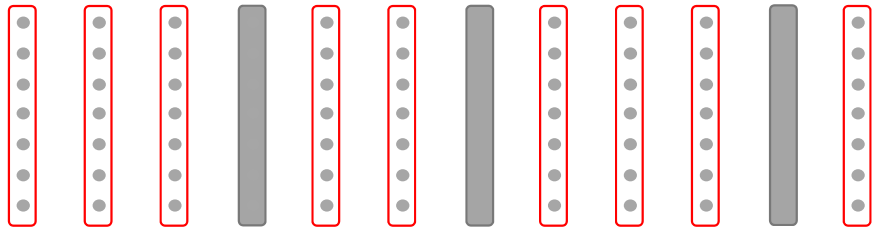
Idea 2: Testing Interleaved RS Codes



Idea 2: Testing Interleaved RS Codes



Idea 2: Testing Interleaved RS Codes

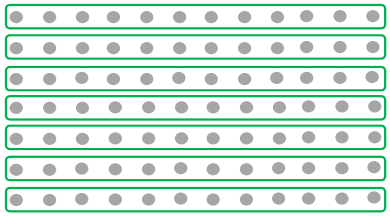


$f_1, f_2, f_3, \ldots$

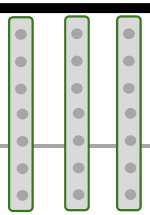
$z(x) = \sum_i f_i p_i(x)$

Prover

Verifier



$i_1, i_2, i_3, \ldots$



Check

- $z(x)$ is of degree $t+\ell$
- $z(i) = \sum_i f_i p_i(i)$

Idea 3: Testing Quadratic Constraints

Prover

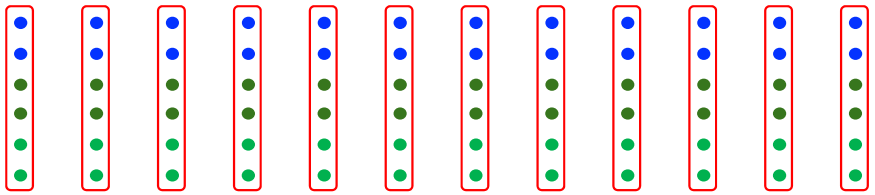
Verifier

Idea 3: Testing Quadratic Constraints

Prover

Verifier

Idea 3: Testing Quadratic Constraints

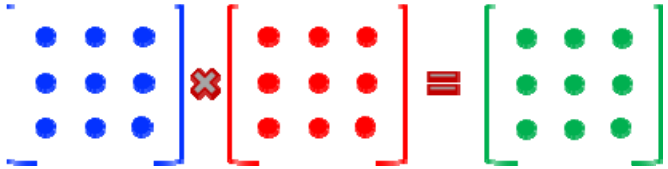


$f_1, f_2, f_3, \ldots$

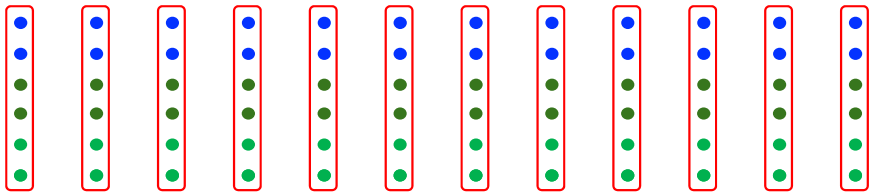
$$z(x) = \sum_i f_i(p_i(x)q_i(x) - r_i(x))$$

Prover

Verifier



Idea 3: Testing Quadratic Constraints



$f_1, f_2, f_3, \ldots$

$$z(x) = \sum_i f_i(p_i(x)q_i(x) - r_i(x))$$

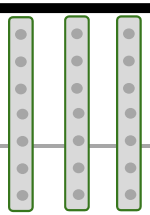
Prover

Verifier

$i_1, i_2, i_3, \ldots$

Check

$$z(i) = \sum_i f_i(p_i(i)q_i(i) - r_i(i))$$



Post-Quantum Signatures from NIZK [CDGORRSZ17,KKW18]

Obtaining (Post Quantum) Signatures from NIZK

The signature scheme:

PK: $y=PRF_k(0^k)$ where PRF is a block cipher

Sig(m): a proof for (y,k) on a challenge H(a,m)

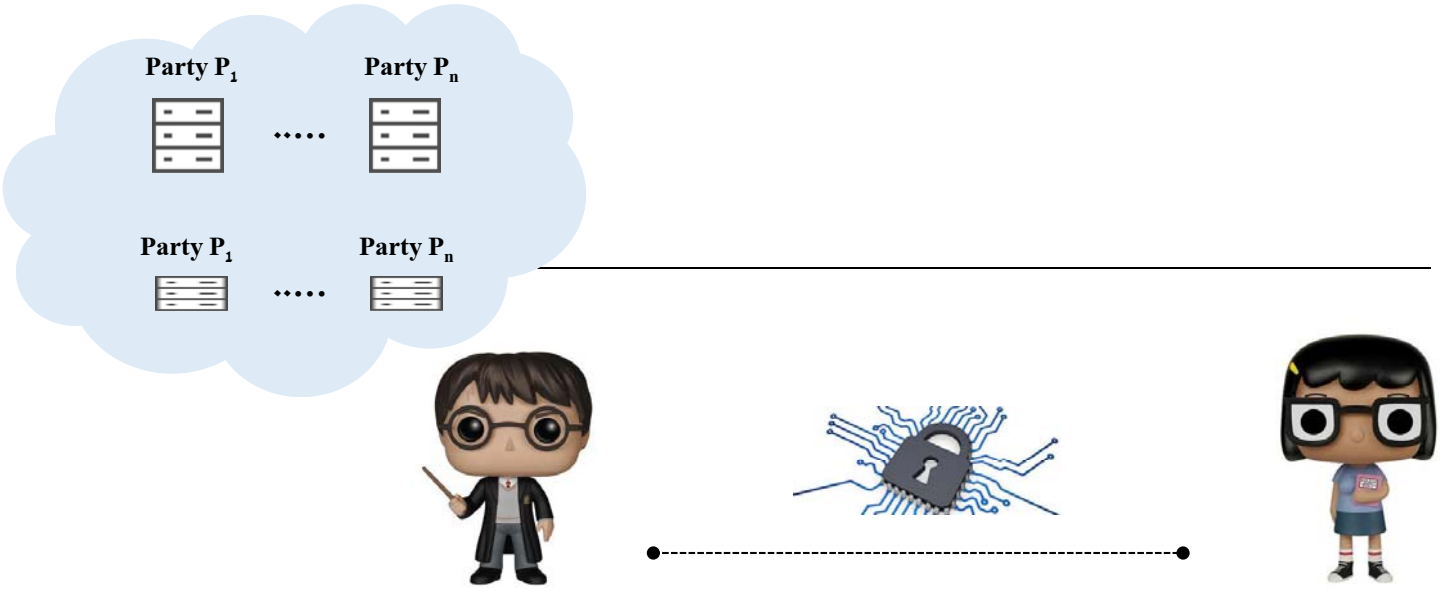
Advantages:

- Based on symmetric-key primitives
- Easily extendable to ring and group signatures

High-Level Overview [KKW18]

Use MPC-in-the-head in the **preprocessing model**

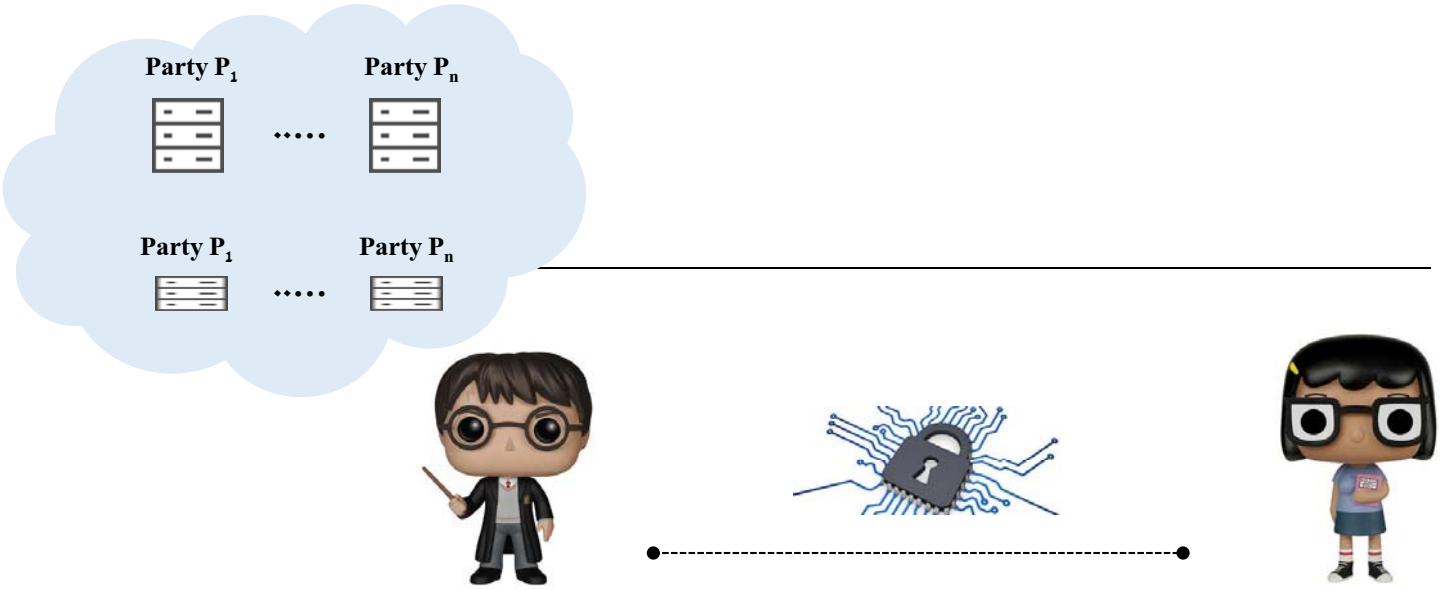
- Check consistency of preprocessing using cut-and-choose



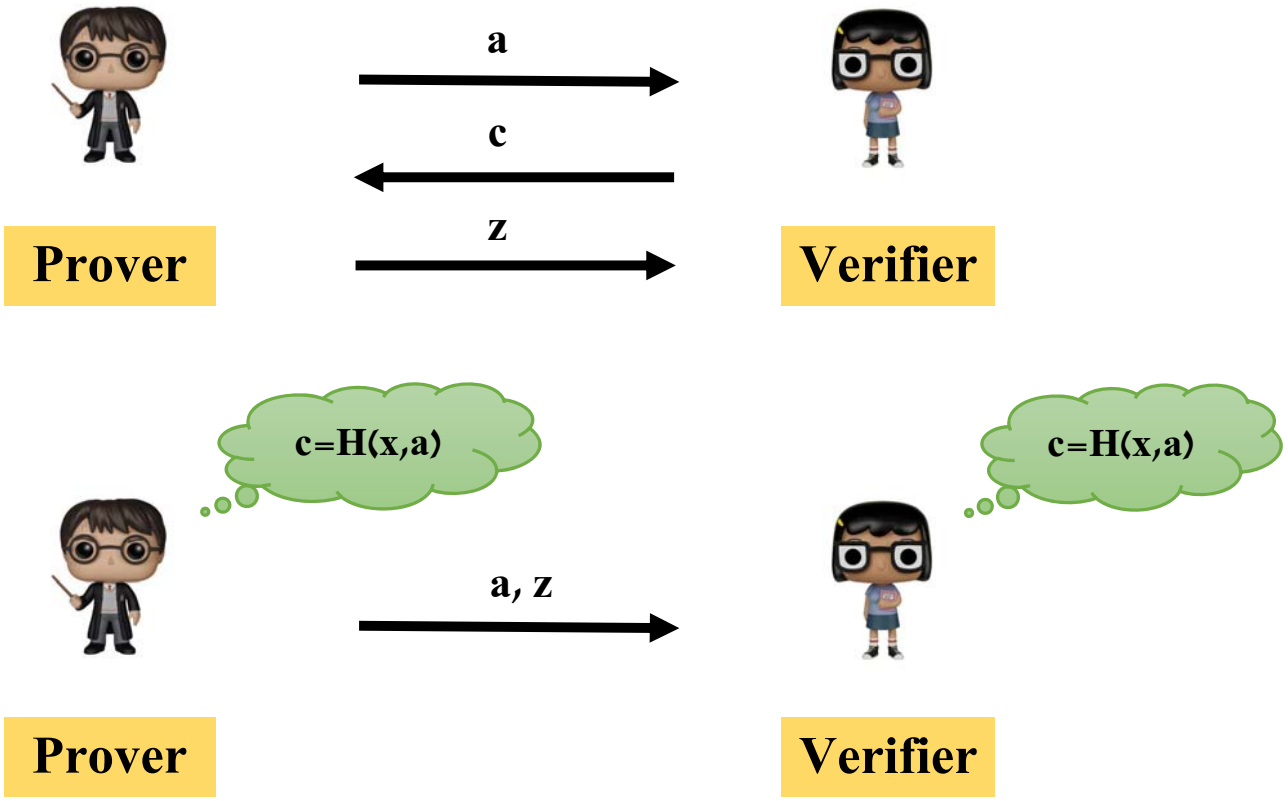
High-Level Overview [KKW18]

MPC-in-the-head can be instantiated with dishonest majority protocols

- Semi-honest instances for generating correlated randomness
- Implies two versions of 5/3 rounds



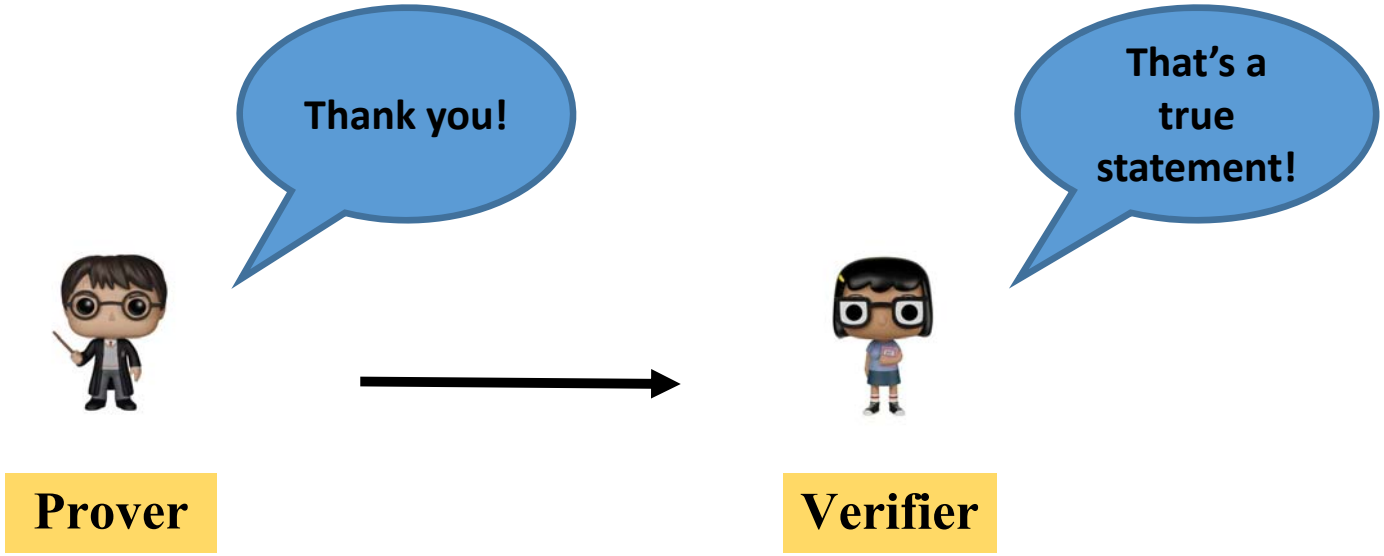
Removing Interaction via the Fiat-Shamir Transform



Analysis can be extended to any constant round public-coin protocol and beyond [BCS16]

Scalable Transparent Proofs (STARK,Aurora)

- Proof length and round complexity scale with $\log |C|$ [BBHR18,BCRSVW18]
- Prover's running time better in Ligero


Thank you!
That's a true statement!
Prover
Verifier