# Sub-Linear
# Lattice-Based
# Zero-Knowledge Arguments
# for Arithmetic Circuits

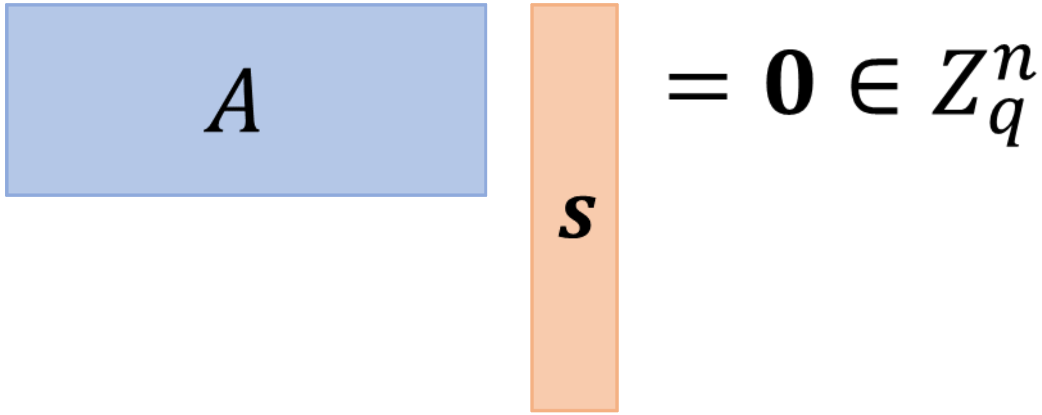
Bar-Ilan University
UCL IBM Research

Carsten Baum, Jonathan Bootle, Andrea Cerulli,
Rafael del Pino, Jens Groth and Vadim Lyubashevsky

# Lattice-Based Zero-Knowledge Arguments for Arithmetic Circuits

Short Integer Solution (SIS) Problem

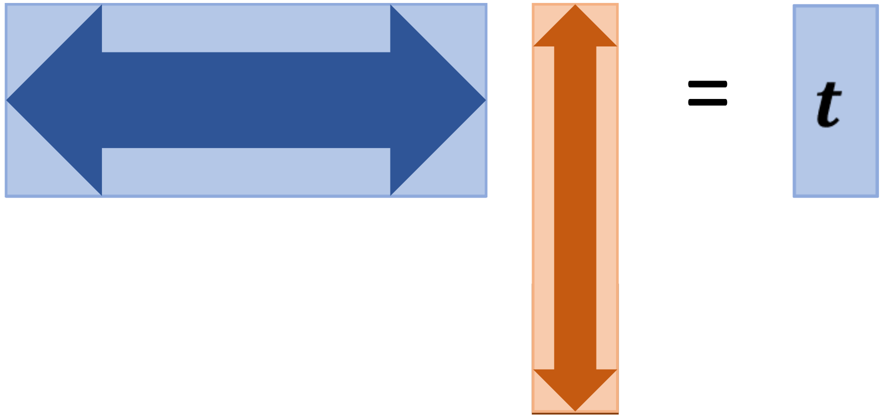
- Input: Random matrix $\mathrm{A} \in Z_q^{n \times m}$
- Goal: Find non-trivial $\boldsymbol{s} \in Z^m$ with $A\boldsymbol{s} = 0 \bmod q$ and $||s||_\infty < \beta$



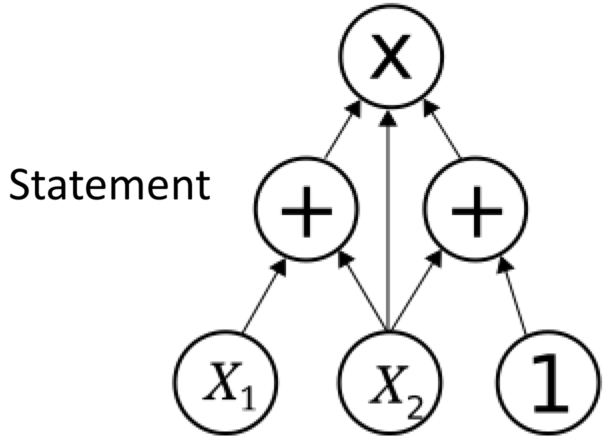
# Lattice-Based
Zero-Knowledge Arguments
for Arithmetic Circuits

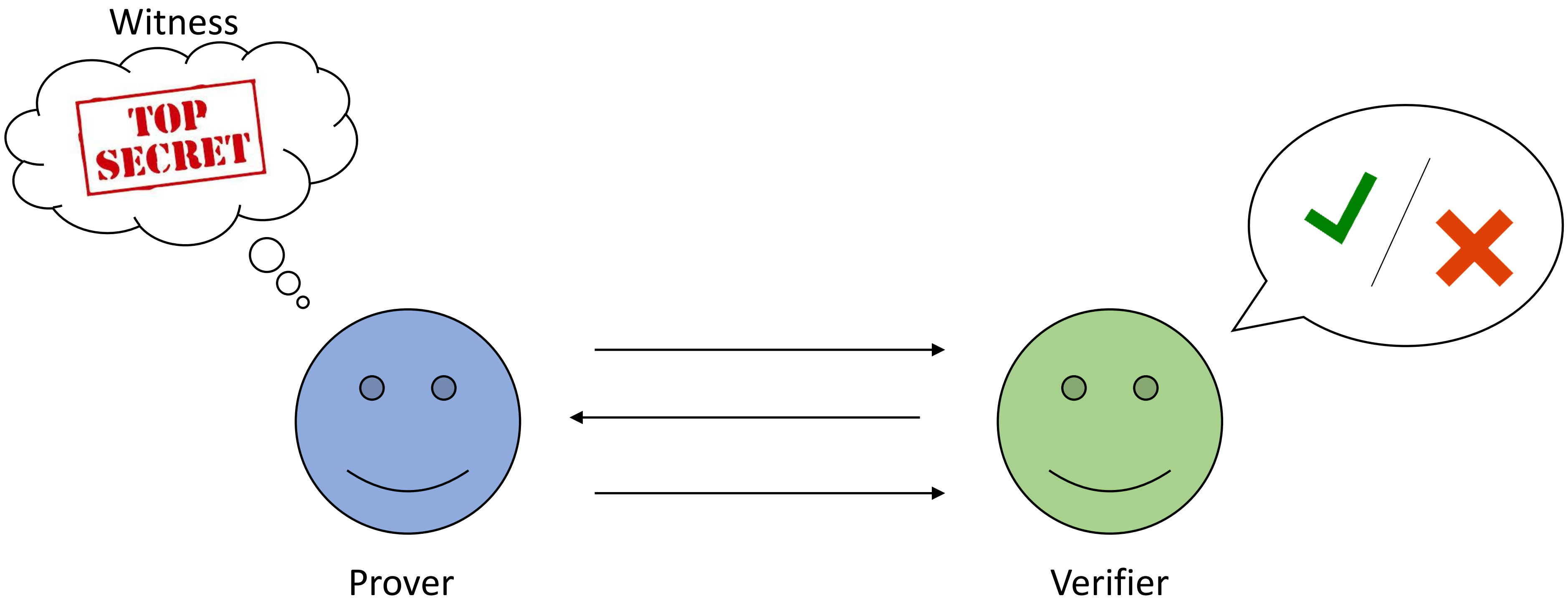
Commitment/hash from SIS:

- Binding/collision resistant by SIS
- Hiding by Leftover Hash Lemma
- Homomorphic
- Compressing [A96]



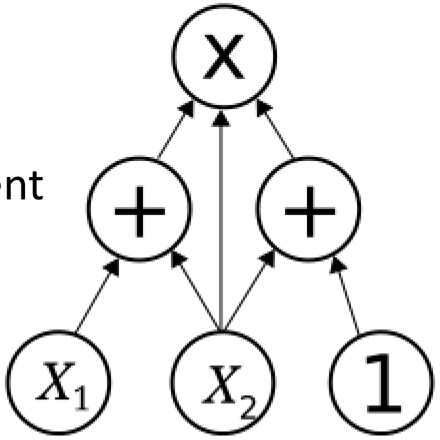
# Lattice-Based Zero-Knowledge Arguments for Arithmetic Circuits



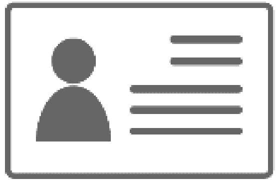



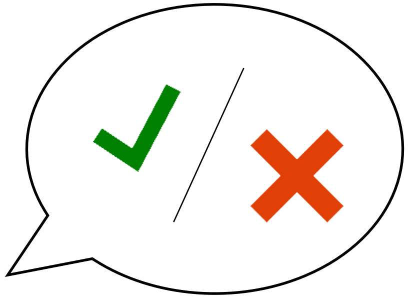
# Lattice-Based Zero-Knowledge Arguments for Arithmetic Circuits

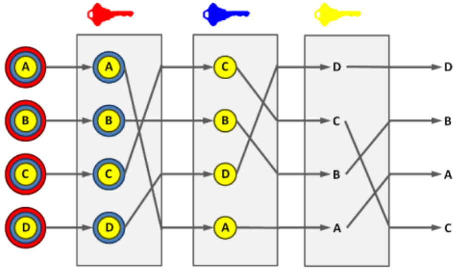
Statement

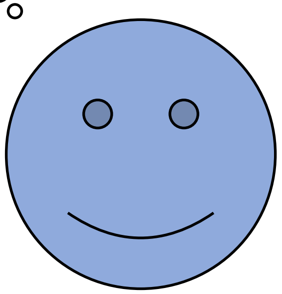


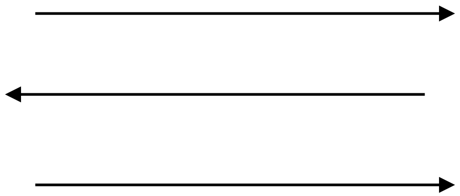
Witness

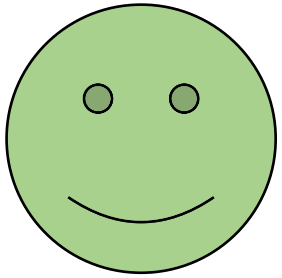



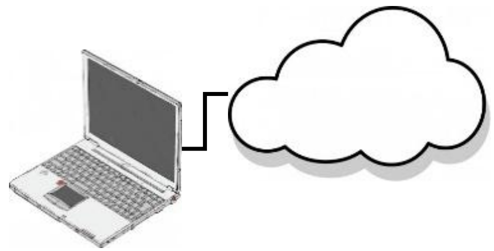












Prover

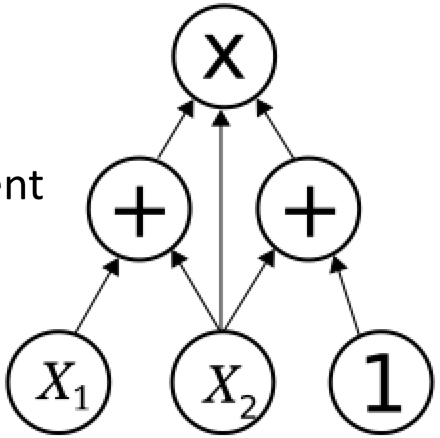
Verifier

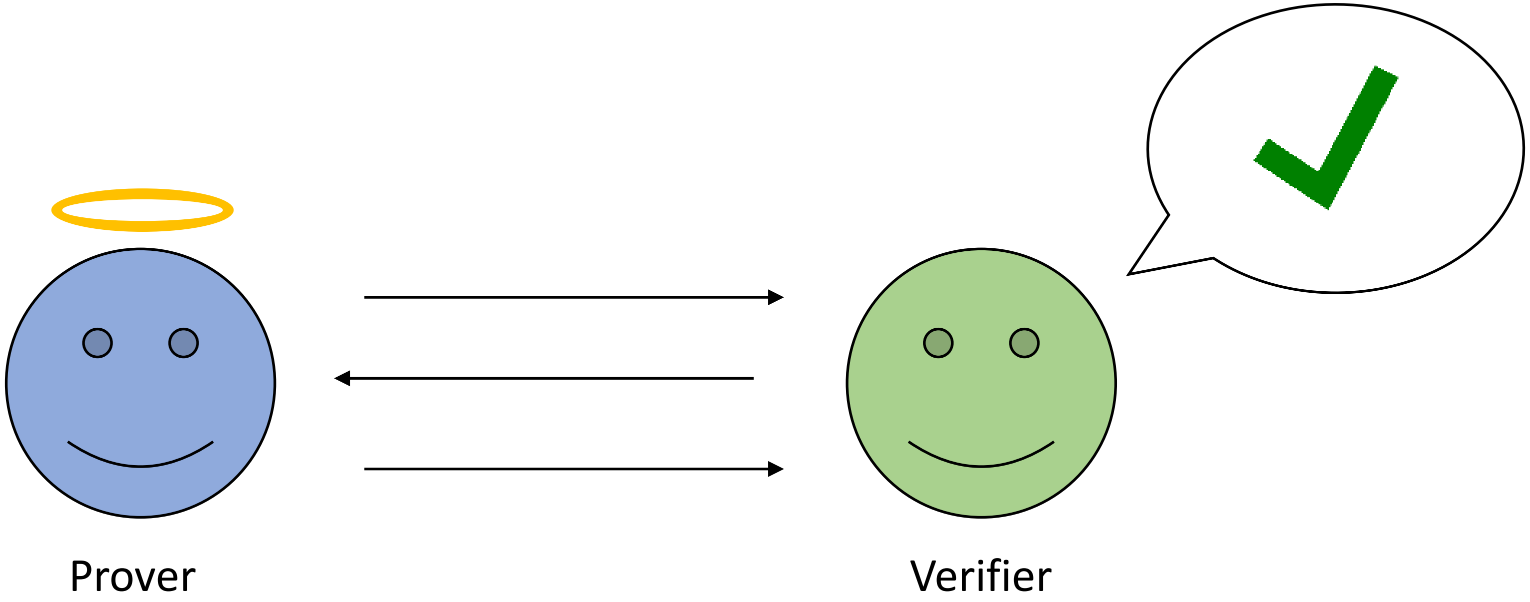
Lattice-Based

# Zero-Knowledge Arguments

for Arithmetic Circuits

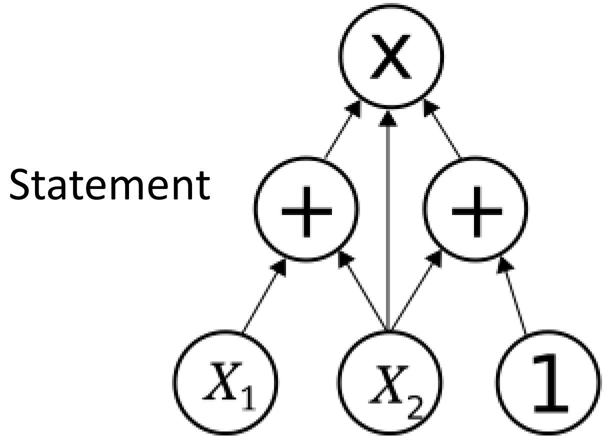
Statement

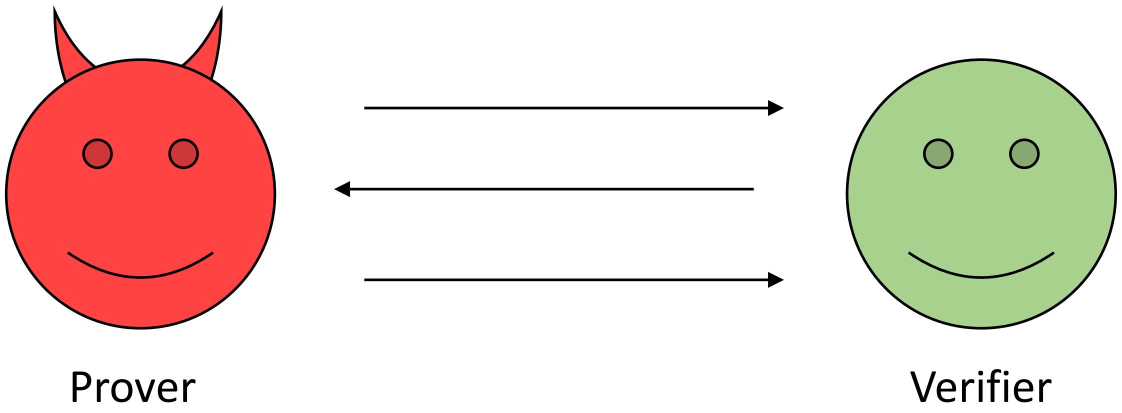


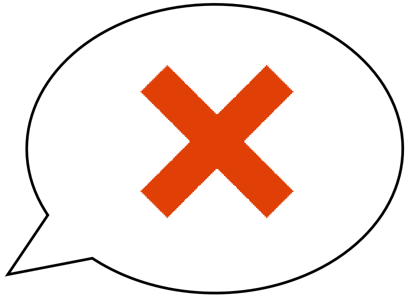



Completeness:
An honest prover
convinces the verifier.

# Lattice-Based Zero-Knowledge Arguments for Arithmetic Circuits



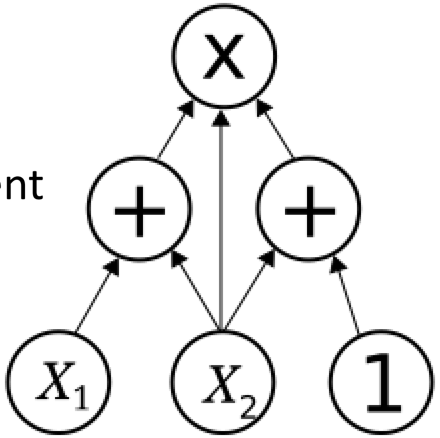
Completeness:
An honest prover convinces the verifier.

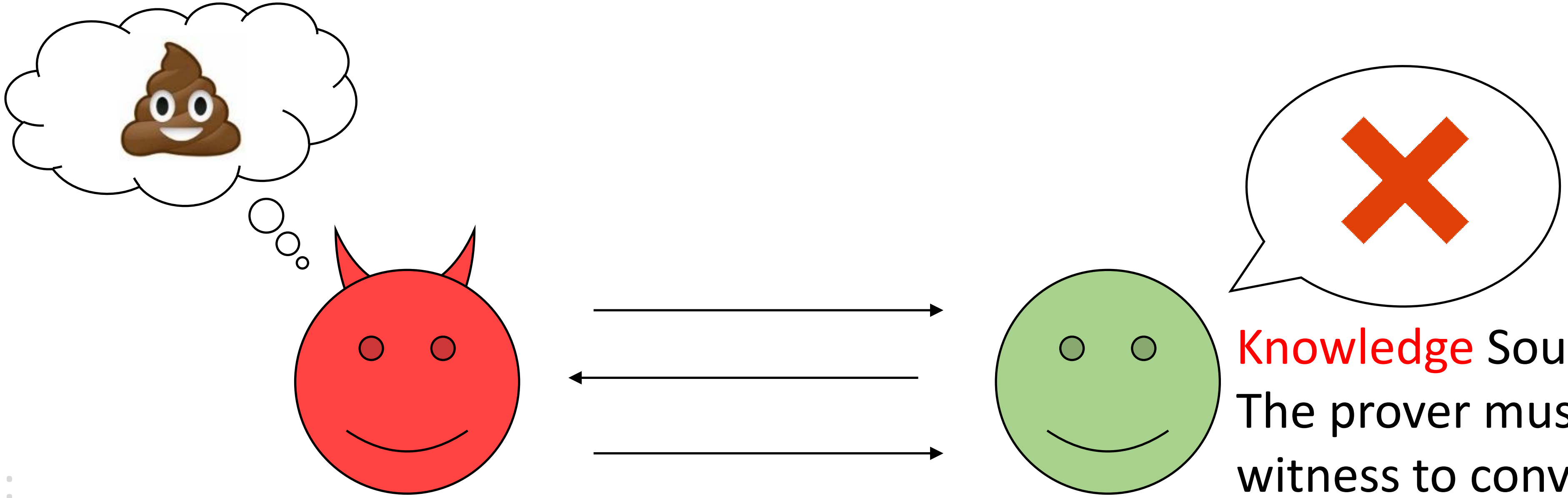
Soundness:
A dishonest prover never convinces the verifier.

Computational guarantee
-> argument

# Lattice-Based
# Zero-Knowledge Arguments
# for Arithmetic Circuits

Statement



Prover

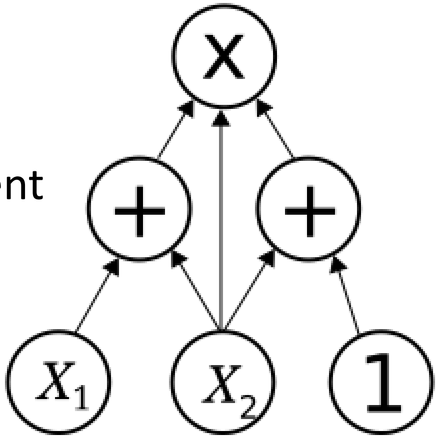
Verifier

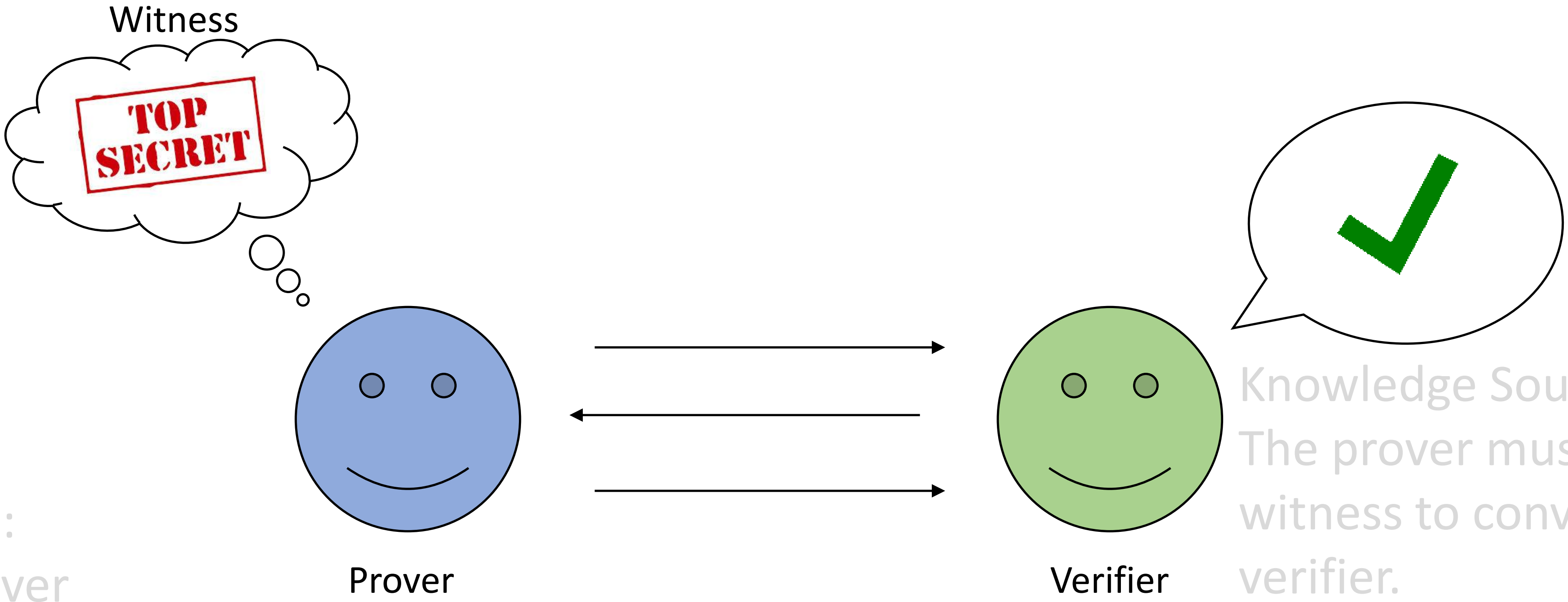
Completeness:
An honest prover
convinces the verifier.

Knowledge Soundness:
The prover must know a witness to convince the verifier.
-> Proof/argument of knowledge

# Lattice-Based
# Zero-Knowledge Arguments
# for Arithmetic Circuits

Statement




Knowledge Soundness:
The prover must know a witness to convince the verifier.
-> Proof/argument of knowledge

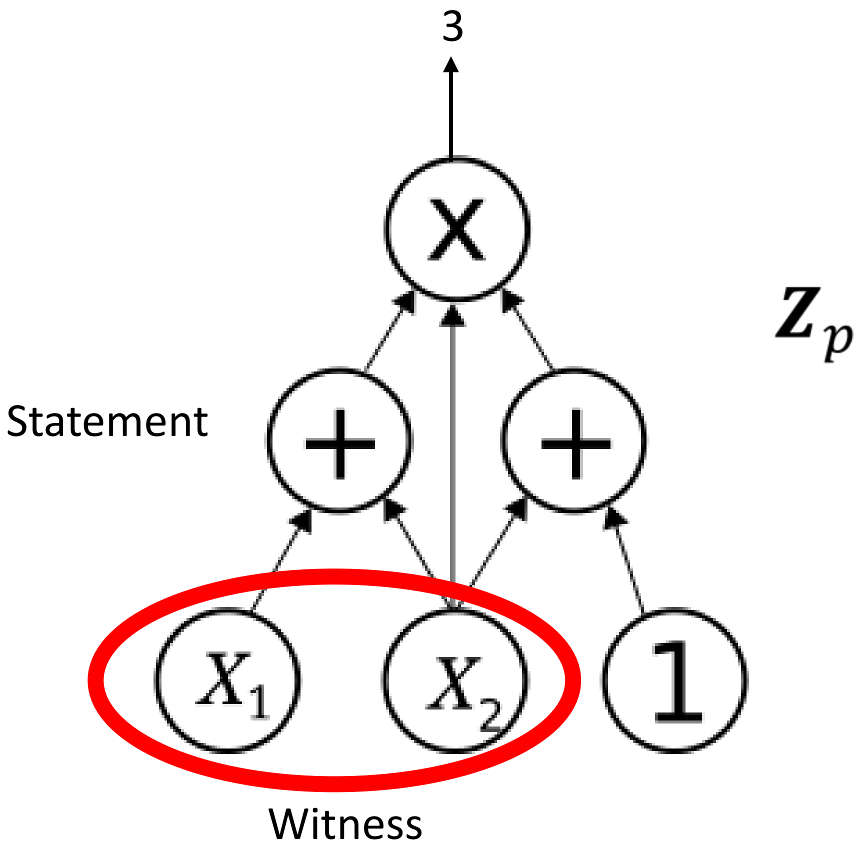
Completeness:
An honest prover convinces the verifier.

Zero-knowledge:
Nothing but the truth of the statement is revealed.

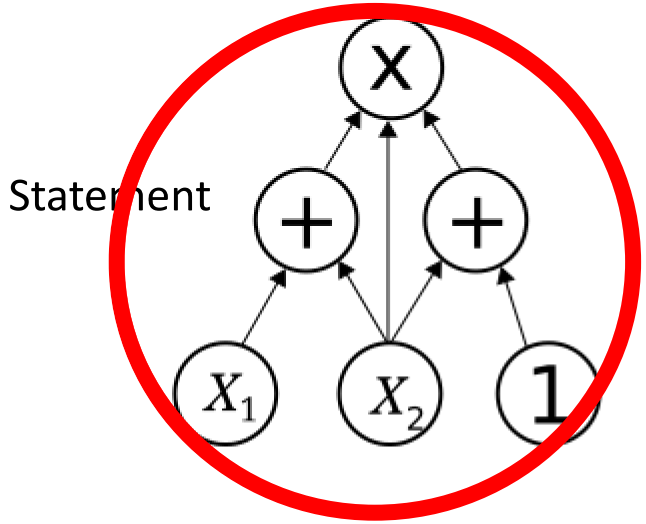
# Lattice-Based
# Zero-Knowledge Arguments
# for Arithmetic Circuits

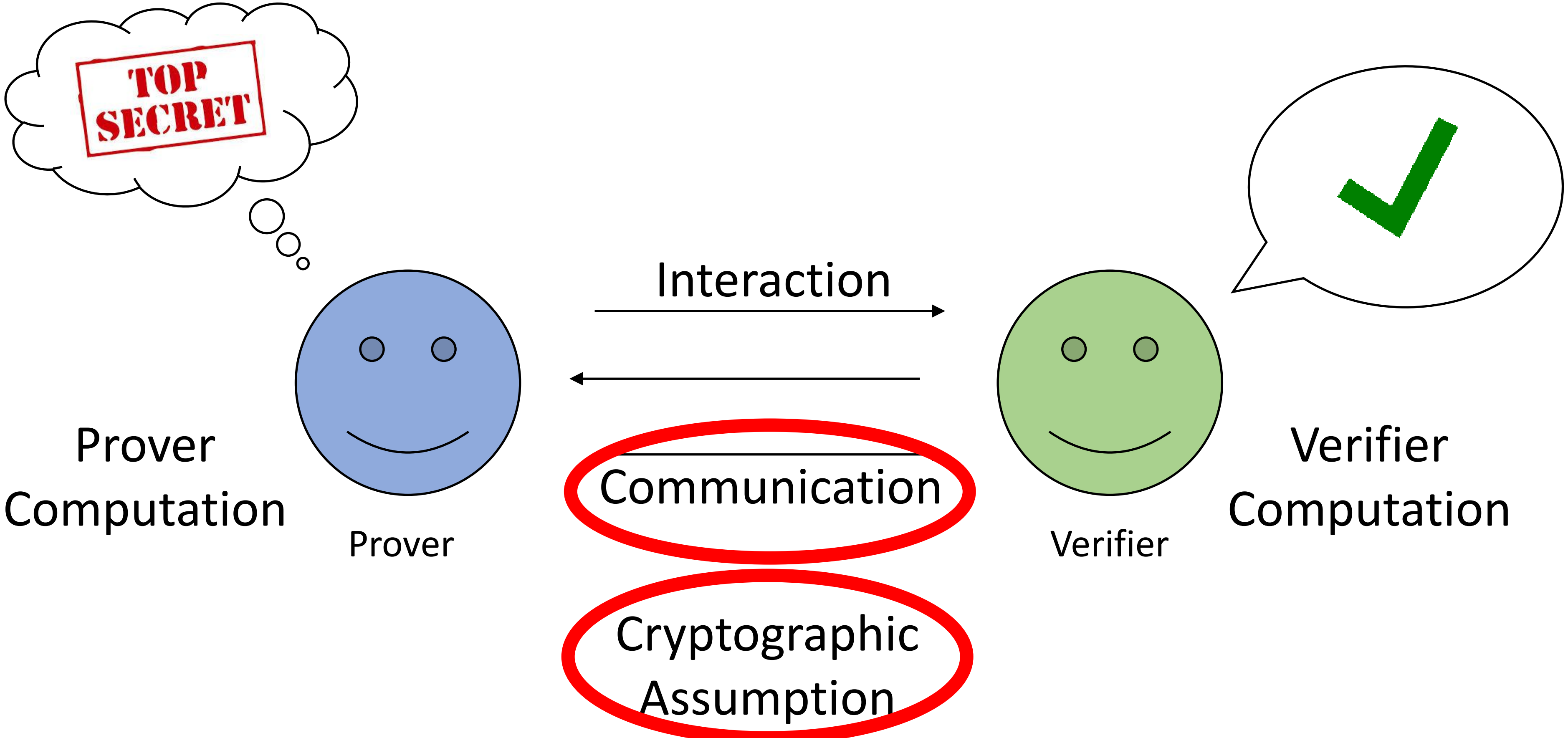
Why arithmetic circuits?

- C to circuit compilers
- Models cryptographic computations
- Witness existence? NP-Complete



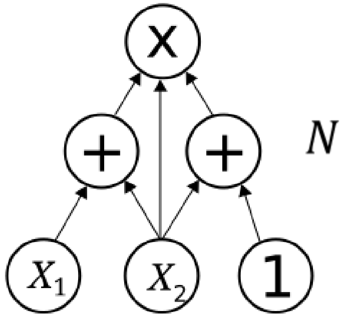
# Lattice-Based Zero-Knowledge Arguments for Arithmetic Circuits



# Results Table

| | Expected # Moves | Communication | Prover Complexity | Verifier Complexity |
|---|---|---|---|---|
| [DL12] | $O(1)$ | $O(N\lambda)$ | $O(N\mathrm{polylog}(\lambda)\ )$ | $O(N\mathrm{polylog}(\lambda))$ |
| [BKLP15] | $O(1)$ | $O(N\lambda)$ | $O(N\mathrm{polylog}(\lambda))$ | $O(N\mathrm{polylog}(\lambda))$ |
| This Work | $O(1)$ | $O\left(\sqrt{N\lambda\log^3 N}\right)$ | $O\left(N \log N\ (\log^2\lambda)\right)$ | $O(N\log^3\lambda)$ |




$N$ gates

Security parameter $\lambda$

# Arithmetic Circuit Argument

The interesting parts

Extension Fields

Proof of Knowledge

Rejection Sampling

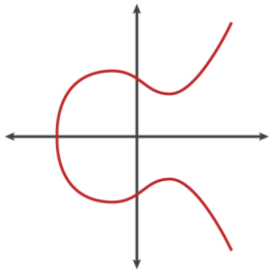
Arithmetic Circuits

Matrix Equations

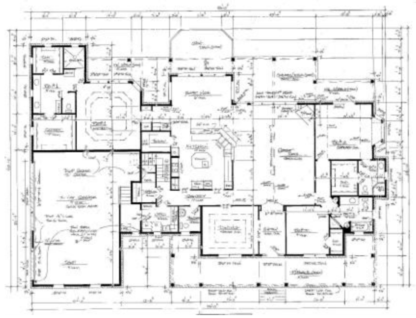
Polynomials

Commitments

Protocol



Featured in prior works
DLOG Protocols
Information Theoretic Proofs



# Proof of Knowledge

Statement

$$A\boldsymbol{s} = \boldsymbol{t}$$

Witness

# Proof of Knowledge

$$\underbrace{A\boldsymbol{s}_1 = \boldsymbol{t}_1 \qquad A\boldsymbol{s}_2 = \boldsymbol{t}_2 \qquad \cdots \qquad A\boldsymbol{s}_m = \boldsymbol{t}_m}$$

$m \approx \sqrt{N}$

$>\lambda$ preimages. Prior works: $O(\lambda^2)$

$\boldsymbol{s}_1 \approx \sqrt{N}$

->Prover knows $N$ small hashed integers

# Typical Proofs of Knowledge

Completeness: $A s = t$ $\qquad \|s\|_\infty < \beta$

Knowledge Soundness: $A s = 2 t$ $\qquad \|s\|_\infty < \kappa\beta$

None for us*

Soundness Slack

We get $O(\lambda)$

Proving $m$ statements requires $O(m)$-sized proofs

# Simplistic Protocol

$$As = t$$

P

V

$$Ay = w$$

$w$ $\longrightarrow$

$c \in \{0,1\}$ $\longleftarrow$

$$z = cs + y$$

Rejection Sampling

$z$ $\longrightarrow$

Check: $\|z\|_\infty < B$

$$Az = ct + w$$

# Our Protocol

$$z = \sum s_i c_i + y, \quad c_i \in \{0,1\}$$

$$z = s_1 c_1 + s_2 c_2 \dots + s_n c_m + y$$

$$z' = s_2 c_2 \dots + s_n c_m + y$$

Extraction with probability $\approx pr - 1/2$ for prover with success probability $pr$

# Our Protocol

$$\boxed{\boldsymbol{z}} = \sum \boxed{\boldsymbol{s}_i}\, \boldsymbol{c}_i^T + \boxed{\boldsymbol{y}} \qquad \boldsymbol{c}_i^T \in \{0,1\}^{O(\lambda)}$$

Extraction with probability $\approx pr - 1/2^{\lambda}$ for $\lambda$ parallel repetitions

- Communication scales like $\log(m)$, not $m$
- Minimum (commitment size) + (proof size) is $O(\sqrt{N})$ for circuit protocol

# Proof-of-Knowledge Performance

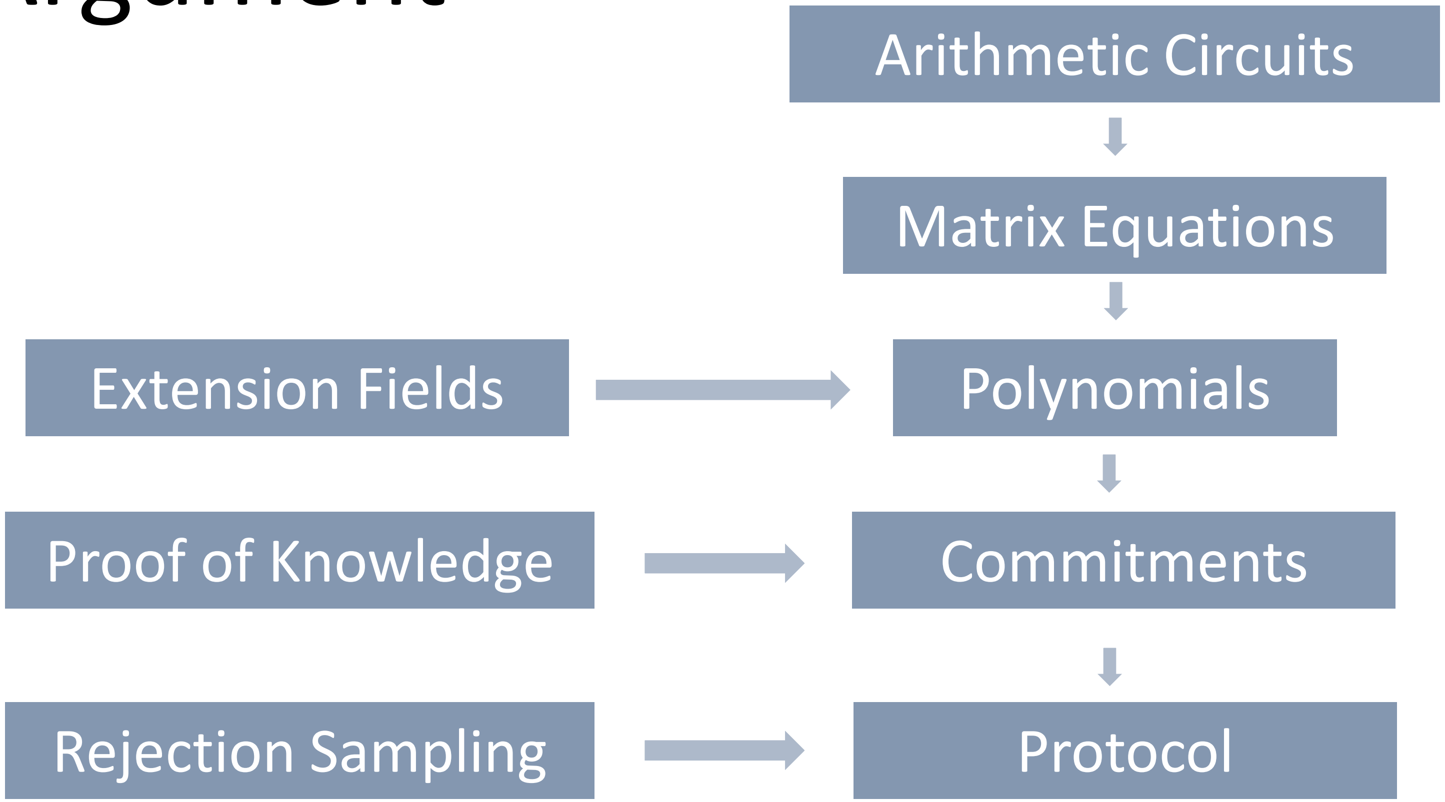
| | Expected # Moves | Communication | Prover Complexity | Verifier Complexity |
|---|---|---|---|---|
| [BDLN16] | $O(1)$ | $O(m)$ | $O(m)$ | $O(m)$ |
| [CDXY17] | $O(1)$ | $O(m)$ | $O(m)$ | $O(m)$ |
| This Work | $O(1)$ | $O(\lambda \log(m\lambda))$ | $O(m)$ | $O(m)$ |
| This Work | $O(1)$ | $O\left(\sqrt{N\lambda \log^3 N}\right)$ | $O(N\log^3 \lambda)$ | $O(\sqrt{N}\log^3 \lambda)$ |

$$A\,s = t$$

$N$ hashed integers,
m preimages

Security parameter $\lambda$

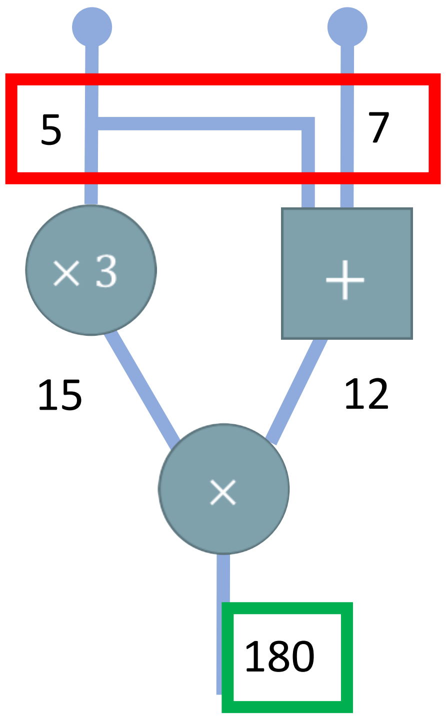
# Arithmetic Circuit Argument



# High Level Structure

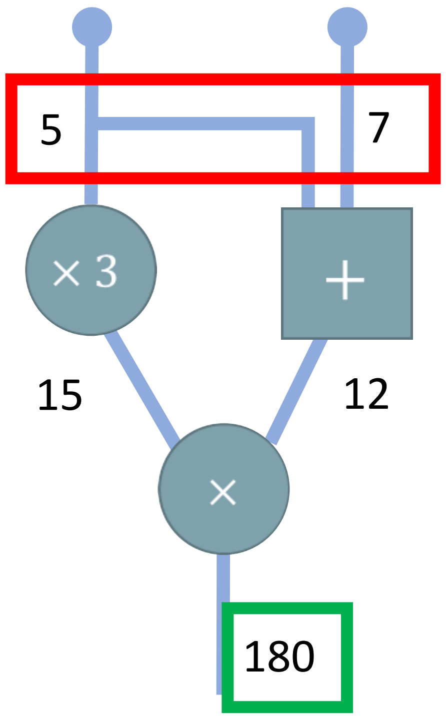
$$\begin{array}{ccccc} L & & R & & O \\ \boxed{\begin{array}{c}3\\15\end{array}} & \circ & \boxed{\begin{array}{c}5\\12\end{array}} & = & \boxed{\begin{array}{c}15\\180\end{array}} \end{array}$$

$$\boxed{5} + \boxed{7} = \boxed{12}$$

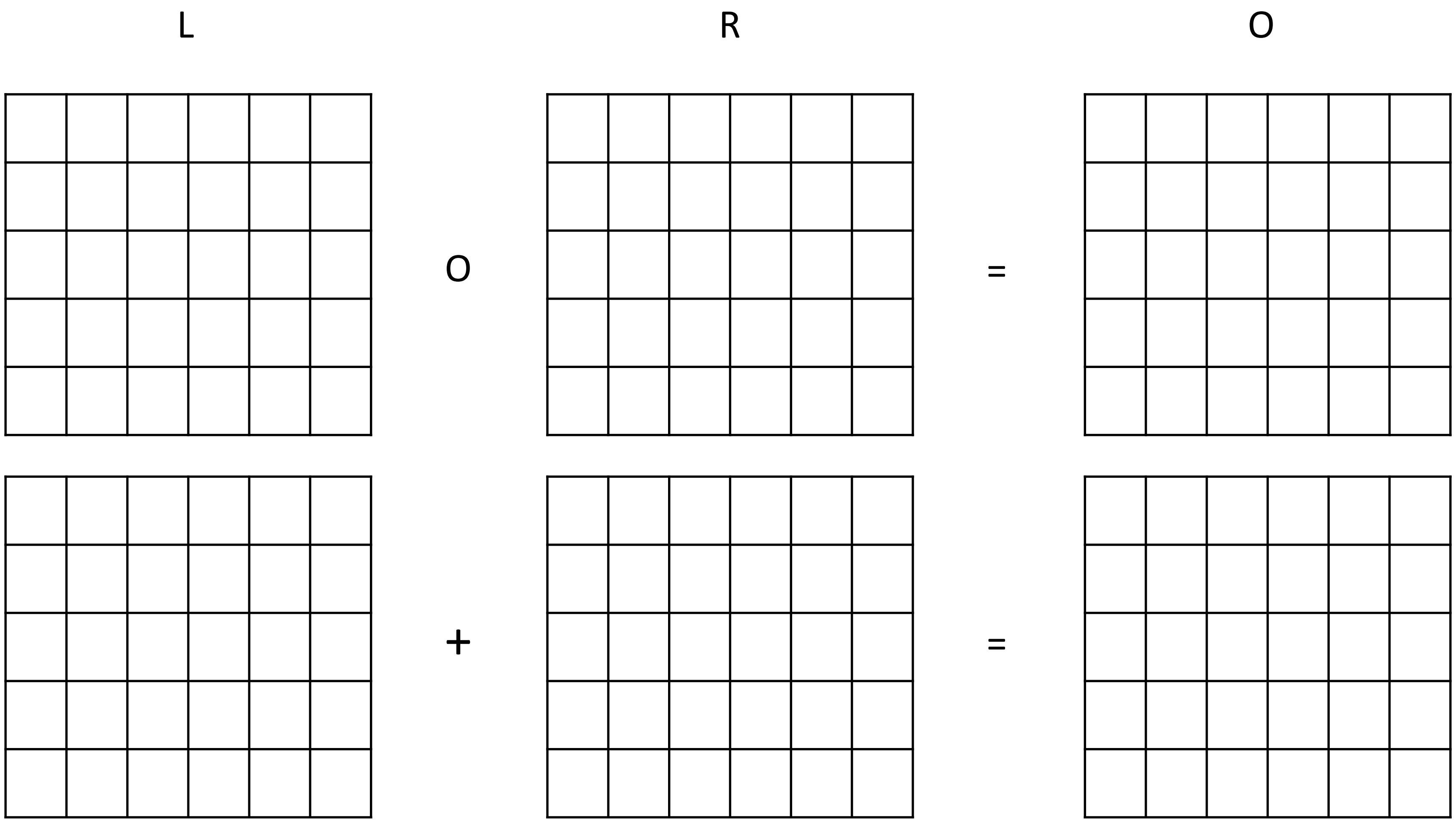


# High Level Structure

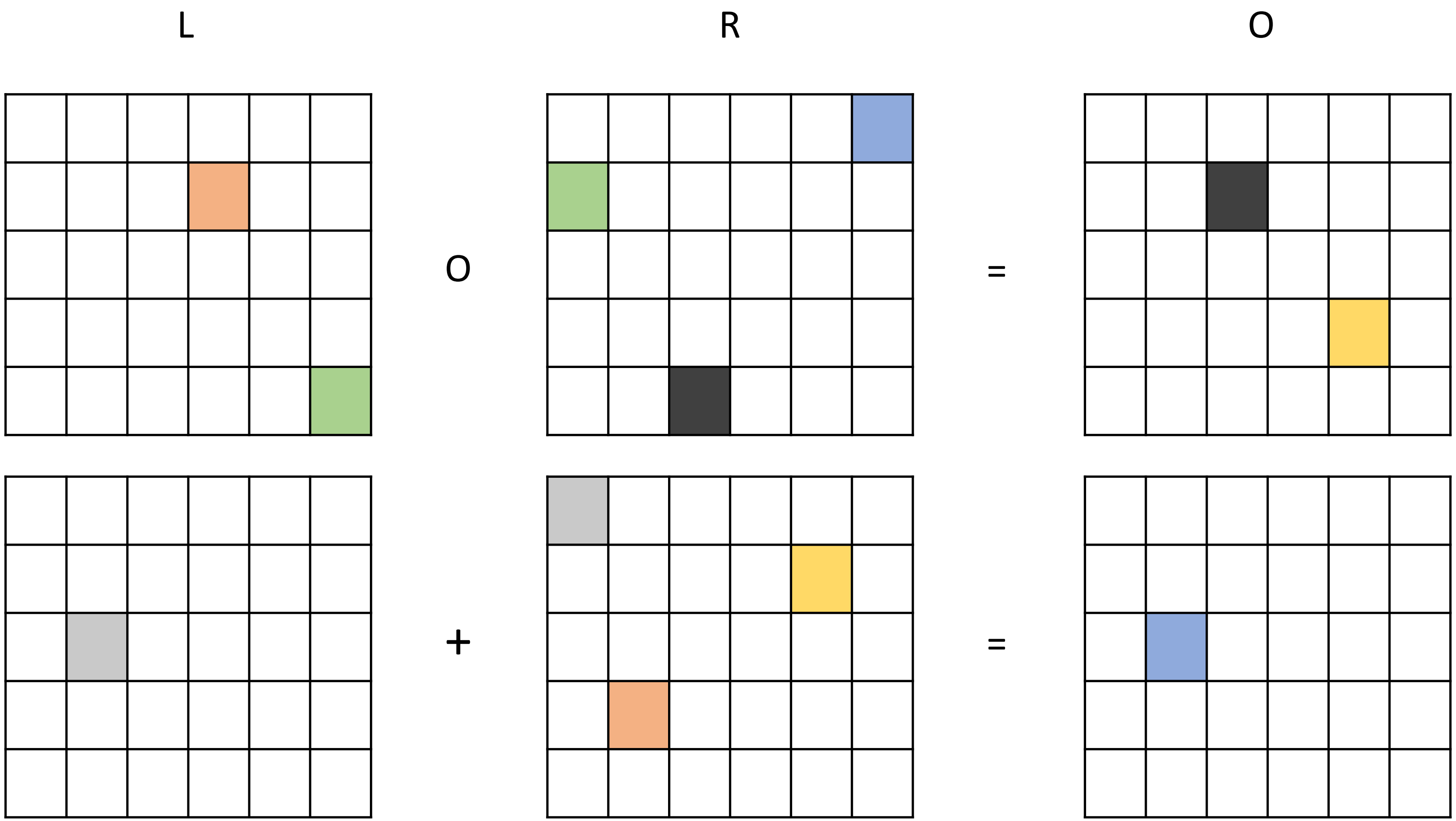
| L | | R | | O |
|---|---|---|---|---|
| 3 | o | 5 | = | 15 |
| 15 | | 12 | | 180 |
| 5 | + | 7 | = | 12 |



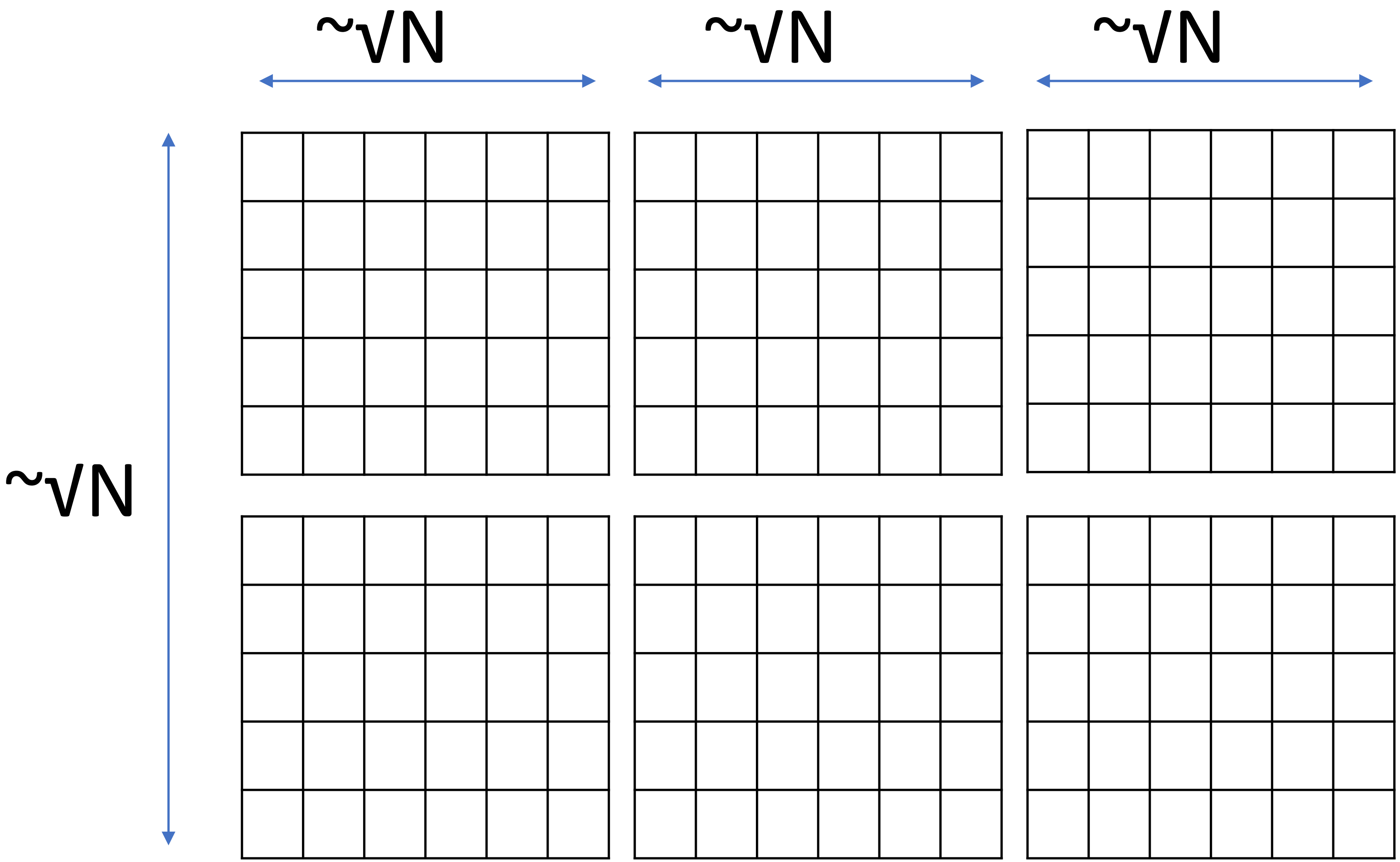
# High Level Structure



# High Level Structure



# Matrix Dimensions



# Paradigm from Previous Arguments

- Commit to vectors ([G09], [S09],[BCGGHJ17])
- Random challenge $x$
- Prover opens linear combinations
- Verifier conducts polynomial identity test
- AC-SAT in coefficients

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $3x$ | 2 | 6 | 6 | 2 | 0 | 1 | 9 | 2 | 7 | 4 |
| $+4x^2$ | 5 | 3 | 7 | 2 | 8 | 3 | 6 | 1 | 6 | 9 |
| $+8x^3$ | 5 | 7 | 6 | 7 | 1 | 4 | 2 | 6 | 8 | 3 |
| $+7x^4$ | 6 | 3 | 7 | 2 | 7 | 5 | 3 | 2 | 4 | 7 |
| = | 5 | 2 | 8 | 7 | 3 | 1 | 0 | 4 | 7 | 3 |

# Protocol Flow

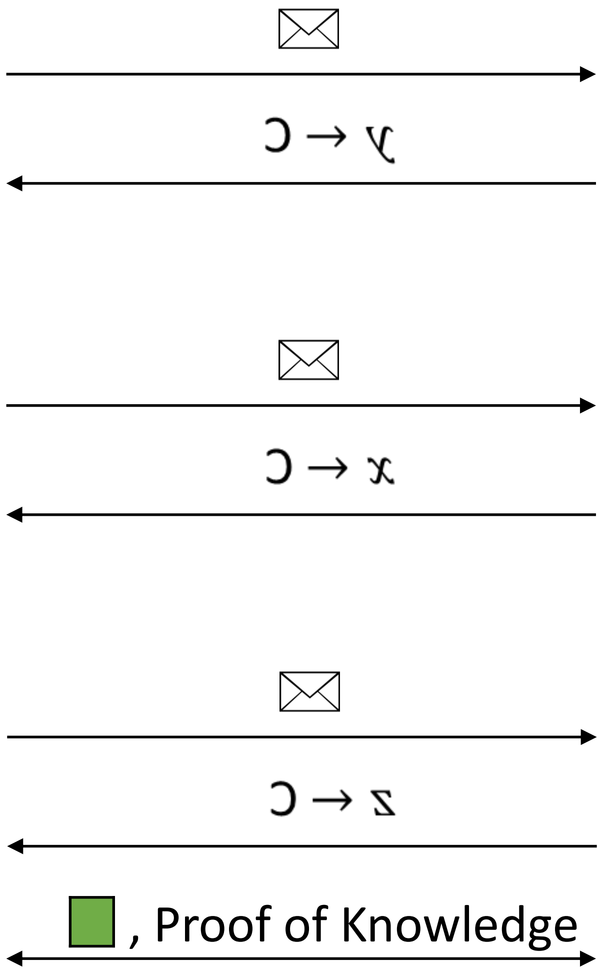
P

1. Commit to wire values

2. Commit to polynomial coefficients

3. Commit to mod p correction factors

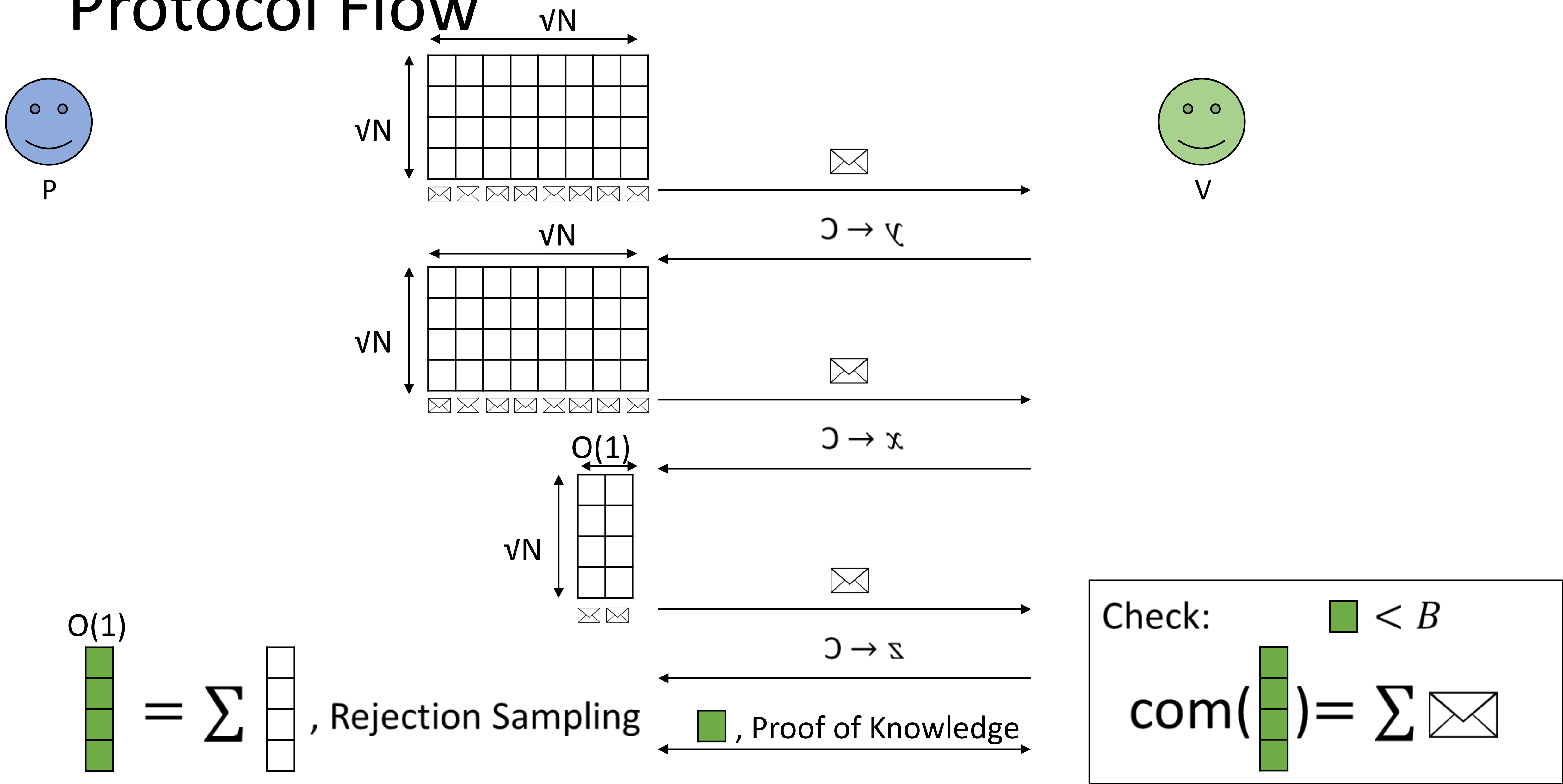
4. Compute linear combinations, do rejection sampling, proof of knowledge



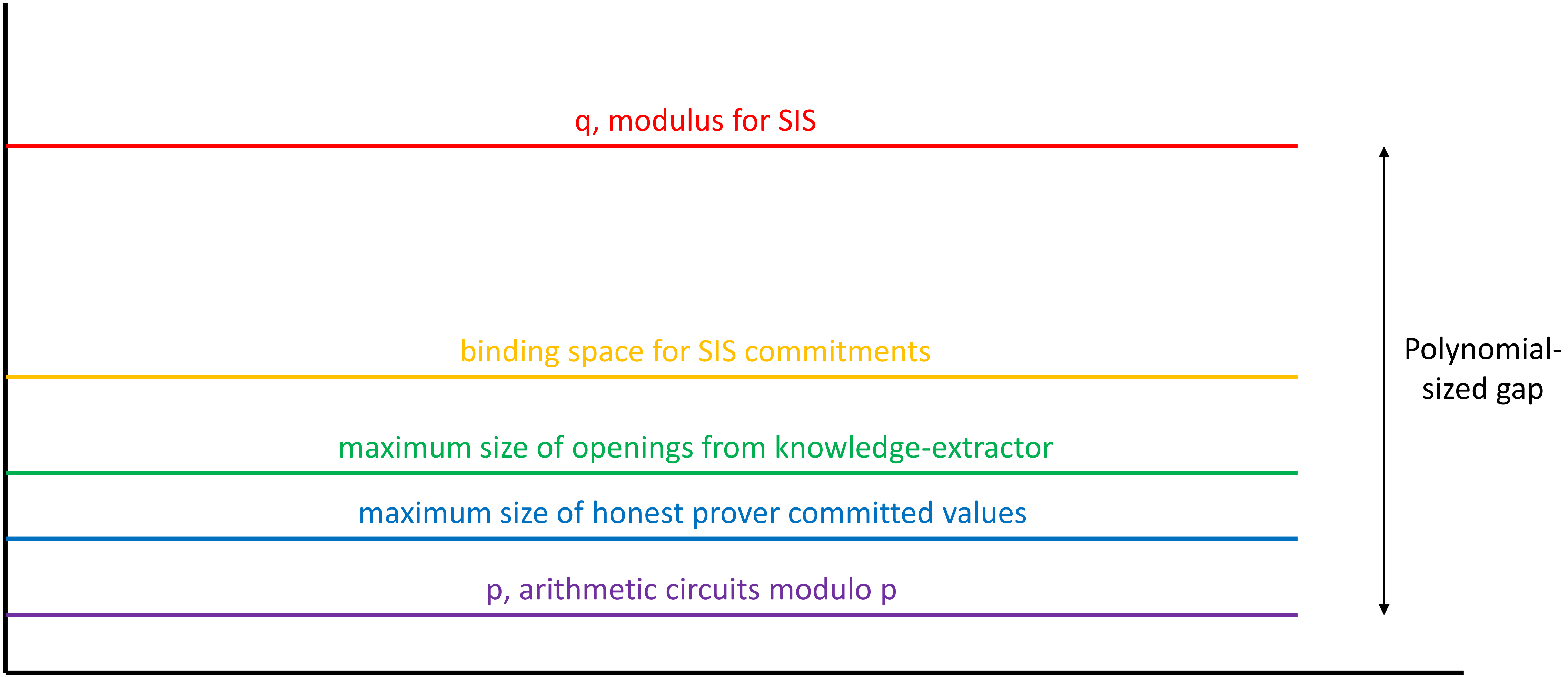

V

Check size bounds and linear combinations

# Protocol Flow



# Parameter Choice



# Additional Issues

- DLOG: $p \approx 2^{\lambda}$
- SIS: modulus usually $\mathrm{poly}(\lambda)$
- Use field extension techniques in $GF(p^k)$ building on [CDK14]
- Embed useful conditions into extension field operations

Schwarz-Zippel Lemma:

$$\underbrace{\Pr[f(r_1, r_2, \ldots, r_n) = 0] \leq \frac{d}{p^{k}}}$$

Not negligible!

Negligible!

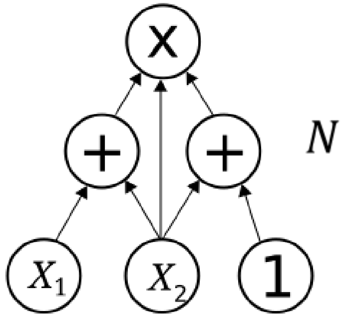
| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $\boldsymbol{a}$ | Empty | $\times$ | Empty | $\boldsymbol{b}$ | $=$ | Rubbish | $\boldsymbol{a} \cdot \boldsymbol{b}$ | Rubbish |

# Future Work:
# Can we match the $O(\log N)$ proof sizes of DLOG protocols?

# Thanks!

| Expected # Moves | Communication | Prover Complexity | Verifier Complexity |
|---|---|---|---|
| $O(1)$ | $O\left(\sqrt{N\lambda\log^3 N}\right)$ | $O(N \log N\ (\log^2\lambda))$ | $O(N\log^3\lambda)$ |

- General Statements
- Sub-linear proofs
- Relies on SIS




$N$ gates

Security parameter $\lambda$