# Verifiable Computation for Approximate Homomorphic Encryption Schemes

Ignacio Cascudo, Anamaria Costache, **Daniele Cozzo**, Dario Fiore,
Antonio Guimarães, Eduardo Soria-Vazquez

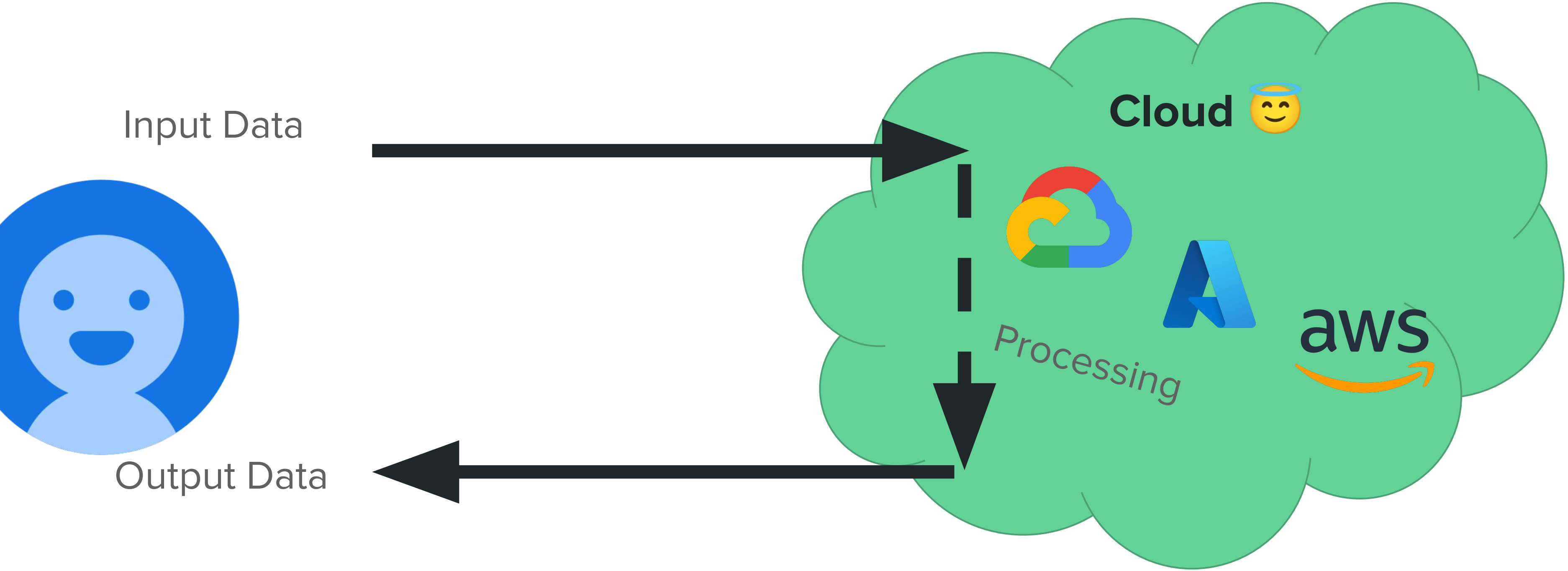
institute IMDEA software

NTNU
Norwegian University of Science and Technology

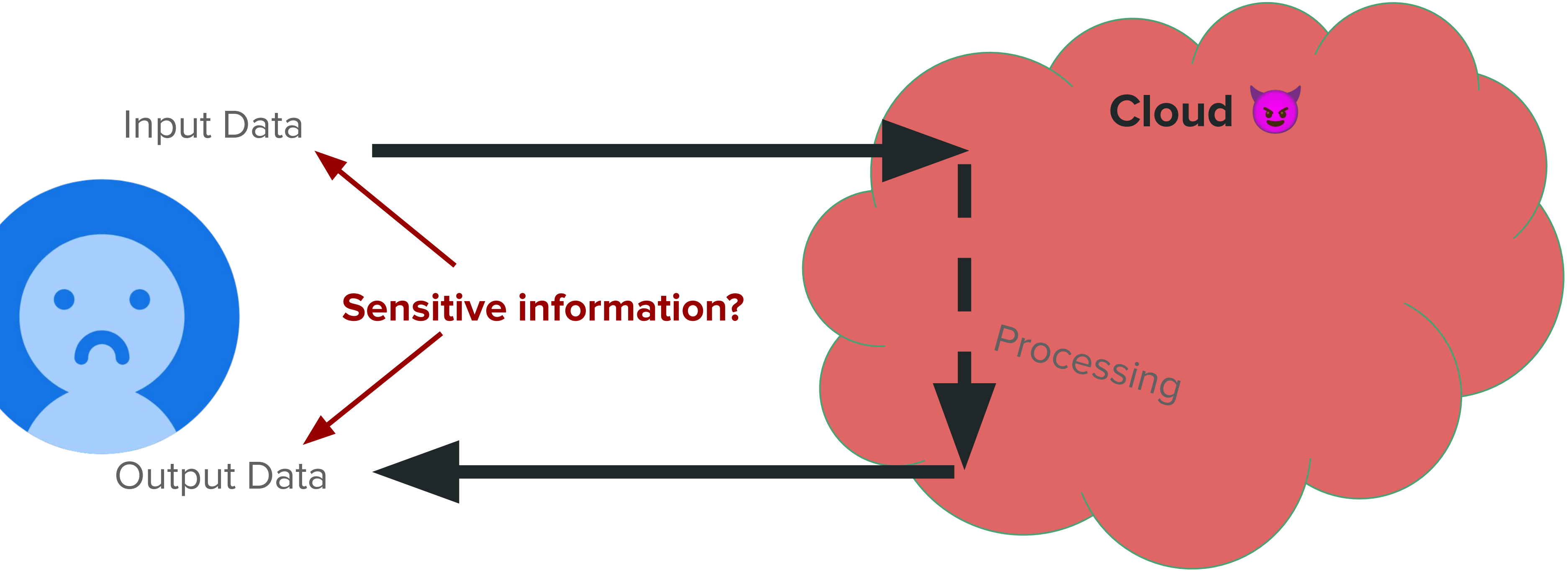
Technology Innovation Institute

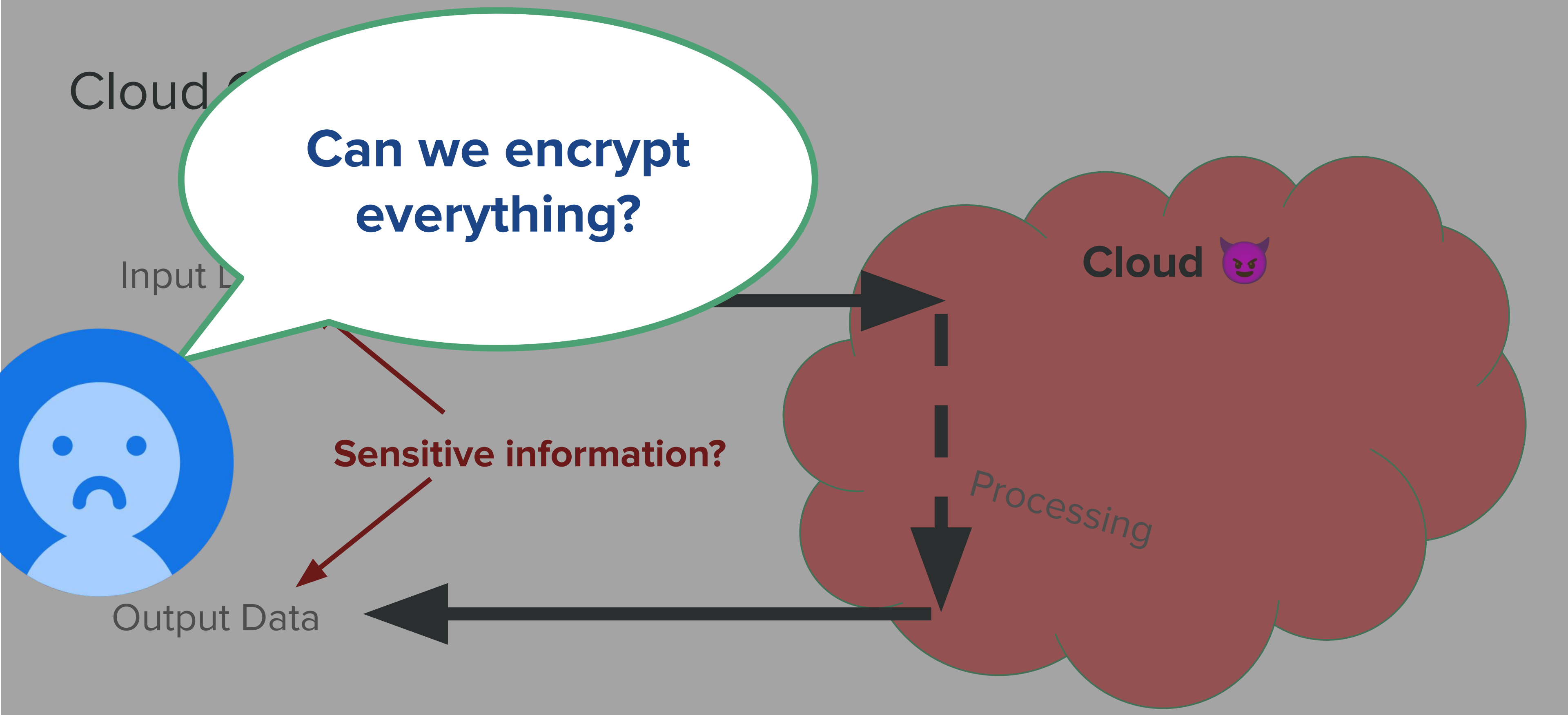
# Context

# Cloud Computing

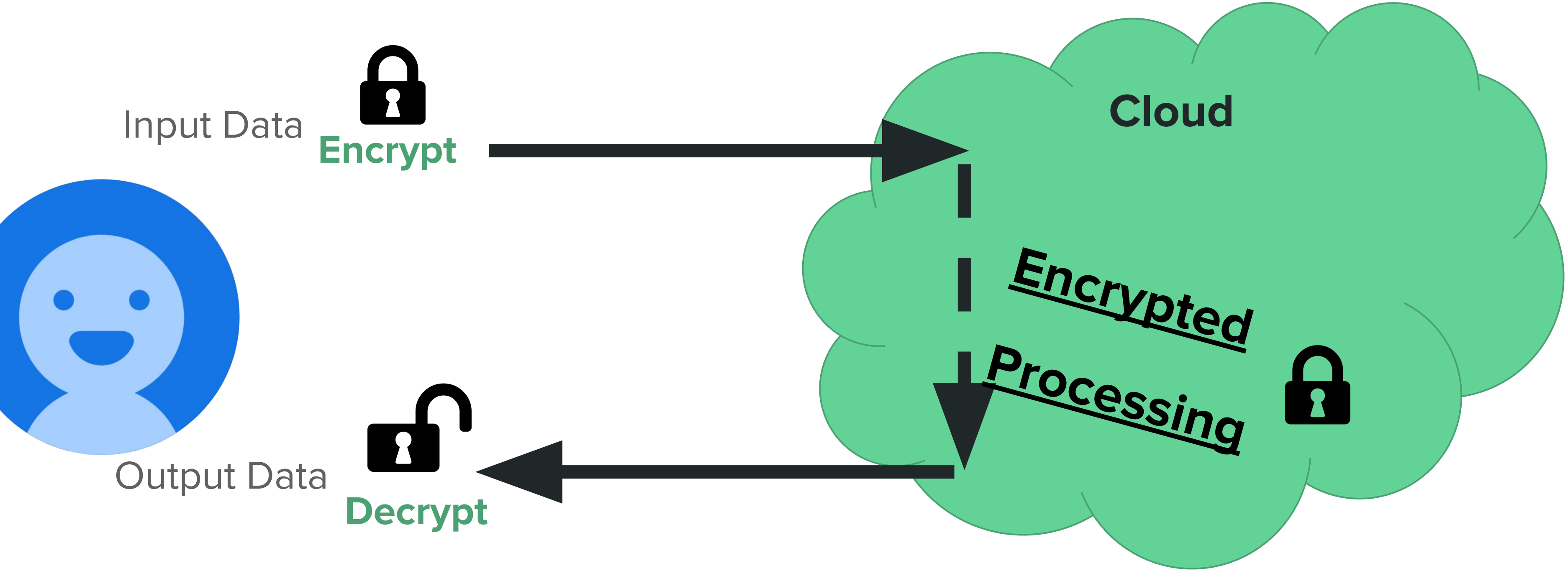


# Cloud Computing



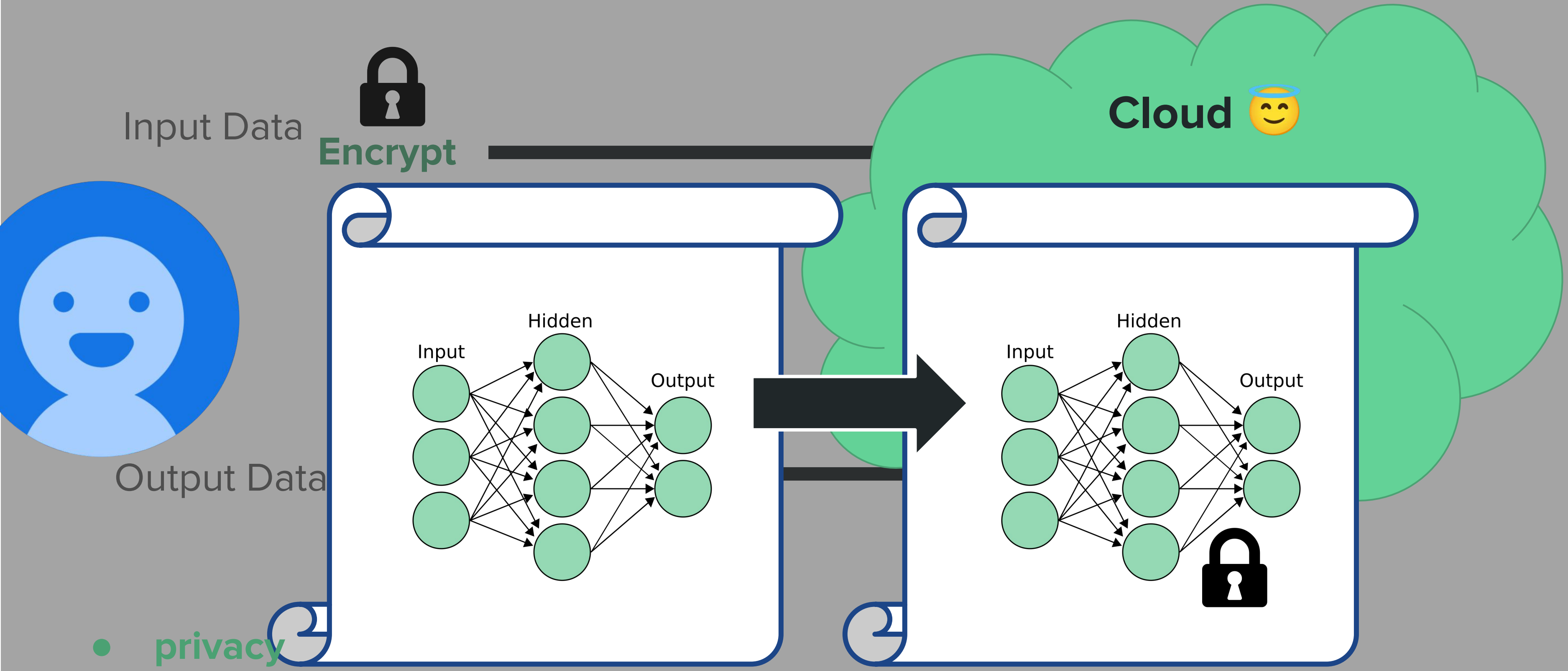

Cloud
Can we encrypt everything?
Input
Cloud 😈
Sensitive information?
Processing
Output Data

# Homomorphic Encryption (HE)




- privacy

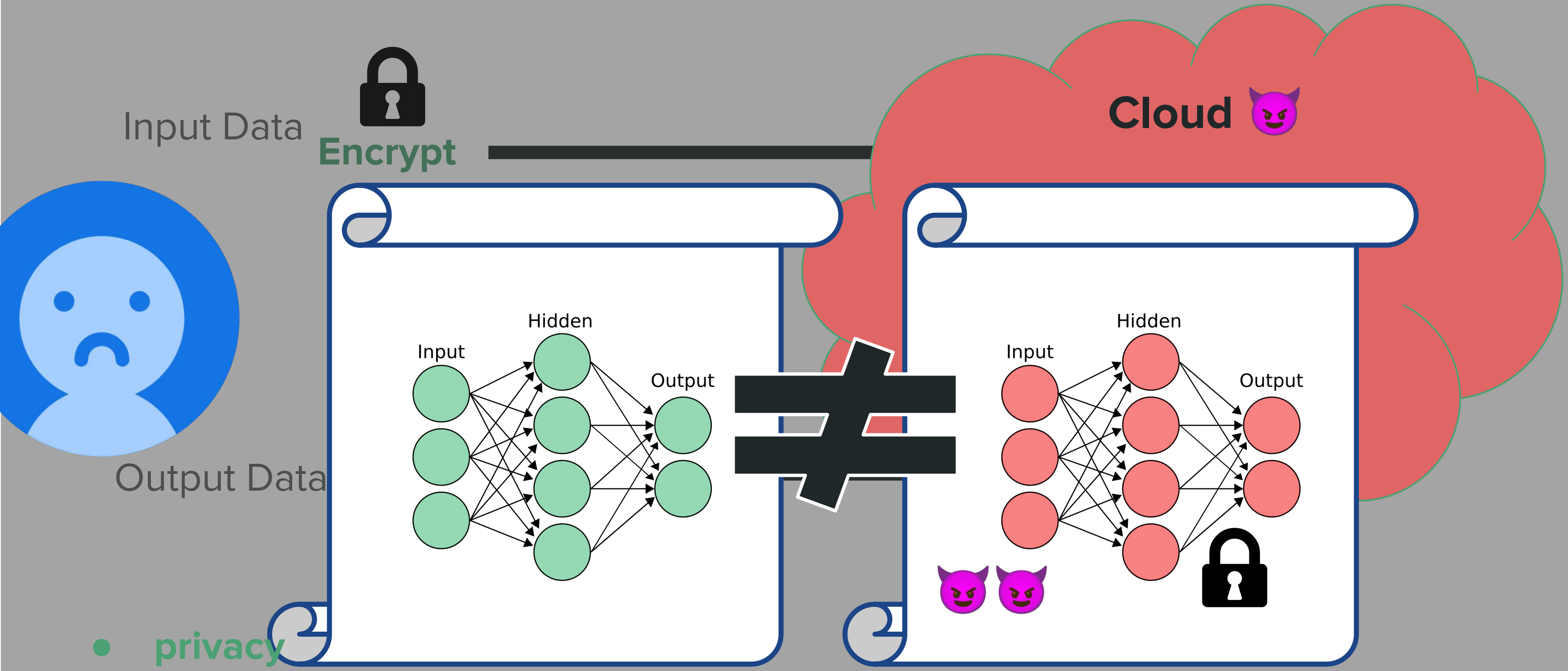
# Homomorphic Encryption (HE)




- privacy

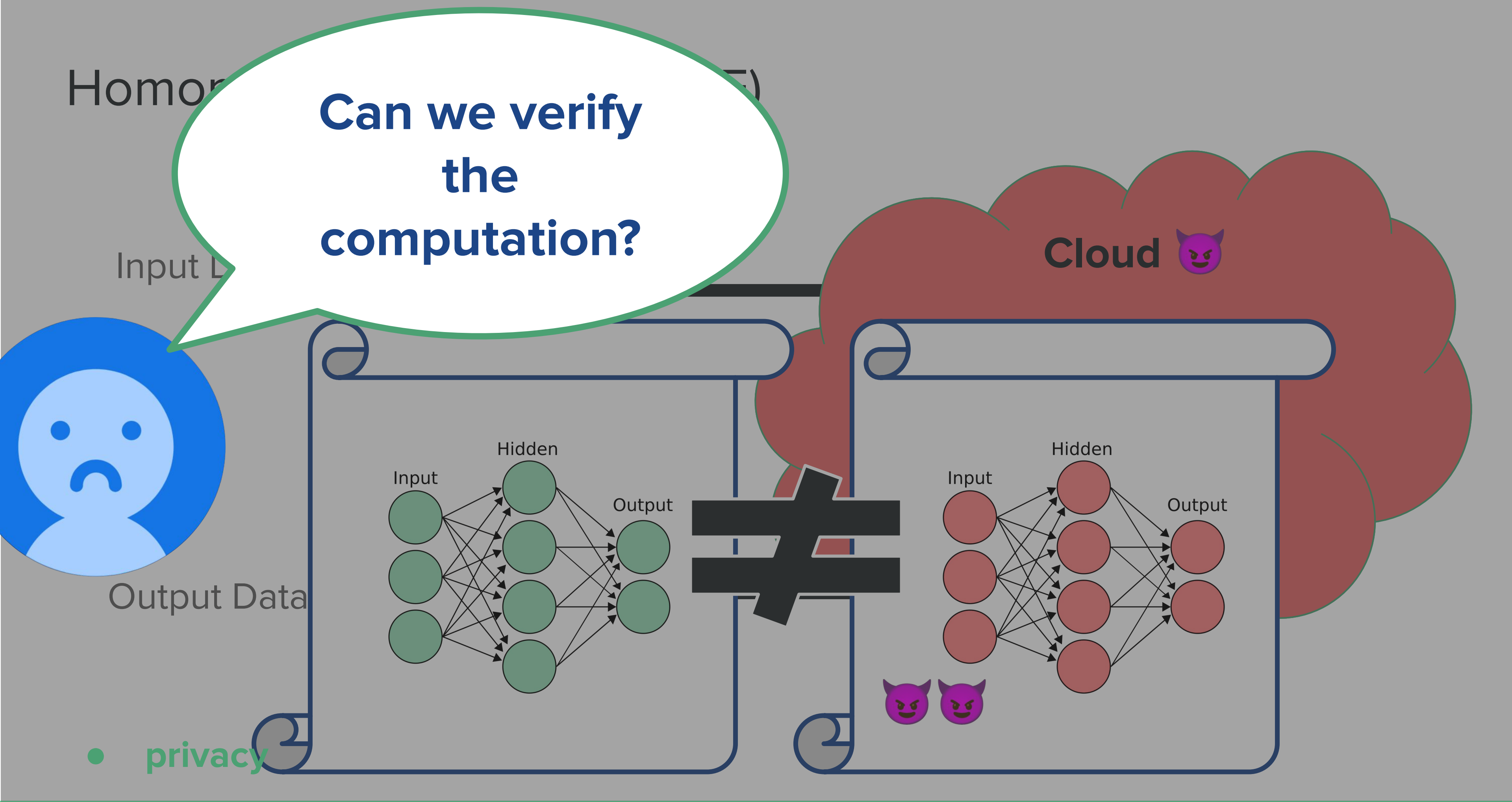
# Homomorphic Encryption (HE)

Input Data

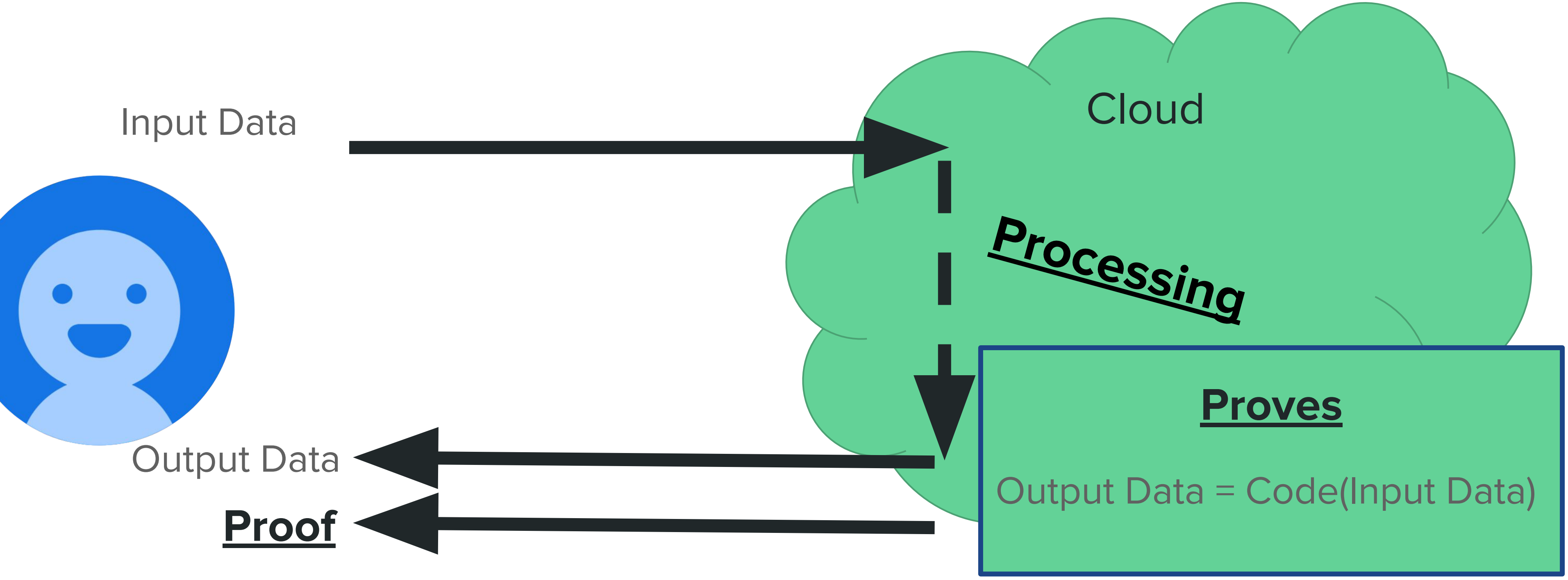



Output Data

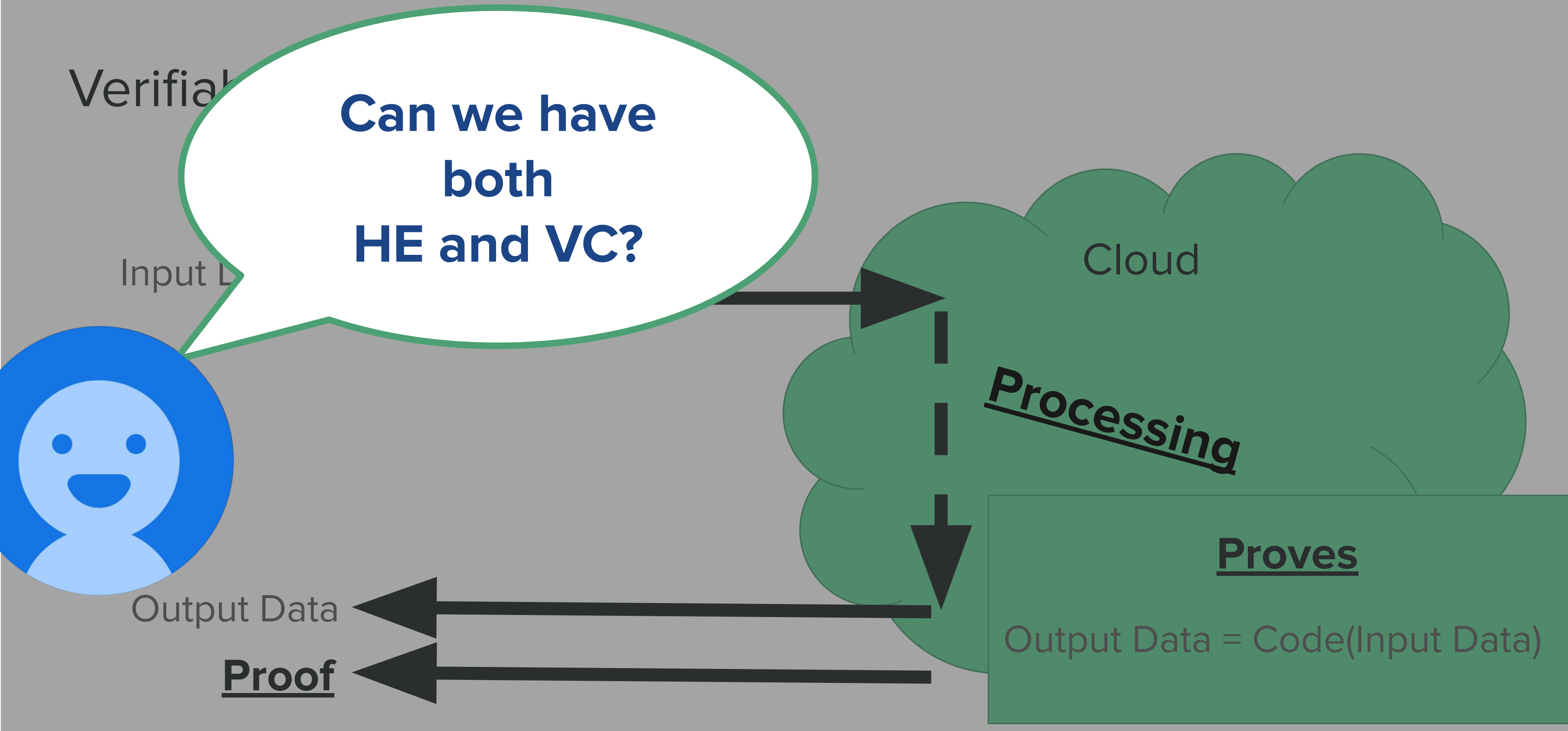
- privacy


Homor
E)
Can we verify the computation?
Input L
Cloud
Hidden
Input
Output
Hidden
Input
Output
Output Data
privacy

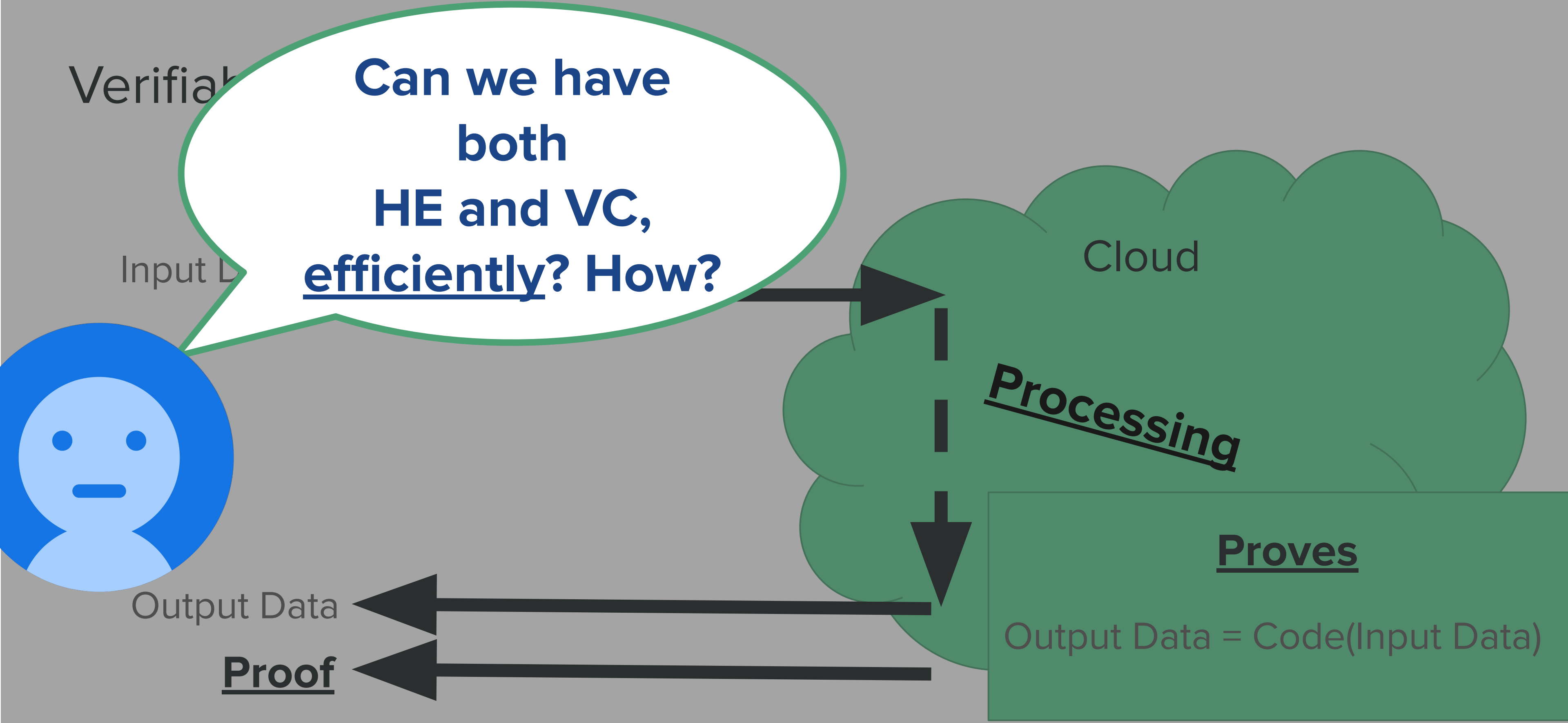
# Verifiable computation (VC)




- **Privacy**
- **Integrity**
- **Succinctness**




- **Privacy**
- **Integrity**
- **Succinctness**




- Privacy
- Integrity
- Succinctness

# VC-HE

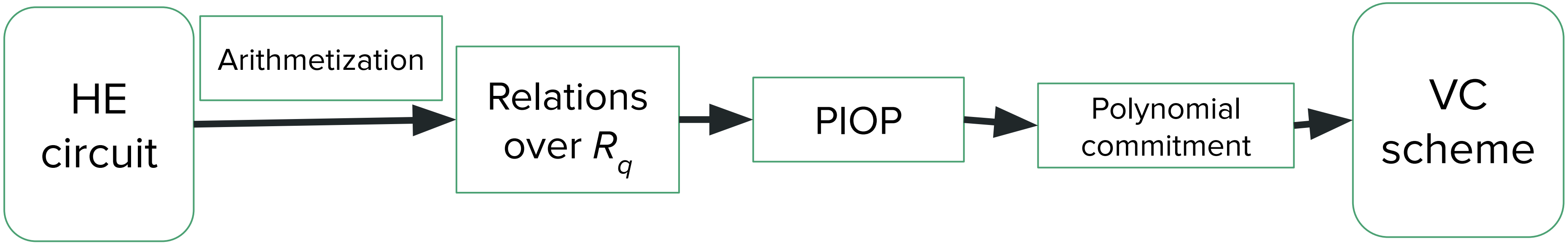
# VC-HE

| | Public Verification | Native $R_q$ Arithmetic | Efficient Key Switching / Rescale | Efficient Bootstrapping | CKKS (approximate schemes) |
|---|---|---|---|---|---|
| Generic SNARK[1] | ✓ | ❌ | ❌ | ❌ | ✓ |
| Rinocchio[2] | ❌ | ✓ | ❌ | ❌ | ✓ |
| HE-IOPs[3] | ❌ | ✓ | ✓ | ✓ | ❌ |
| **Our Work** | ✓ | ✓ | ✓ | ? | ✓ |


[1] A. Viand, C. Knabenhans, and A. Hithnawi, "*Verifiable Fully Homomorphic Encryption*" arXiv:2301.07041
[2] C. Ganesh, A. Nitulescu, and E. Soria-Vazquez, "*Rinocchio: SNARKs for Ring Arithmetic*" Journal of Cryptology, 2023
[3] D. F. Aranha, A. Costache, A. Guimarães, and E. Soria-Vazquez, "*HELIOPOLIS: Verifiable Computation over Homomorphically Encrypted Data from Interactive Oracle Proofs is Practical*" ASIACRYPT 2024

# Our contributions

- VC-HE for **CKKS**
- **Modular** solution




- **Proof-friendly CKKS**
- **AC satisfiability**
- **Range check**
- **Custom GKR**
- **Range proofs for polynomial rings**
- **Improved Brakedown**

# Proof-friendly CKKS

The polynomial ring $R_q = \mathbb{Z}_q[X]/(X^N + 1)$

$$q \approx 2^{300} \qquad N \approx 2^{14}$$

The polynomial ring $R_q = \mathbb{Z}_q[X]/(X^N + 1)$

$$R_q$$

# The polynomial ring $R_q = \mathbb{Z}_q[X]/(X^N + 1)$

$$R_q$$

- Efficient HE computations
    - RNS

# The polynomial ring $R_q = \mathbb{Z}_q[X]/(X^N + 1)$

$$q = \prod_{i=1}^{L} p_i$$

$$R_q \cong \begin{array}{|c|} \hline R_{p_1} \\ \hline R_{p_2} \\ \hline R_{p_3} \\ \hline \end{array}$$

- Efficient HE computations
  - RNS

# The polynomial ring $R_q = \mathbb{Z}_q[X]/(X^N + 1)$

$$R_q \overset{q = \prod_{i=1}^{L} p_i}{\cong} \begin{array}{|c|}\hline R_{p_1} \\ \hline R_{p_2} \\ \hline R_{p_3} \\ \hline \end{array}$$

- Efficient HE computations
  - RNS
- Soundness
  - Large exceptional set

# The polynomial ring $R_q = \mathbb{Z}_q[X]/(X^N+1)$

$$R_q \underset{q=\prod_{i=1}^{L} p_i}{\cong} \begin{array}{c} R_{p_1} \\ R_{p_2} \\ R_{p_3} \end{array}$$

$$R_{p_1} \underset{X^N+1=\prod_{i=1}^{k}(X^d-\zeta^{2i-1}) \mod p_1}{\cong} R_{11} \quad R_{12} \quad R_{13} \quad R_{14}$$

$$R_{p_2} \underset{X^N+1=\prod_{i=1}^{k}(X^d-\zeta^{2i-1}) \mod p_2}{\cong} R_{21} \quad R_{22} \quad R_{23} \quad R_{24}$$

$$R_{p_3} \underset{X^N+1=\prod_{i=1}^{k}(X^d-\zeta^{2i-1}) \mod p_3}{\cong} R_{31} \quad R_{32} \quad R_{33} \quad R_{34}$$

- Efficient HE computations
  - RNS
- Soundness
  - Large exceptional set

# The polynomial ring $R_q = \mathbb{Z}_q[X]/(X^N + 1)$

$$R_q \cong \quad q = \prod_{i=1}^{L} p_i$$

$$R_{p_1} \quad X^N + 1 = \prod_{i=1}^{k}(X^d - \zeta^{2i-1}) \mod p_1 \quad \cong \quad \mathbb{F}_{p_1^d} \; \mathbb{F}_{p_1^d} \; \mathbb{F}_{p_1^d} \; \mathbb{F}_{p_1^d}$$

$$R_{p_2} \quad X^N + 1 = \prod_{i=1}^{k}(X^d - \zeta^{2i-1}) \mod p_2 \quad \cong \quad \mathbb{F}_{p_2^d} \; \mathbb{F}_{p_2^d} \; \mathbb{F}_{p_2^d} \; \mathbb{F}_{p_2^d}$$

$$R_{p_3} \quad X^N + 1 = \prod_{i=1}^{k}(X^d - \zeta^{2i-1}) \mod p_3 \quad \cong \quad \mathbb{F}_{p_3^d} \; \mathbb{F}_{p_3^d} \; \mathbb{F}_{p_3^d} \; \mathbb{F}_{p_3^d}$$

- Efficient HE computations
  - RNS
- Soundness
  - Large exceptional set

# The polynomial ring $R_q = \mathbb{Z}_q[X]/(X^N + 1)$

$R_q$

$q = \prod_{i=1}^{L} p_i$

$\cong$

$R_{p_1}$

$R_{p_2}$

$R_{p_3}$

$X^N + 1 = \prod_{i=1}^{k} (X^d - \zeta^{2i-1}) \mod p_1$

$\cong$

$\mathbb{F}_{p_1^d}$ $\mathbb{F}_{p_1^d}$ $\mathbb{F}_{p_1^d}$ $\mathbb{F}_{p_1^d}$

$X^N$

$\mathbb{F}_{p_2^d}$

$X^N$

$\mathbb{F}_{p_3^d}$

**Size of our exceptional set!**

- Efficient HE computations
  - RNS
- Soundness
  - Large exceptional set

# The polynomial ring $R_q = \mathbb{Z}_q[X]/(X^N + 1)$

$$R_q \; \underset{\cong}{q = \prod_{i=1}^{L} p_i} \; \begin{array}{c} R_{p_1} \\ R_{p_2} \\ R_{p_3} \end{array}$$

$X^N + 1 = \prod_{i=1}^{k} (X^d - \zeta^{2i-1}) \mod p_1$

$\cong$

$\mathbb{F}_{p_1^d}$ $\mathbb{F}_{p_1^d}$ $\mathbb{F}_{p_1^d}$ $\mathbb{F}_{p_1^d}$

$X^N$ ... $\mathbb{F}_{p_2^d}$

$X$ ... $\mathbb{F}_{p_3^d}$

**Size of our exceptional set!**

To get **128** bits of security:
$|p_i| = 32$, $d = 4$
or
$|p_i| = 64$, $d = 2$

- Efficient HE computations
  - RNS
- Soundness
  - Large exceptional set

# The polynomial ring $R_q = \mathbb{Z}_q[X]/(X^N + 1)$

$$R_q \overset{q = \prod_{i=1}^{L} p_i}{\cong} R_{p_1} \times R_{p_2} \times R_{p_3}$$

$$X^N + 1 = \prod_{i=1}^{k} (X^d - \zeta^{2i-1}) \mod p_1$$

$$\cong \quad \mathbb{F}_{p_1^d} \quad \mathbb{F}_{p_1^d} \quad \mathbb{F}_{p_1^d} \quad \mathbb{F}_{p_1^d}$$

$X^N$ … $\mathbb{F}_{p_2^d}$

$X^N$ … $\mathbb{F}_{p_3^d}$

**Size of our exceptional set!**

To get **128** bits of security:
$|p_i| = 32$, $d = 4$
or
$|p_i| = 64$, $d = 2$

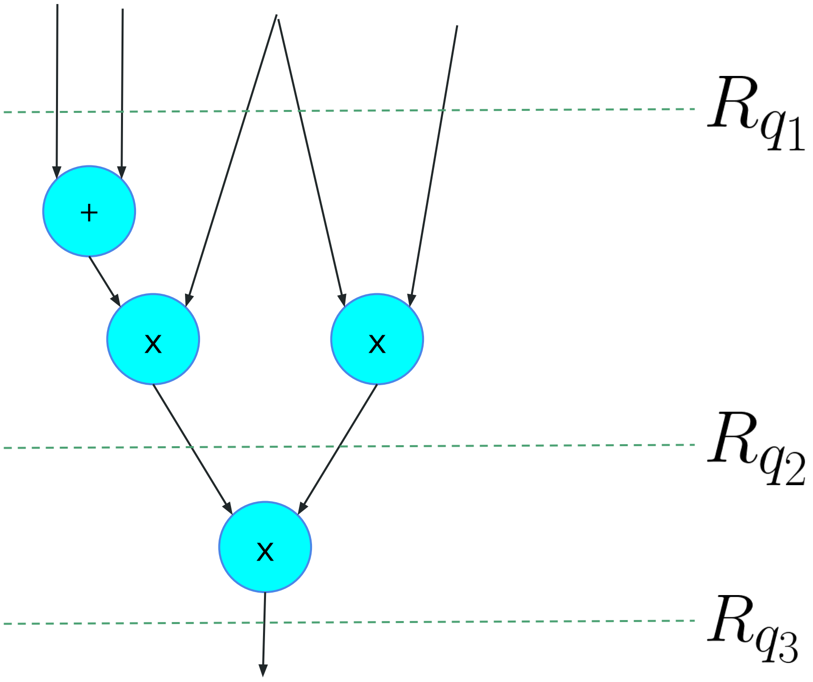
**Optimal performance/security**:
$|p_i| = 49$, $d = 4$

- Efficient HE computations
  - RNS
- Soundness
  - Large exceptional set

# CKKS

$$\begin{cases} q_1 = q \\ q_i = q_{i-1}/p_i \end{cases}$$

$$q = q_1 > q_2 > \cdots > q_L$$



# CKKS

- An approximate scheme:


- RLWE ciphertext:

$$(a, b) \in R_q^2$$

- RNS representation with 3 components:

| | | | | | | |
|---|---|---|---|---|---|---|
| $a_{01}$ | $a_{02}$ | $a_{03}$ | | $b_{01}$ | $b_{02}$ | $b_{03}$ |

# CKKS

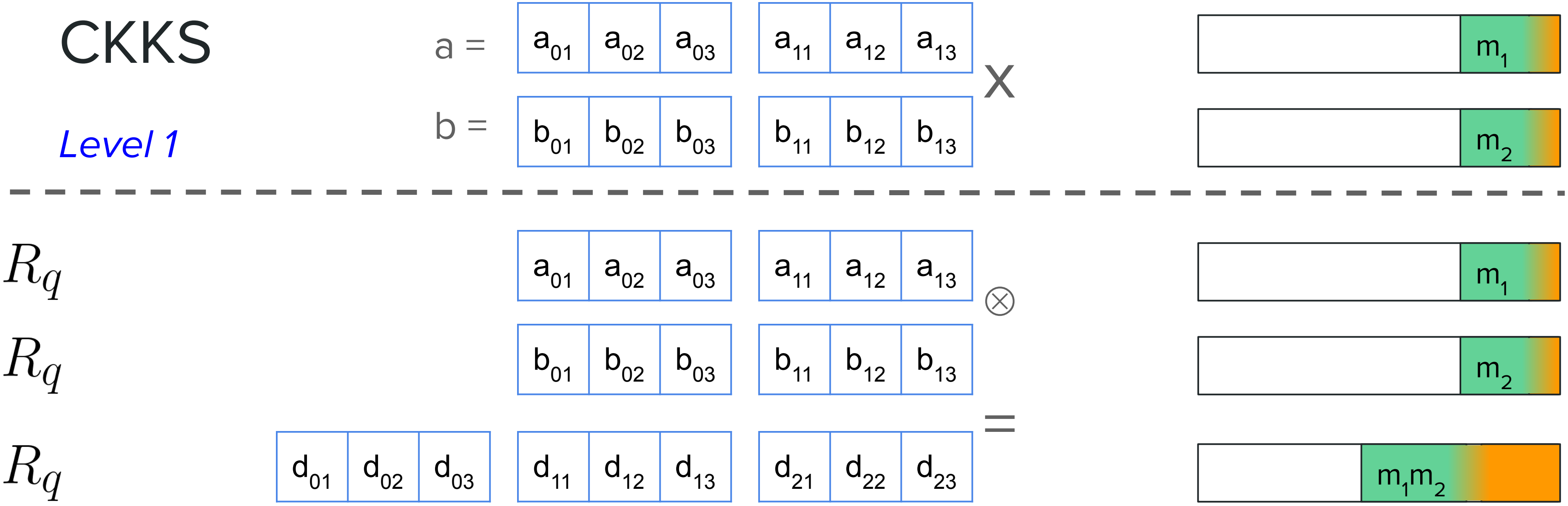
*Level 1*

$$a = \begin{array}{|c|c|c|}\hline a_{01} & a_{02} & a_{03} \\ \hline\end{array} \quad \begin{array}{|c|c|c|}\hline a_{11} & a_{12} & a_{13} \\ \hline\end{array}$$

$\times$

$$b = \begin{array}{|c|c|c|}\hline b_{01} & b_{02} & b_{03} \\ \hline\end{array} \quad \begin{array}{|c|c|c|}\hline b_{11} & b_{12} & b_{13} \\ \hline\end{array}$$

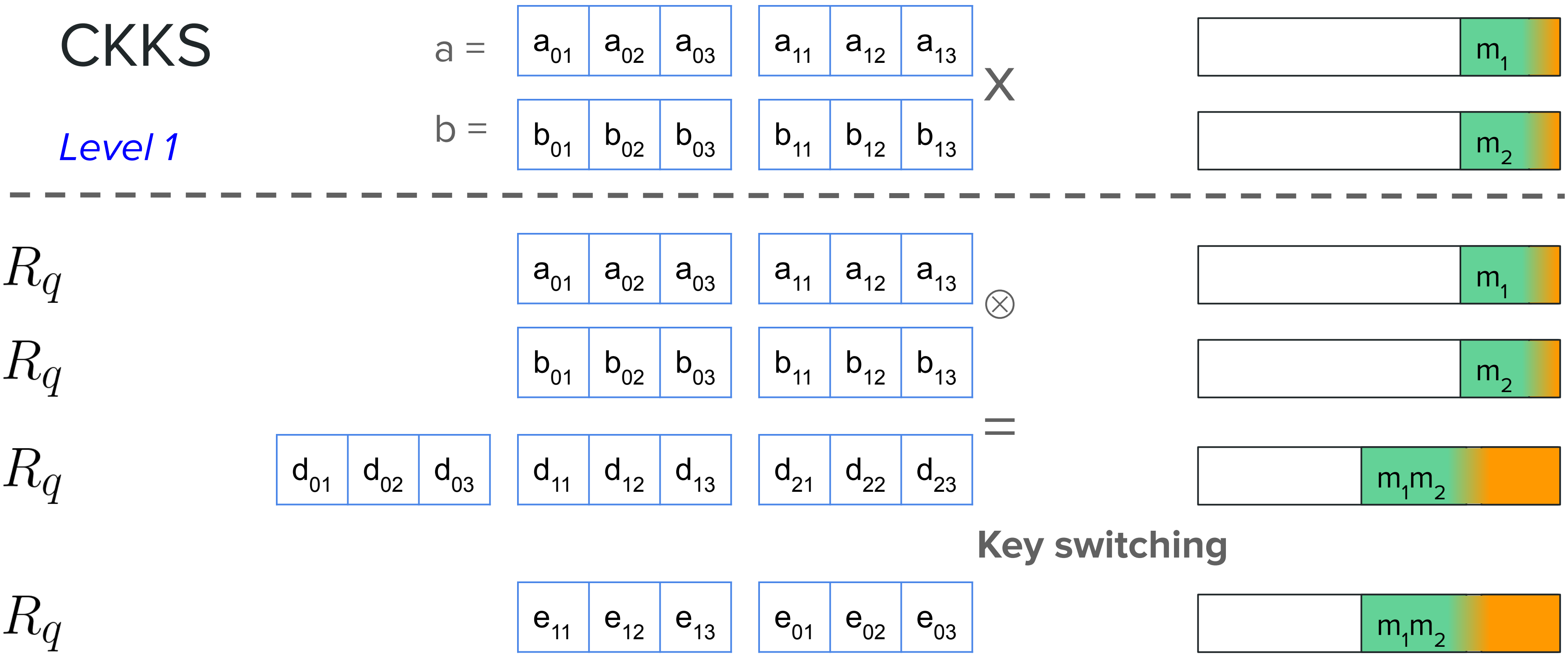
# CKKS



Level 1
a =
$a_{01}$ $a_{02}$ $a_{03}$ $a_{11}$ $a_{12}$ $a_{13}$
X
b =
$b_{01}$ $b_{02}$ $b_{03}$ $b_{11}$ $b_{12}$ $b_{13}$
$m_1$
$m_2$
$R_q$
$a_{01}$ $a_{02}$ $a_{03}$ $a_{11}$ $a_{12}$ $a_{13}$
$\otimes$
$R_q$
$b_{01}$ $b_{02}$ $b_{03}$ $b_{11}$ $b_{12}$ $b_{13}$
=
$R_q$
$d_{01}$ $d_{02}$ $d_{03}$ $d_{11}$ $d_{12}$ $d_{13}$ $d_{21}$ $d_{22}$ $d_{23}$
$m_1$
$m_2$
$m_1m_2$

# CKKS

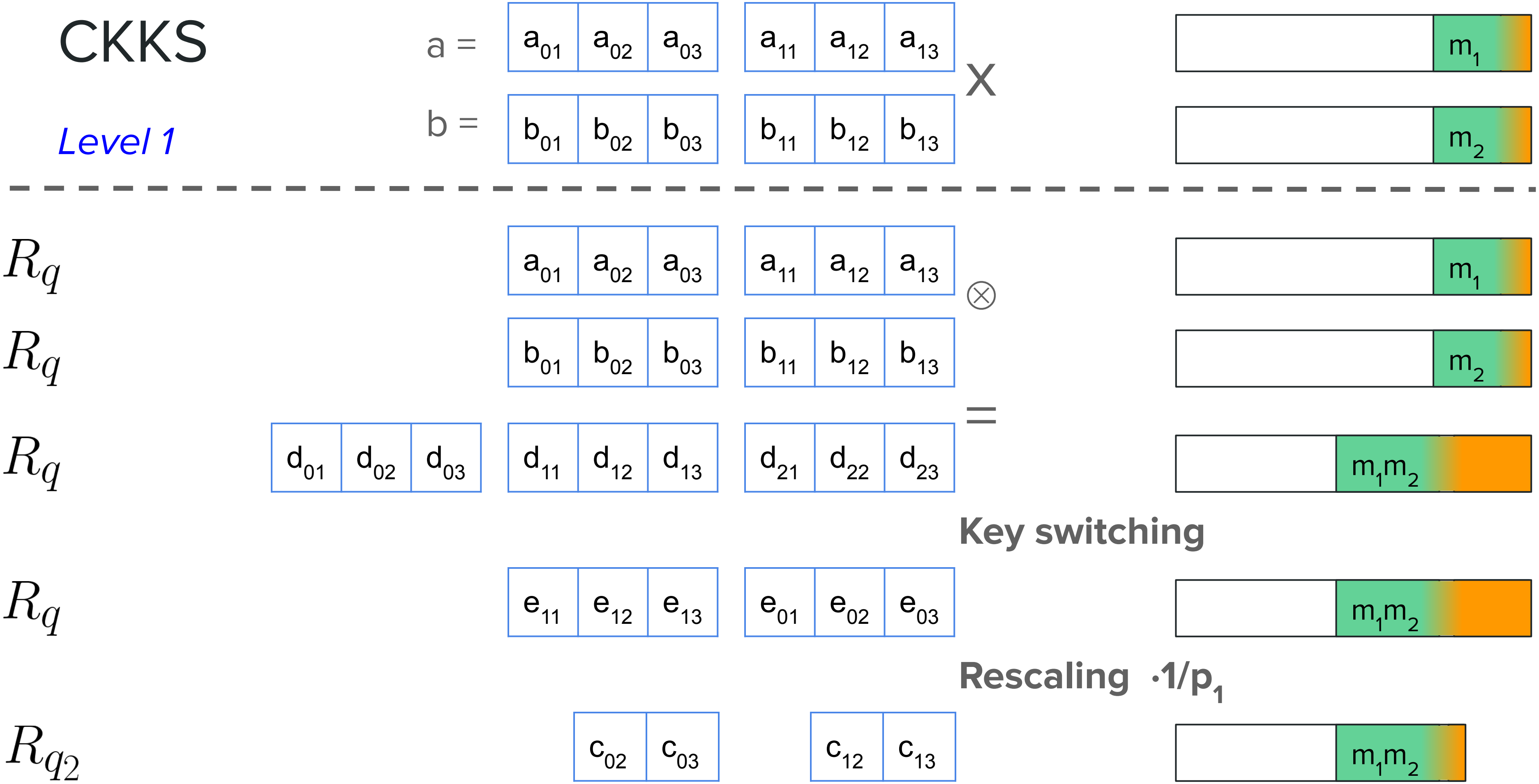
*Level 1*



a =
$a_{01}$ $a_{02}$ $a_{03}$ $a_{11}$ $a_{12}$ $a_{13}$
X
b =
$b_{01}$ $b_{02}$ $b_{03}$ $b_{11}$ $b_{12}$ $b_{13}$
$m_1$
$m_2$
$R_q$
$a_{01}$ $a_{02}$ $a_{03}$ $a_{11}$ $a_{12}$ $a_{13}$
$\otimes$
$m_1$
$R_q$
$b_{01}$ $b_{02}$ $b_{03}$ $b_{11}$ $b_{12}$ $b_{13}$
$m_2$
=
$R_q$
$d_{01}$ $d_{02}$ $d_{03}$ $d_{11}$ $d_{12}$ $d_{13}$ $d_{21}$ $d_{22}$ $d_{23}$
$m_1m_2$
Key switching
$R_q$
$e_{11}$ $e_{12}$ $e_{13}$ $e_{01}$ $e_{02}$ $e_{03}$
$m_1m_2$

# CKKS

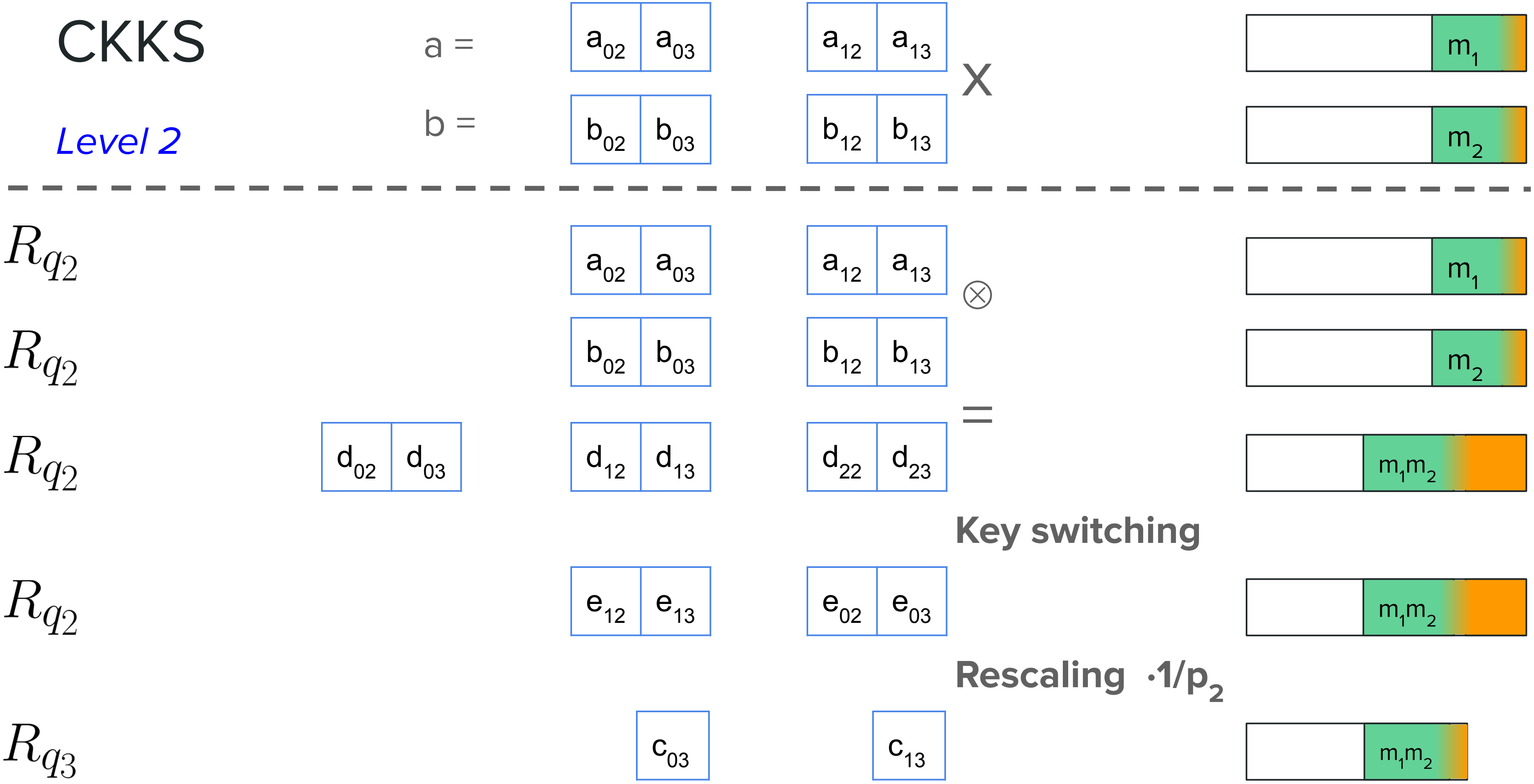
*Level 1*



a =
a01 a02 a03
a11 a12 a13
X
b =
b01 b02 b03
b11 b12 b13
m1
m2
Rq
a01 a02 a03
a11 a12 a13
⊗
m1
Rq
b01 b02 b03
b11 b12 b13
m2
=
Rq
d01 d02 d03
d11 d12 d13
d21 d22 d23
m1m2
Key switching
Rq
e11 e12 e13
e01 e02 e03
m1m2
Rescaling ·1/p1
Rq2
c02 c03
c12 c13
m1m2

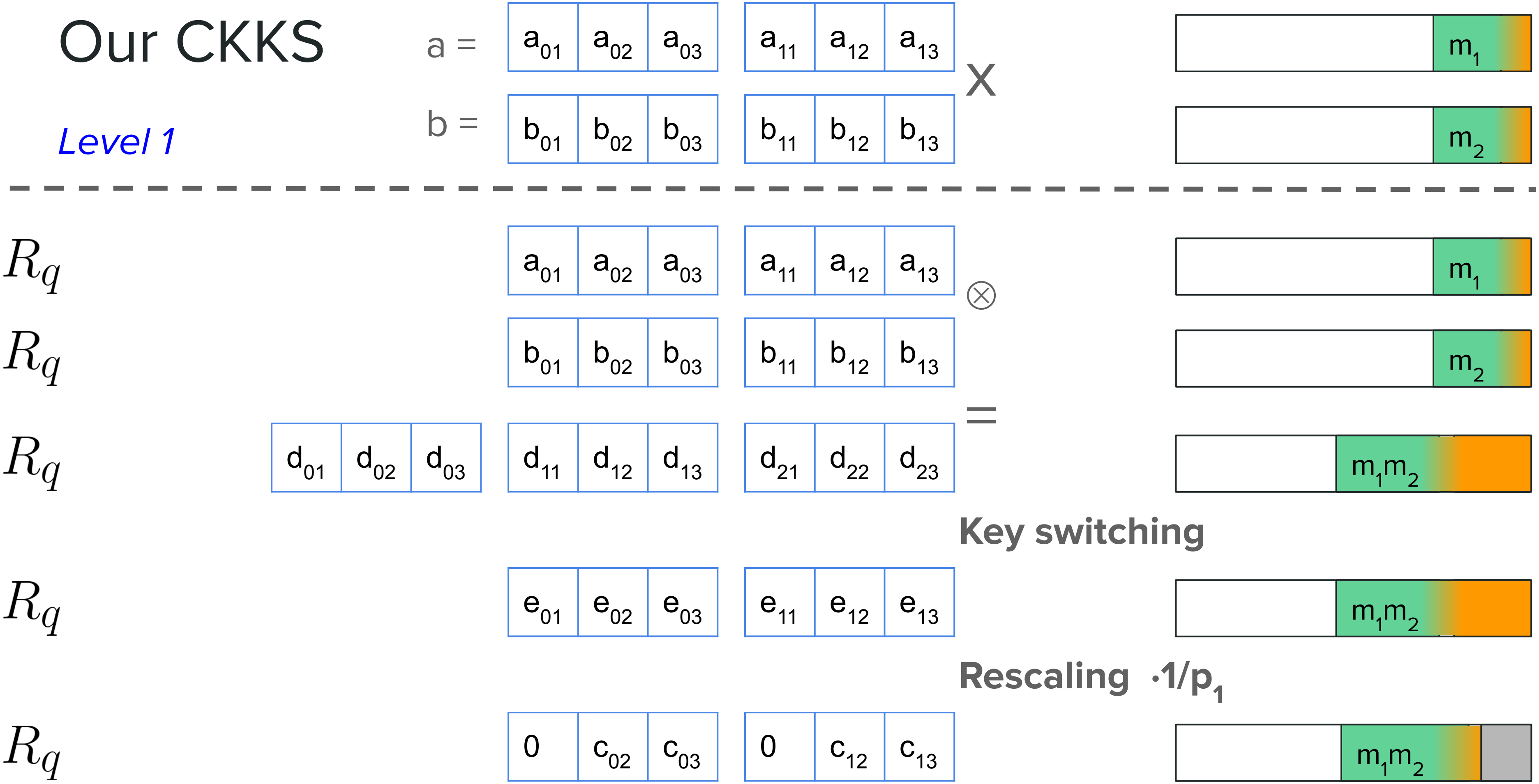
# CKKS

*Level 2*


a =
b =
a02 a03
a12 a13
b02 b03
b12 b13
X
m1
m2
Rq2
Rq2
Rq2
Rq2
Rq3
⊗
=
d02 d03
d12 d13
d22 d23
m1m2
Key switching
e12 e13
e02 e03
Rescaling ·1/p2
c03
c13

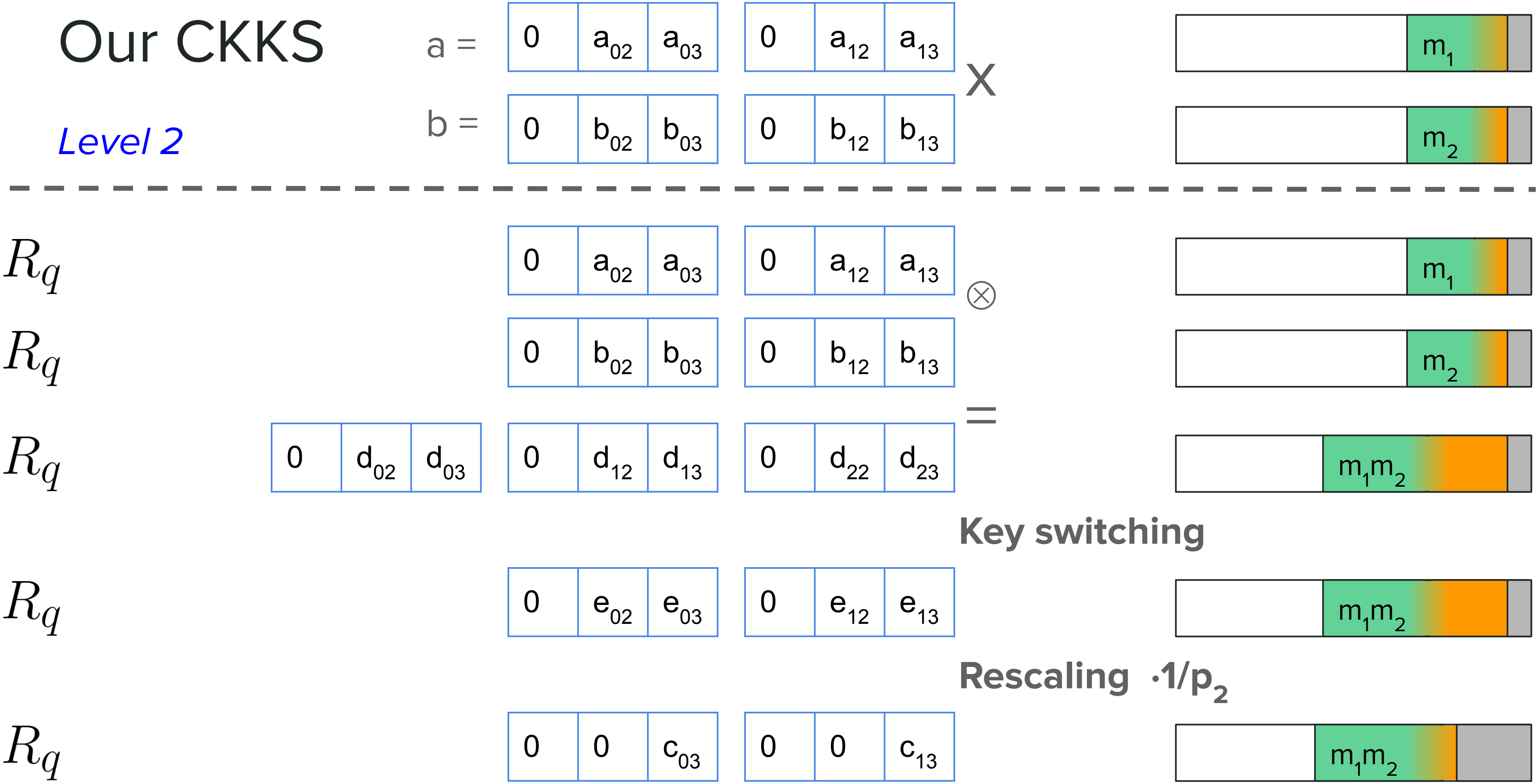
# Our CKKS

*Level 1*



# Our CKKS

*Level 2*



# Proof-friendly CKKS vs CKKS

| | Proof-friendly CKKS | | CKKS |
|---|---|---|---|
| | d = 2 | d = 4 | HEXL |
| CKKS multiplication | 7.394ms | 8.457ms | 7.197ms |

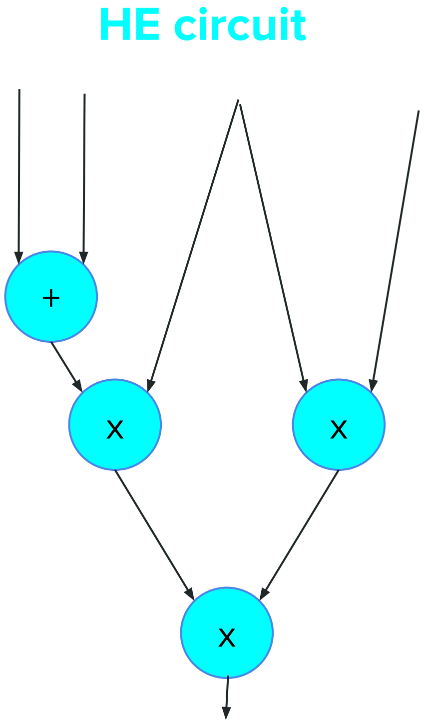
N = 16384
#RNS components (L) = 6

# Proof-friendly CKKS in summary

- Carefully chosen ring setup
  - High **soundness** for proof system
  - **Efficiency** of computations
- Ring does not change
  - Proof system works on **same ring**
- Noise analysis
  - Easier to **prove bounds** on ciphertexts (see later)

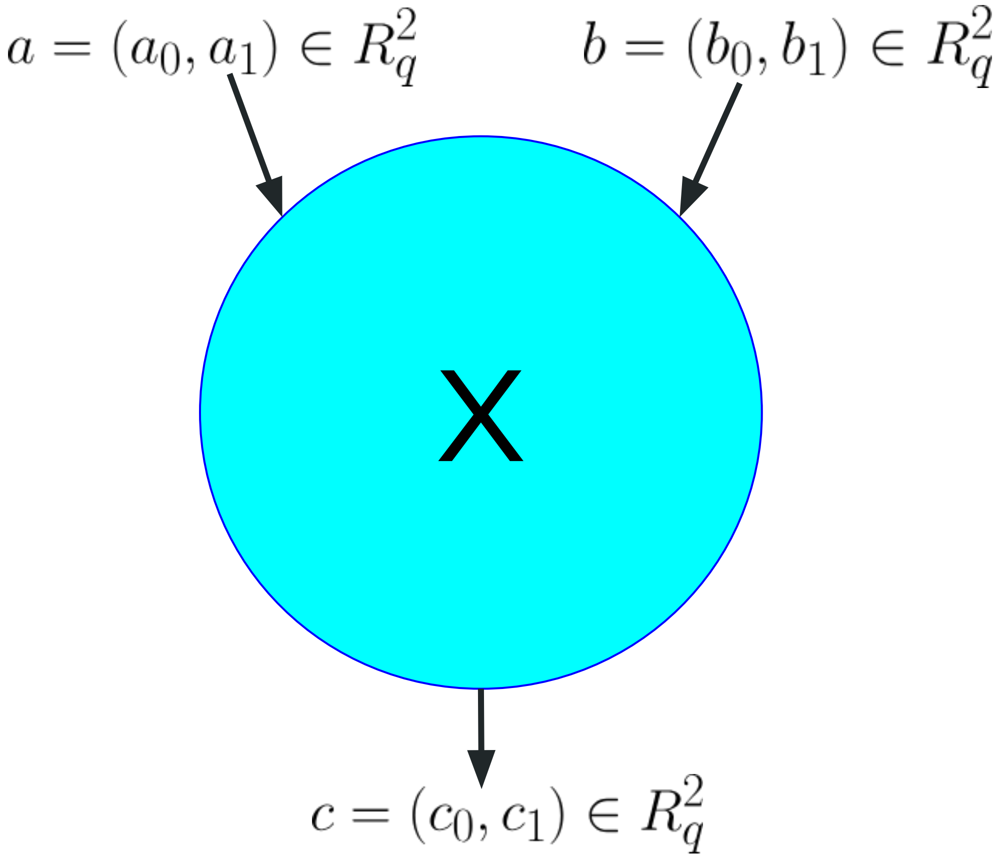
# Arithmetization

# Arithmetization

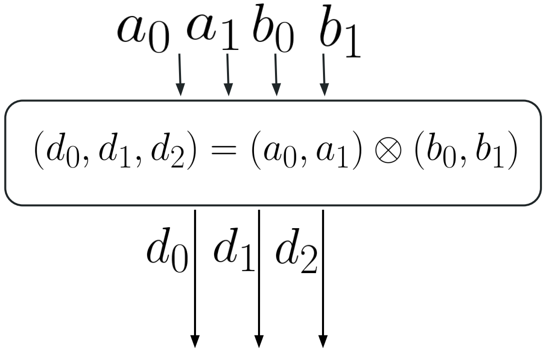



$$\left\{ \begin{array}{c} \text{Relations} \\ \text{over} \\ R_q \end{array} \right\}$$

# Arithmetization



# Arithmetization




**Polynomial mult**

# Arithmetization

$a_0 \quad a_1 \quad b_0 \quad b_1$

$$(d_0, d_1, d_2) = (a_0, a_1) \otimes (b_0, b_1)$$

**Polynomial mult**

$d_0 \quad d_1 \quad d_2$

$$e_0 = d_0 + \langle \text{evk}, CRT^{-1}(d_2) \rangle$$
$$e_1 = d_1 + \langle \text{evk}, CRT^{-1}(d_2) \rangle$$

**Key-switching**

$e_0 \quad e_1$

# Arithmetization

$a_0 \; a_1 \; b_0 \; b_1$

$$(d_0, d_1, d_2) = (a_0, a_1) \otimes (b_0, b_1)$$

**Polynomial mult**

$d_0 \; d_1 \; d_2$

$$e_0 = d_0 + \langle \text{evk}, CRT^{-1}(d_2) \rangle$$
$$e_1 = d_1 + \langle \text{evk}, CRT^{-1}(d_2) \rangle$$

**Key-switching**

$e_0 \quad e_1$

$$c_0 = (e_0 - [e_0]_{p_l}) \cdot p_l^{-1}$$
$$c_1 = (e_1 - [e_1]_{p_l}) \cdot p_l^{-1}$$

**Rescaling**

$c_1 \quad c_0$

# Arithmetization

$a_0 \; a_1 \; b_0 \; b_1$

$$(d_0, d_1, d_2) = (a_0, a_1) \otimes (b_0, b_1)$$

$d_0 \; d_1 \; d_2$

$$e_0 = d_0 + \langle \mathrm{evk}, CRT^{-1}(d_2) \rangle$$
$$e_1 = d_1 + \langle \mathrm{evk}, CRT^{-1}(d_2) \rangle$$

$e_0 \; e_1$

$$c_0 = (e_0 - [e_0]_{p_l}) \cdot p_l^{-1}$$
$$c_1 = (e_1 - [e_1]_{p_l}) \cdot p_l^{-1}$$

$c_1 \; c_0$

# Arithmetization

**Algebraic operation**

$a_0 \; a_1 \; b_0 \; b_1$

$$(d_0, d_1, d_2) = (a_0, a_1) \otimes (b_0, b_1)$$

$d_0 \; d_1 \; d_2$

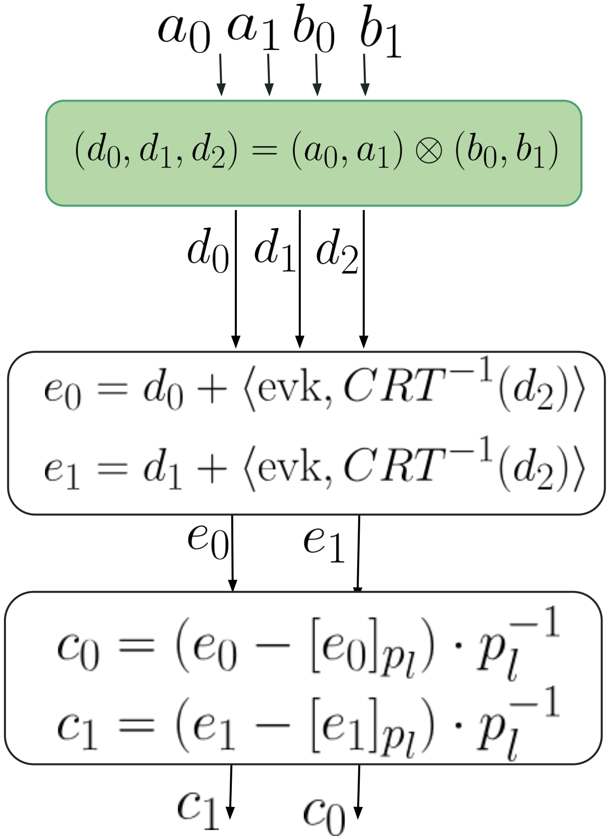
$$e_0 = d_0 + \langle \text{evk}, CRT^{-1}(d_2) \rangle$$
$$e_1 = d_1 + \langle \text{evk}, CRT^{-1}(d_2) \rangle$$

$e_0 \; e_1$

$$c_0 = (e_0 - [e_0]_{p_l}) \cdot p_l^{-1}$$
$$c_1 = (e_1 - [e_1]_{p_l}) \cdot p_l^{-1}$$

$c_1 \; c_0$

# Arithmetization



# Arithmetization

$a_0 \; a_1 \; b_0 \; b_1$

$$(d_0, d_1, d_2) = (a_0, a_1) \otimes (b_0, b_1)$$

$d_0 \; d_1 \; d_2$

$$e_0 = d_0 + \langle \text{evk}, \boxed{CRT^{-1}(d_2)} \rangle$$
$$e_1 = d_1 + \langle \text{evk}, \boxed{CRT^{-1}(d_2)} \rangle$$

**Not algebraic!**

$e_0 \quad e_1$

$$c_0 = (e_0 - [e_0]_{p_l}) \cdot p_l^{-1}$$
$$c_1 = (e_1 - [e_1]_{p_l}) \cdot p_l^{-1}$$

$c_1 \quad c_0$

# Arithmetization

$a_0 \quad a_1 \quad b_0 \quad b_1$

$$(d_0, d_1, d_2) = (a_0, a_1) \otimes (b_0, b_1)$$

$d_0 \quad d_1 \quad d_2$

$$e_0 = d_0 + \langle \text{evk}, \boxed{CRT^{-1}(d_2)} \rangle$$

$$e_1 = d_1 + \langle \text{evk}, \boxed{CRT^{-1}(d_2)} \rangle$$

$e_0 \quad e_1$

$$c_0 = (e_0 - [e_0]_{p_l}) \cdot p_l^{-1}$$

$$c_1 = (e_1 - [e_1]_{p_l}) \cdot p_l^{-1}$$

$c_1 \quad c_0$

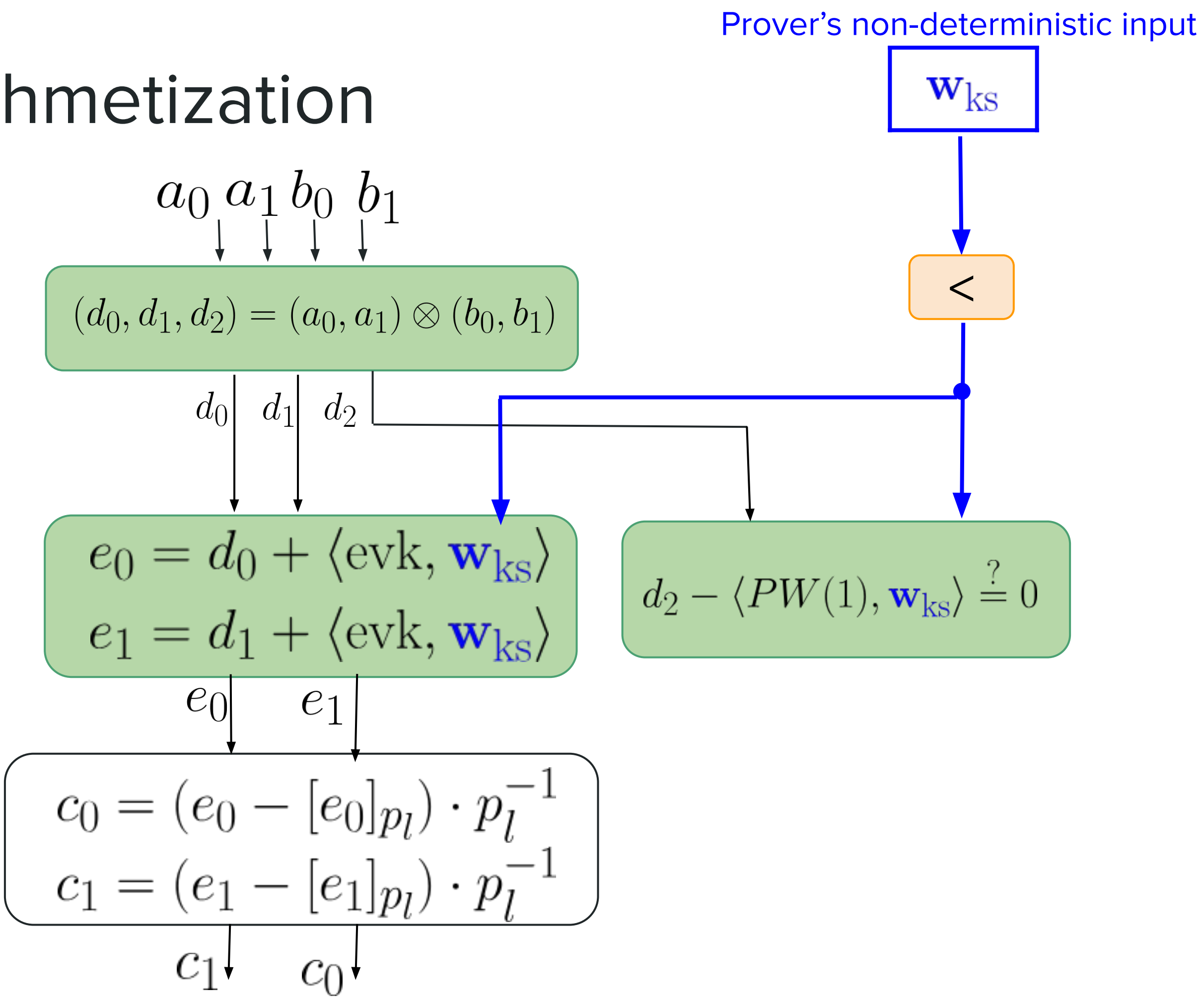
**Not algebraic!**

Prover inputs $\mathbf{w}_{\text{ks}}$

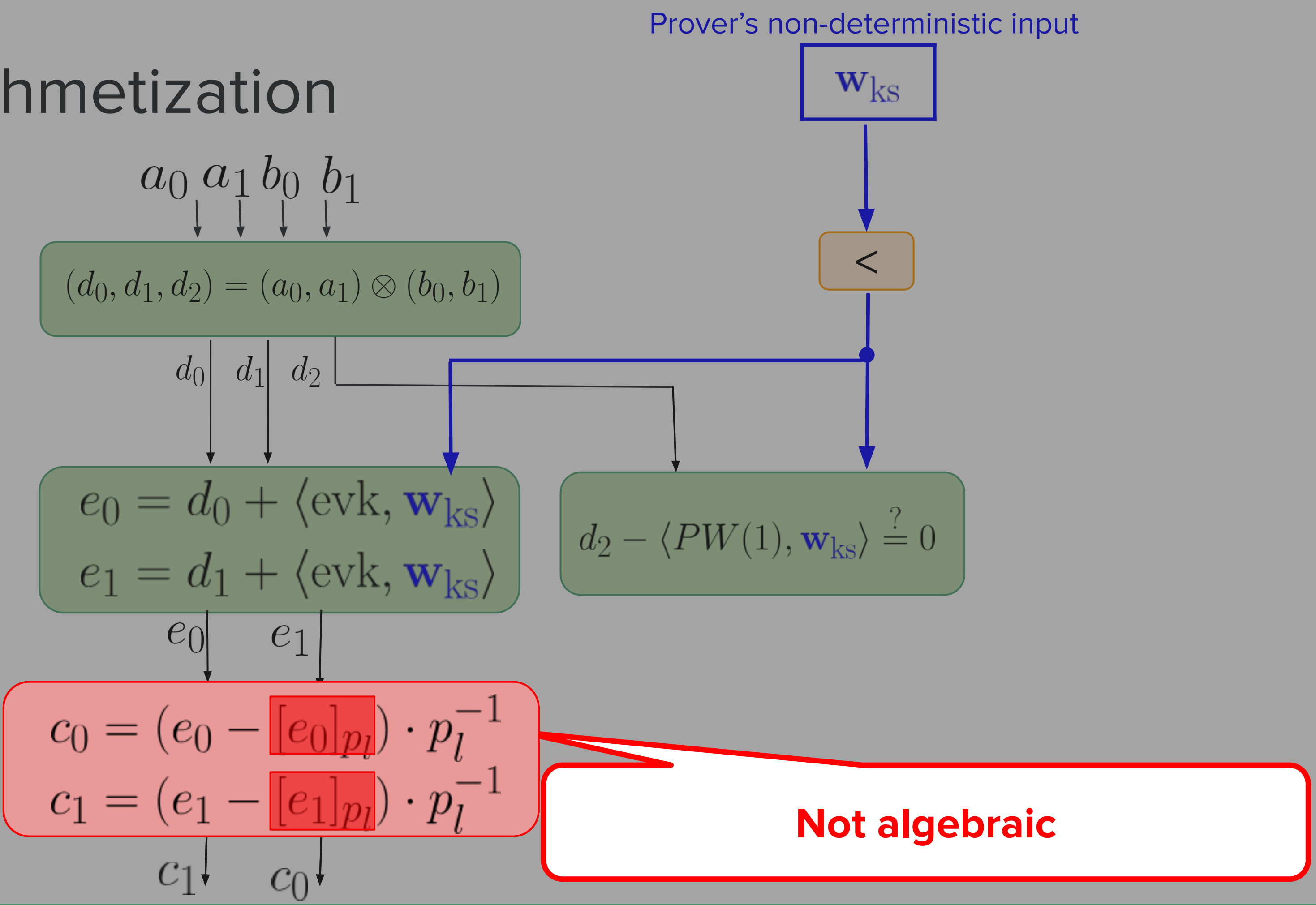
$$\|\mathbf{w}_{\text{ks}}[i]\| < p_i$$

$$d_2 = \langle PW(1), \mathbf{w}_{\text{ks}} \rangle$$

# Arithmetization



# Arithmetization



# Arithmetization

$a_0 \quad a_1 \quad b_0 \quad b_1$

$$(d_0, d_1, d_2) = (a_0, a_1) \otimes (b_0, b_1)$$

$d_0 \quad d_1 \quad d_2$

$$e_0 = d_0 + \langle \mathrm{evk}, \mathbf{w}_{\mathrm{ks}} \rangle$$
$$e_1 = d_1 + \langle \mathrm{evk}, \mathbf{w}_{\mathrm{ks}} \rangle$$

$e_0 \quad e_1$

$$c_0 = (e_0 - \boxed{[e_0]_{p_l}}) \cdot p_l^{-1}$$
$$c_1 = (e_1 - \boxed{[e_1]_{p_l}}) \cdot p_l^{-1}$$

$c_1 \quad c_0$

Can be rewritten as **Euclidean division**

$$e_i = c_i \cdot p_l + [e_i]_{p_l}$$

**Not algebraic**

# Arithmetization

$a_0 \quad a_1 \quad b_0 \quad b_1$

$$(d_0, d_1, d_2) = (a_0, a_1) \otimes (b_0, b_1)$$

$d_0 \quad d_1 \quad d_2$

$$e_0 = d_0 + \langle \mathrm{evk}, \mathbf{w}_{\mathrm{ks}} \rangle$$
$$e_1 = d_1 + \langle \mathrm{evk}, \mathbf{w}_{\mathrm{ks}} \rangle$$

$e_0 \quad e_1$

$$c_0 = (e_0 - \boxed{[e_0]_{p_l}}) \cdot p_l^{-1}$$
$$c_1 = (e_1 - \boxed{[e_1]_{p_l}}) \cdot p_l^{-1}$$

$c_1 \quad c_0$

**Not algebraic**

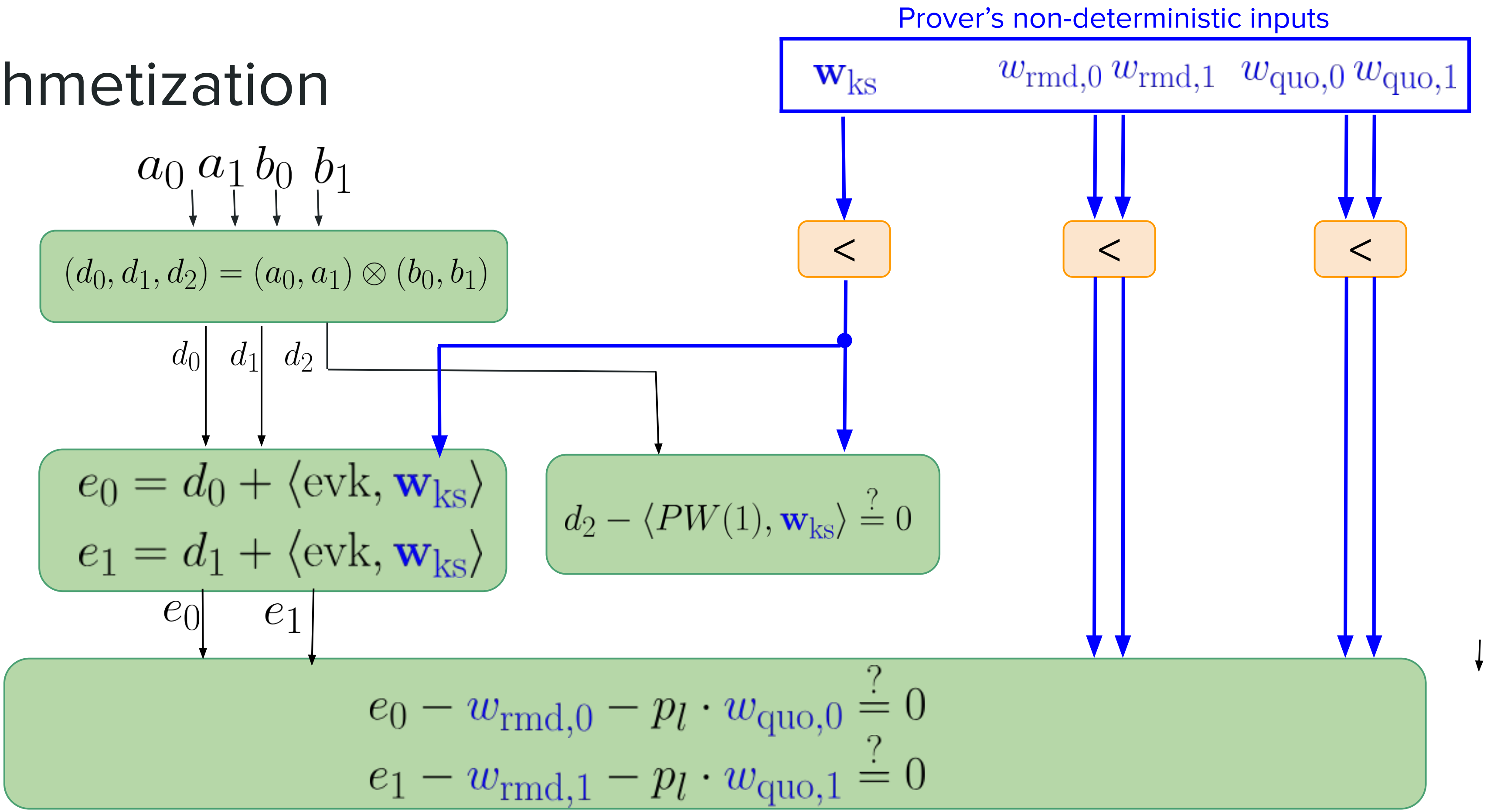
Can be rewritten as **Euclidean division**

$$e_i = c_i \cdot p_l + [e_i]_{p_l}$$

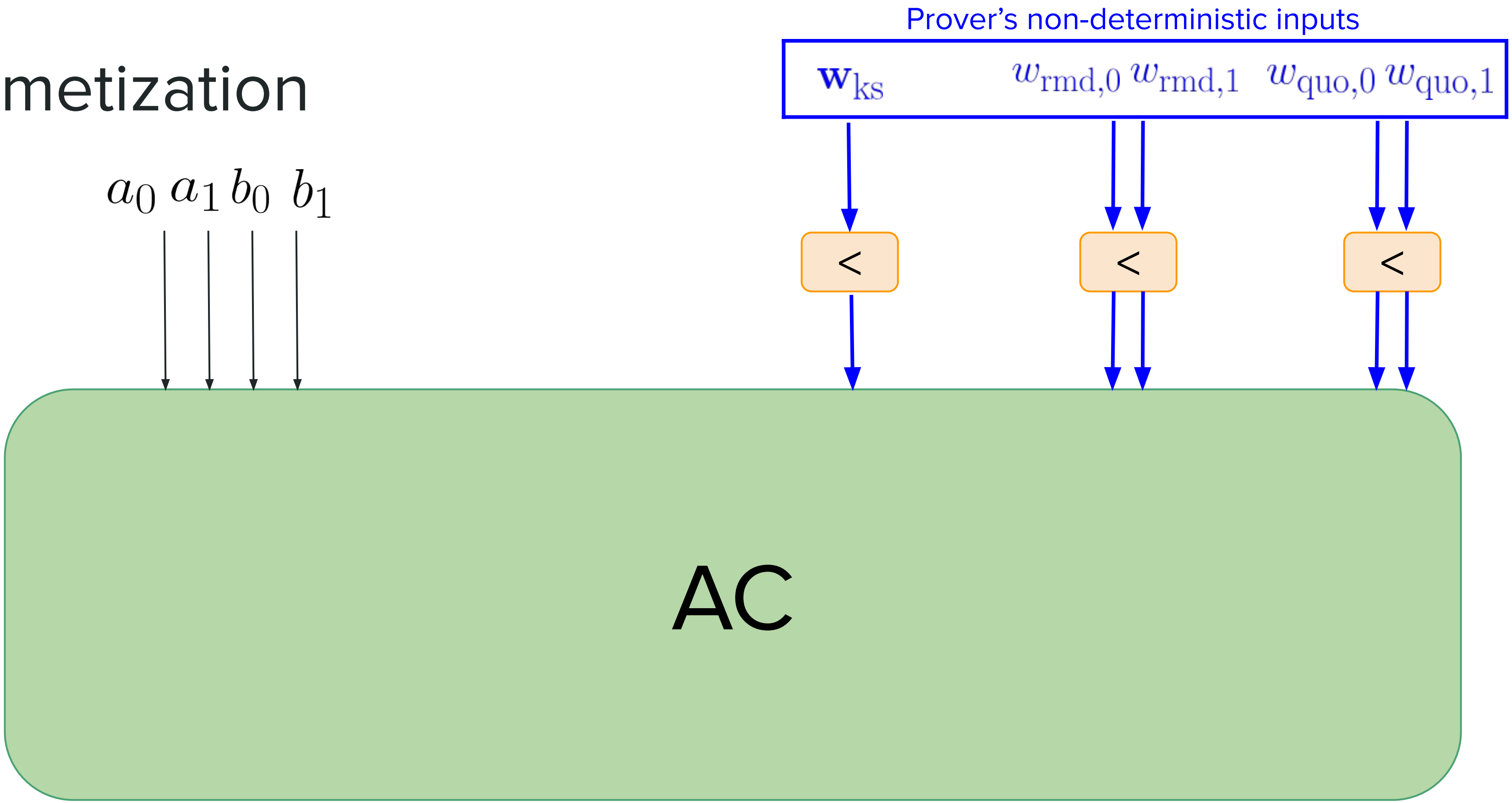
Prover inputs $w_{\mathrm{quo},i} \quad w_{\mathrm{rmd},i}$

$$\|w_{\mathrm{quo},i}\| \leq q_l / p_l$$
$$\|w_{\mathrm{rmd},i}\| < p_l$$
$$e_i = w_{\mathrm{quo},i} \cdot p_l + w_{\mathrm{rmd},i}$$

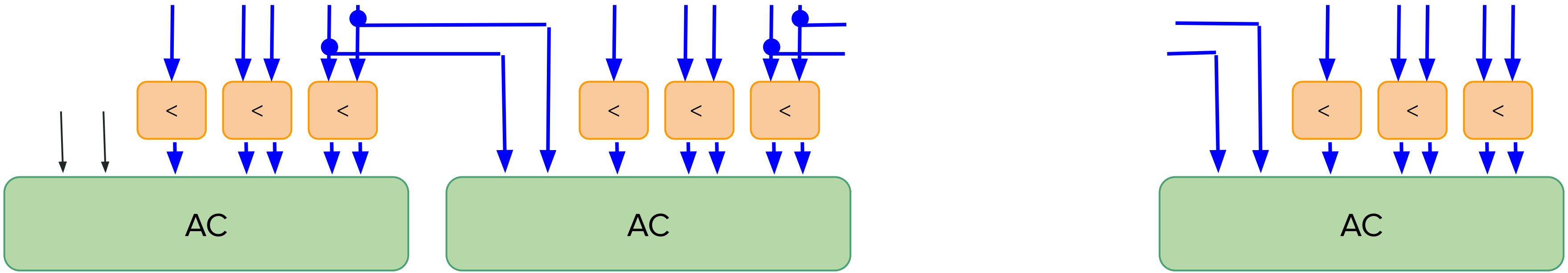
# Arithmetization



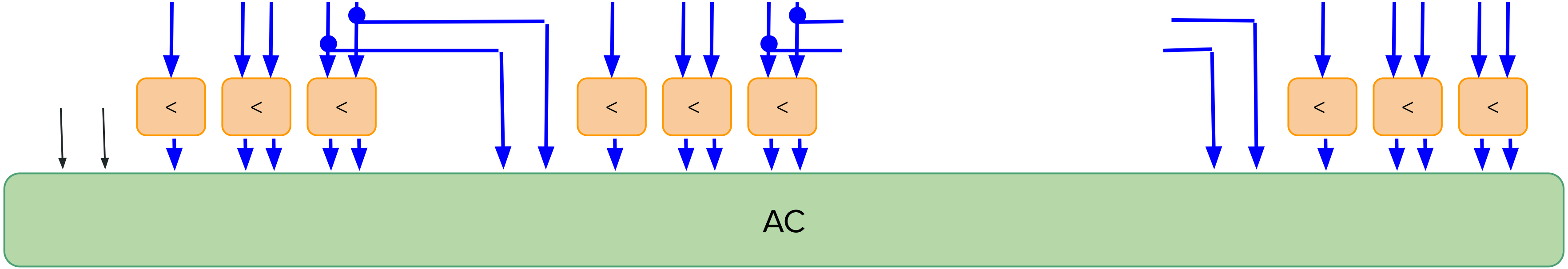
# Arithmetization



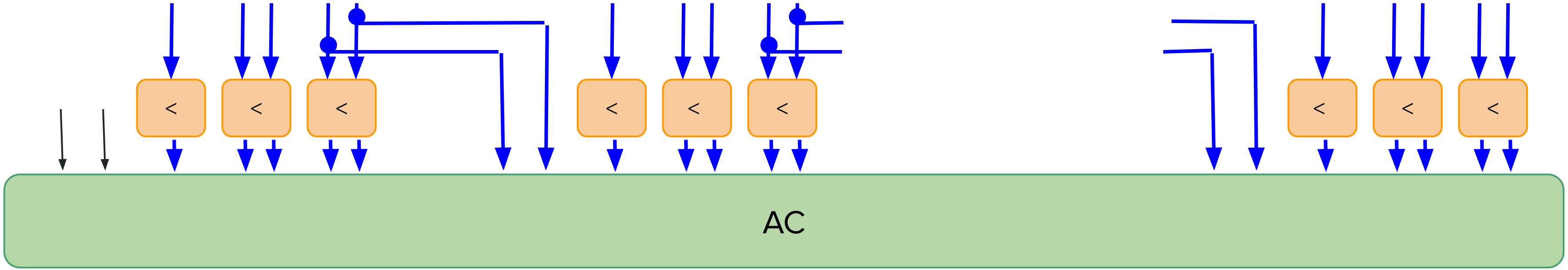
# Flattening the circuit



# Flattening the circuit



# Final set of relations




**HE circuit** satisfiability reduced to **two** main **relations**:

$$\mathcal{R}_{AC} := \{(C; \mathbf{x}, \mathbf{w}) : C(\mathbf{x}, \mathbf{w}) = \mathbf{0}\}$$

$$\mathcal{R}_{range} := \{(B; \mathbf{w}) : \|\mathbf{w}\| < B\}$$

# Proof of AC satisfiability

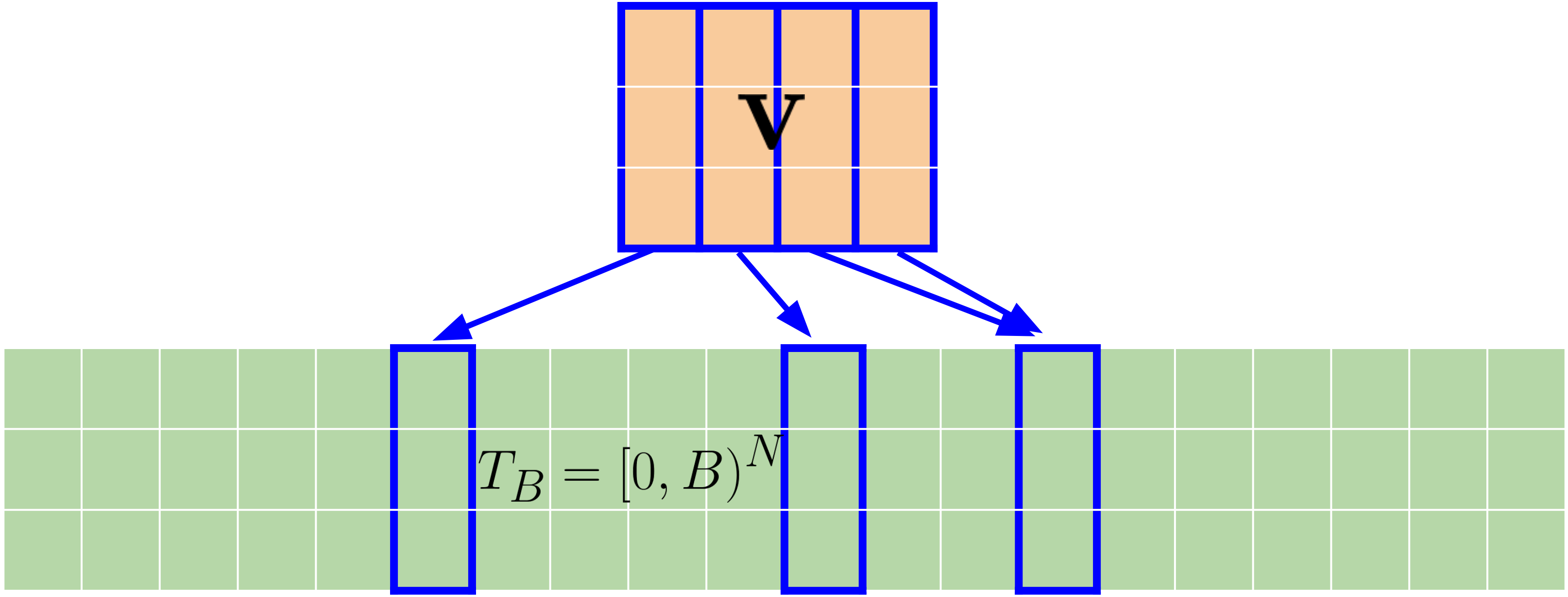
# Proving AC satisfiability

- GKR-style proof system
    - AC satisfiability => layers of equations
    - Consistency of layer *i*-th reduced to that of layer *i+1*-th
    - Works over $\boldsymbol{R_q}$
    - **Custom gates** (rescon, bdcon, ...)
    - Flattened system of relations => **constant depth 4**

# Range checks

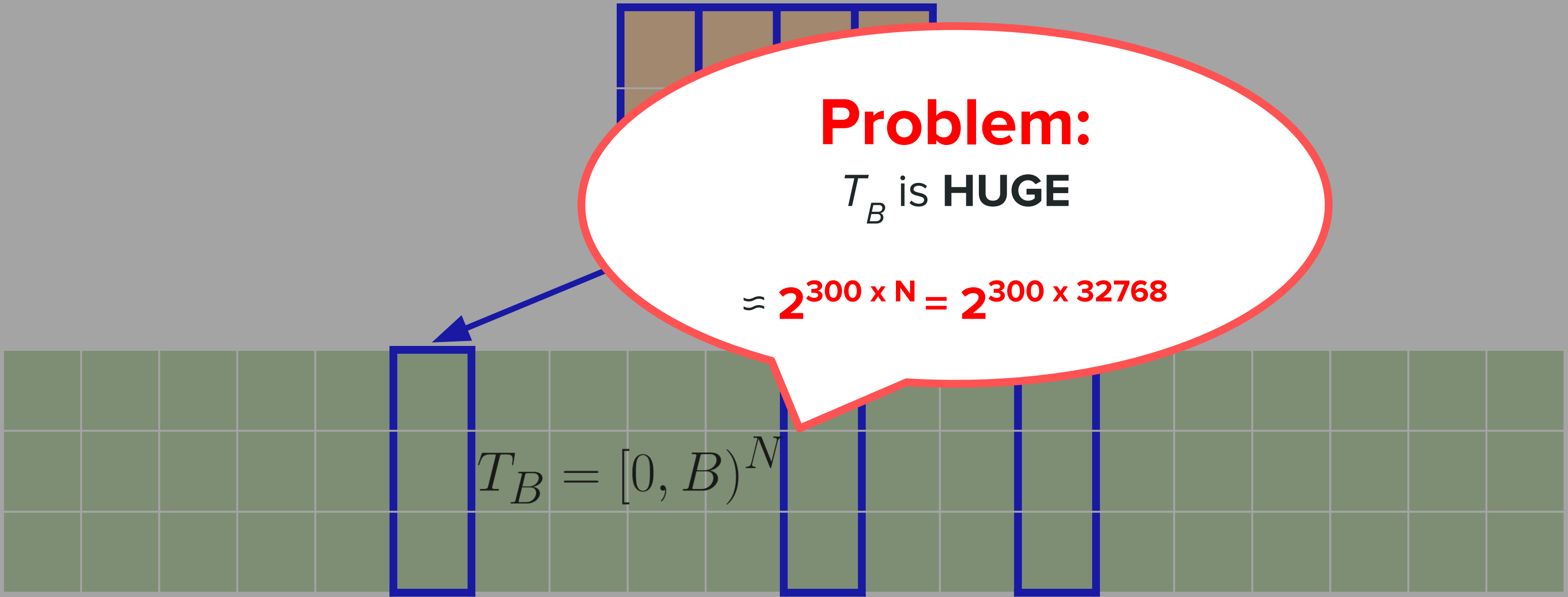
# Proving ranges

- Prove that vector **v** of *m* elements in $R_q$ has coeffs bounded by B (e.g B = $q_l$ )
- Can be seen as a look-up argument



# Proving ranges

- Prove that vector **v** of *m* elements in $R_q$ has coeffs bounded by B (e.g B = $q_l$ )
- Can be seen as a look-up argument

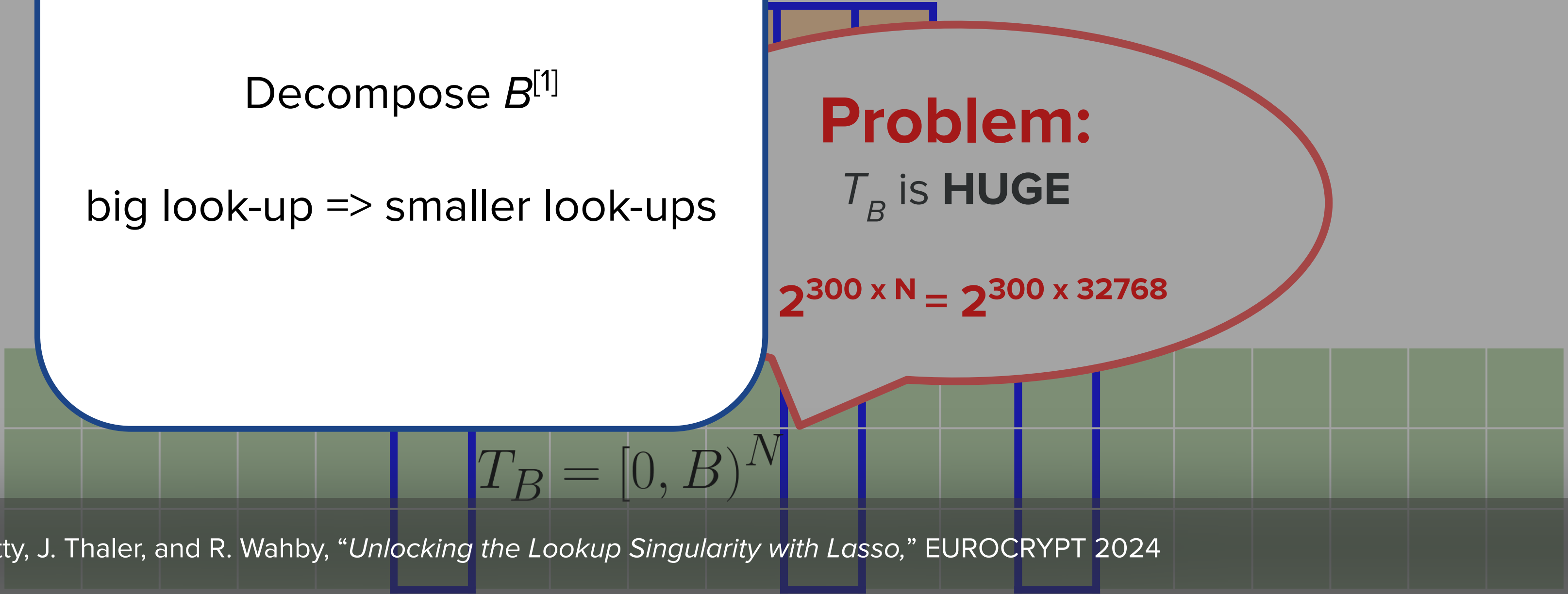


# Proving ranges

- Prove that vector **v** of *m* elements in $R_l$ has coeffs bounded by B (e.g B = $q_l$ )
- C

Decompose $B$[1]

big look-up => smaller look-ups




**Problem:**

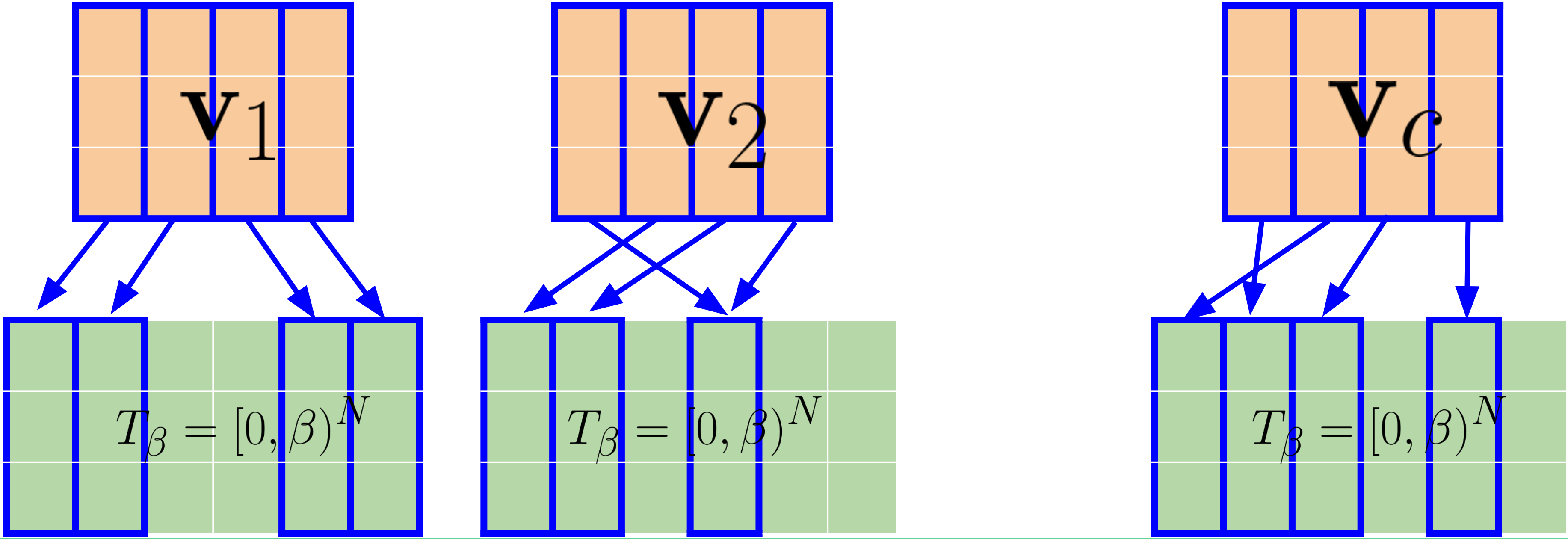
$T_B$ is **HUGE**

$2^{300 \times N} = 2^{300 \times 32768}$

[1] S. Setty, J. Thaler, and R. Wahby, "*Unlocking the Lookup Singularity with Lasso,*" EUROCRYPT 2024

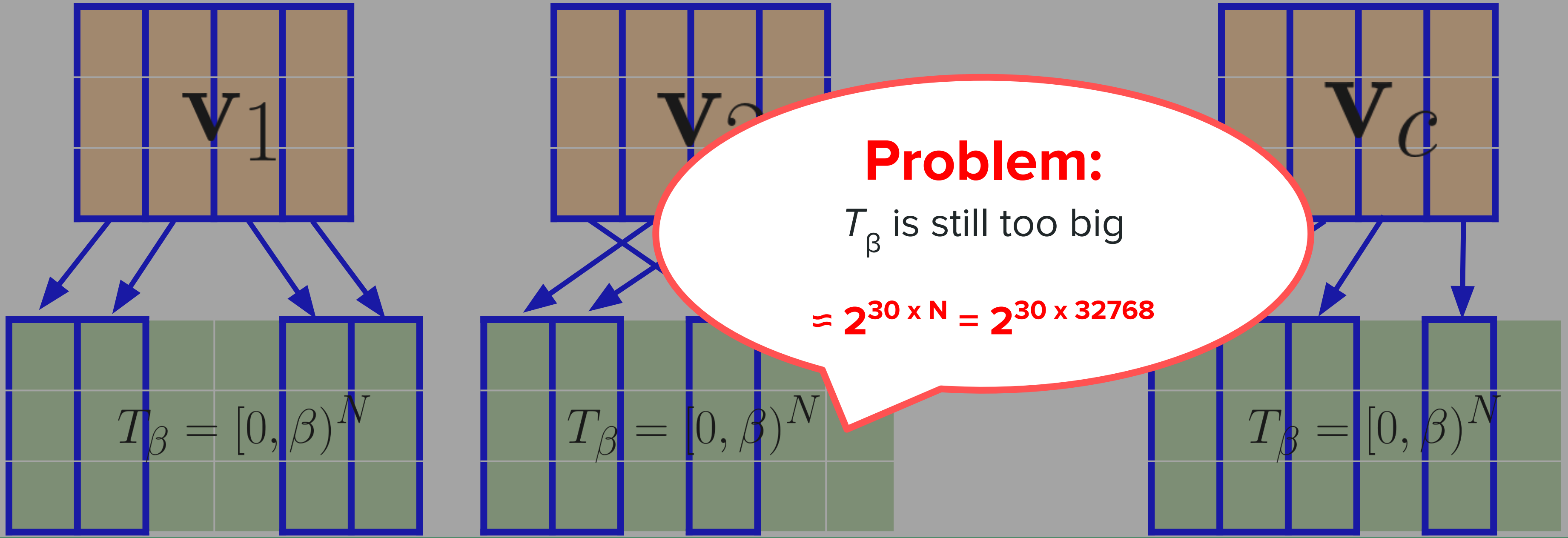
# Proving ranges

- Set $\beta = B^{1/c}$
- Prove that $c$ vectors $\mathbf{v}_1, \ldots, \mathbf{v}_c$ of $m$ elements in $R_q$ have coeffs bounded by $\beta$



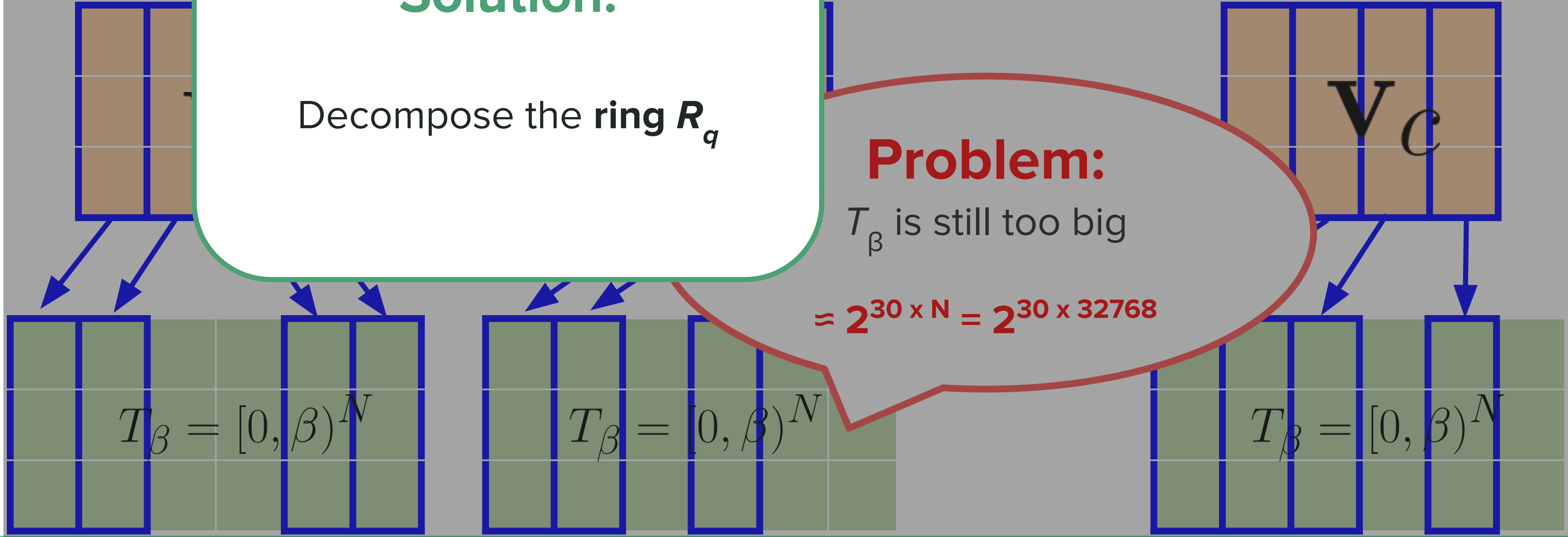
# Proving ranges

- Set $\beta = B^{1/c}$
- Prove that $c$ vectors $\mathbf{v}_1, \ldots, \mathbf{v}_c$ of $m$ elements in $R_q$ have coeffs bounded by $\beta$



# Proving ranges

- Set
- Prov n $R_q$ have coeffs bounded by β




**Solution:**

Decompose the **ring** $\boldsymbol{R_q}$

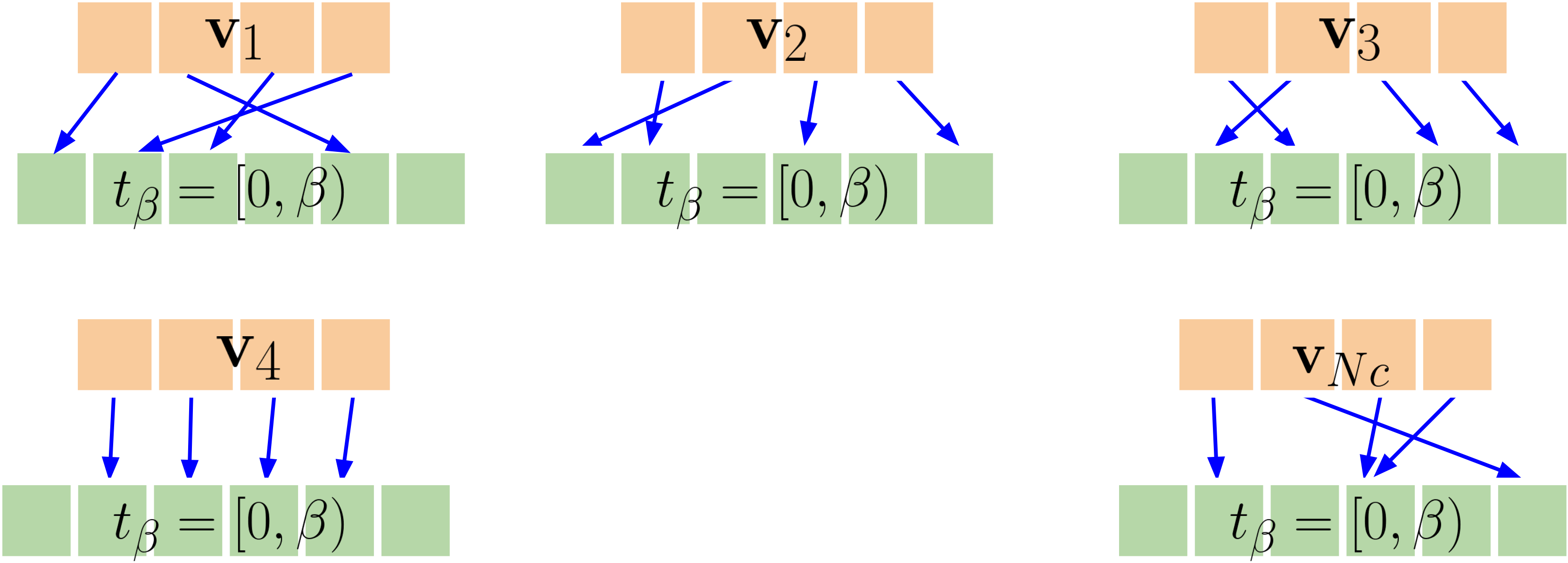
**Problem:**

$T_\beta$ is still too big

$\approx 2^{30 \times N} = 2^{30 \times 32768}$

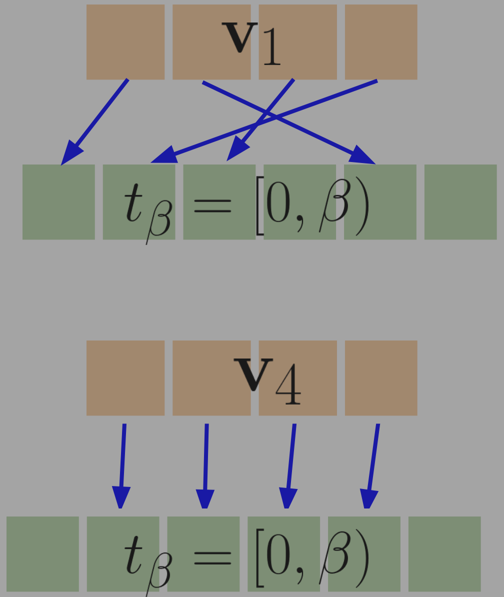
# Proving ranges

- Prove that $cN$ vectors $\mathbf{v}_1, \ldots, \mathbf{v}_{cN}$ of $m$ elements in $Z_q$ have coeffs bounded by β:



# Proving ranges

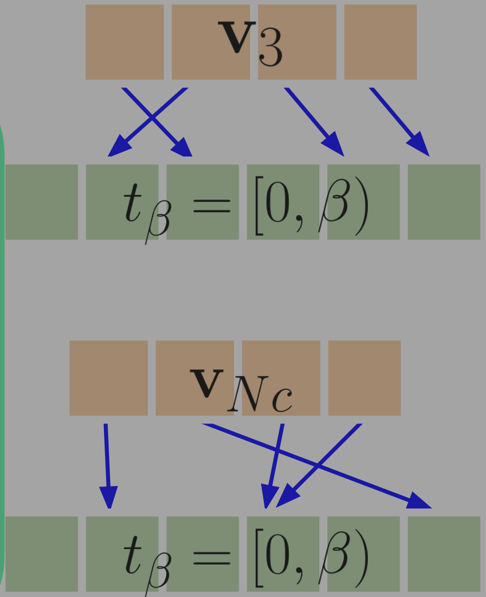
- Prove that $cN$ vectors $\mathbf{v}_1, \ldots, \mathbf{v}_{cN}$ of $m$ elements in $Z_q$ have coeffs bounded by β:




$t_\beta \approx 2^{30}$

This is now a normal look-up argument

Can be done using **memory-check** techniques



# The polynomial commitment

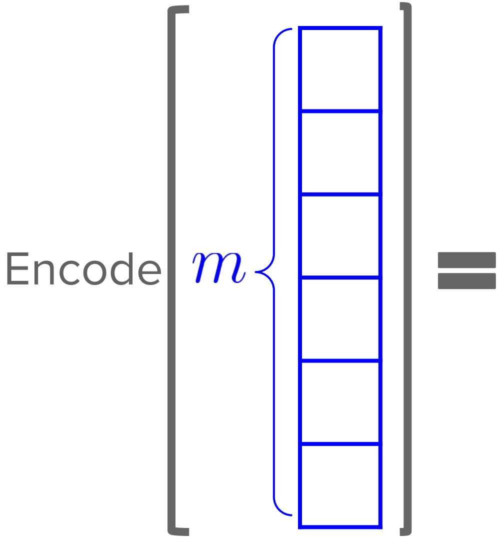
# Polynomial Commitment

Need to commit to elements in $R_q[X_1, \ldots, X_\ell]$ where

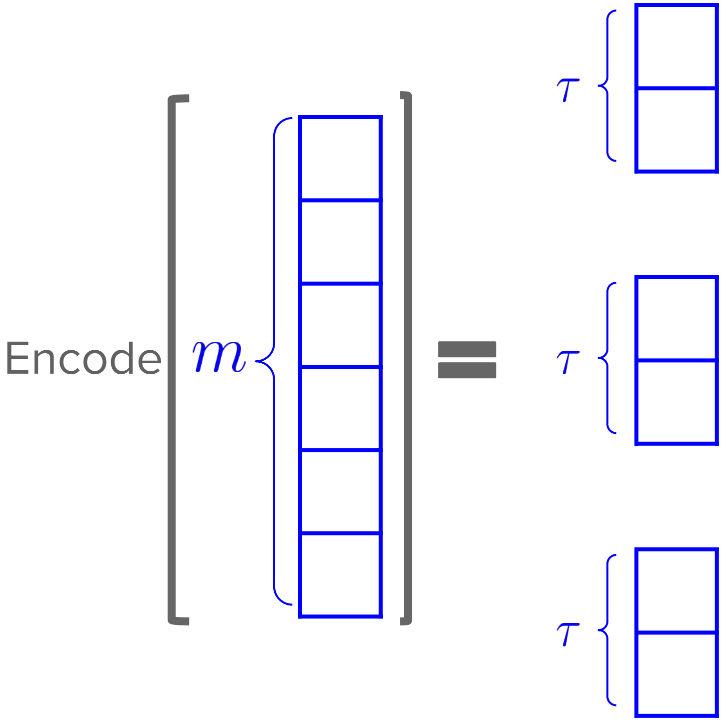
$$R_q \cong \mathbb{F}_{p_0^4} \times \cdots \times \mathbb{F}_{p_L^4}$$

- Reduce MV PC over $R_q$ to MV PC over $\mathbb{F}_{p_i^4}$
- Field-agnostic multivariate PC => **Brakedown**
- $\mathbb{F}_{p_i^4}$ has $N/2$ roots of unity. **Can we use them?**

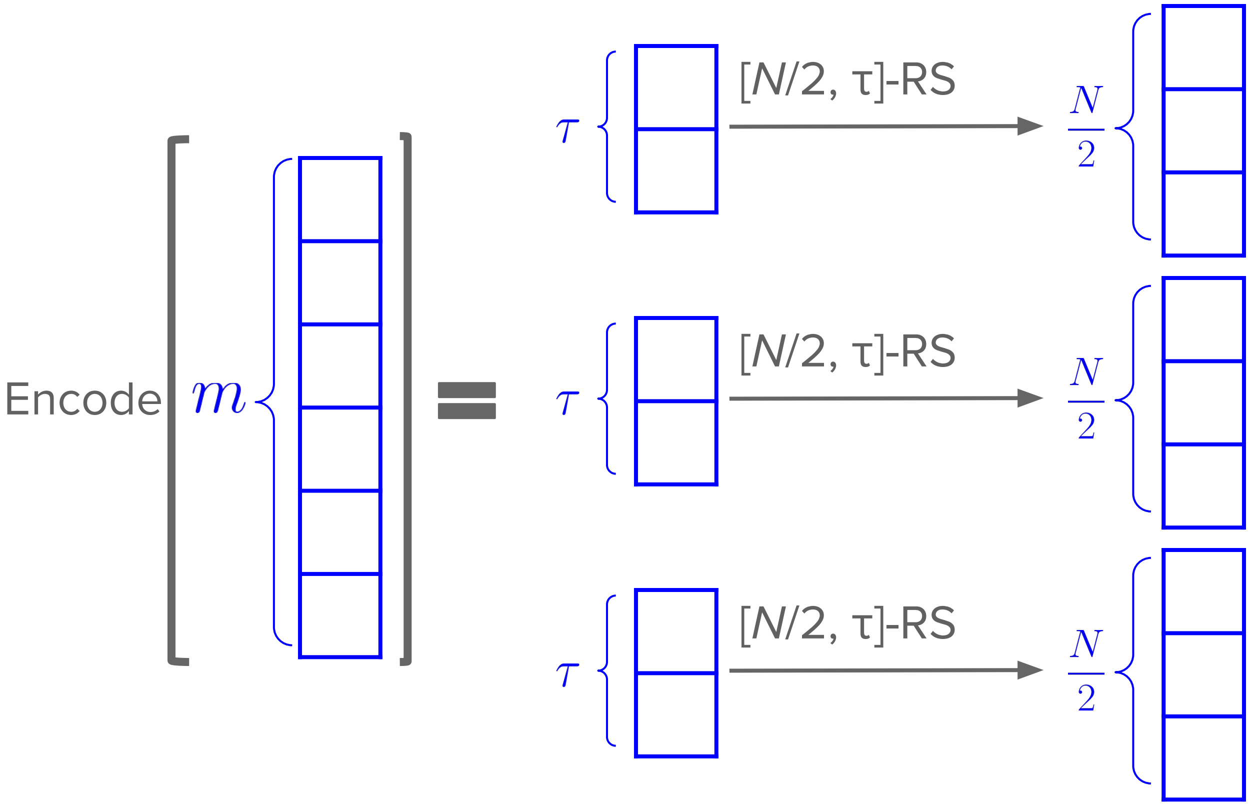
# Piecewise Reed-Solomon code



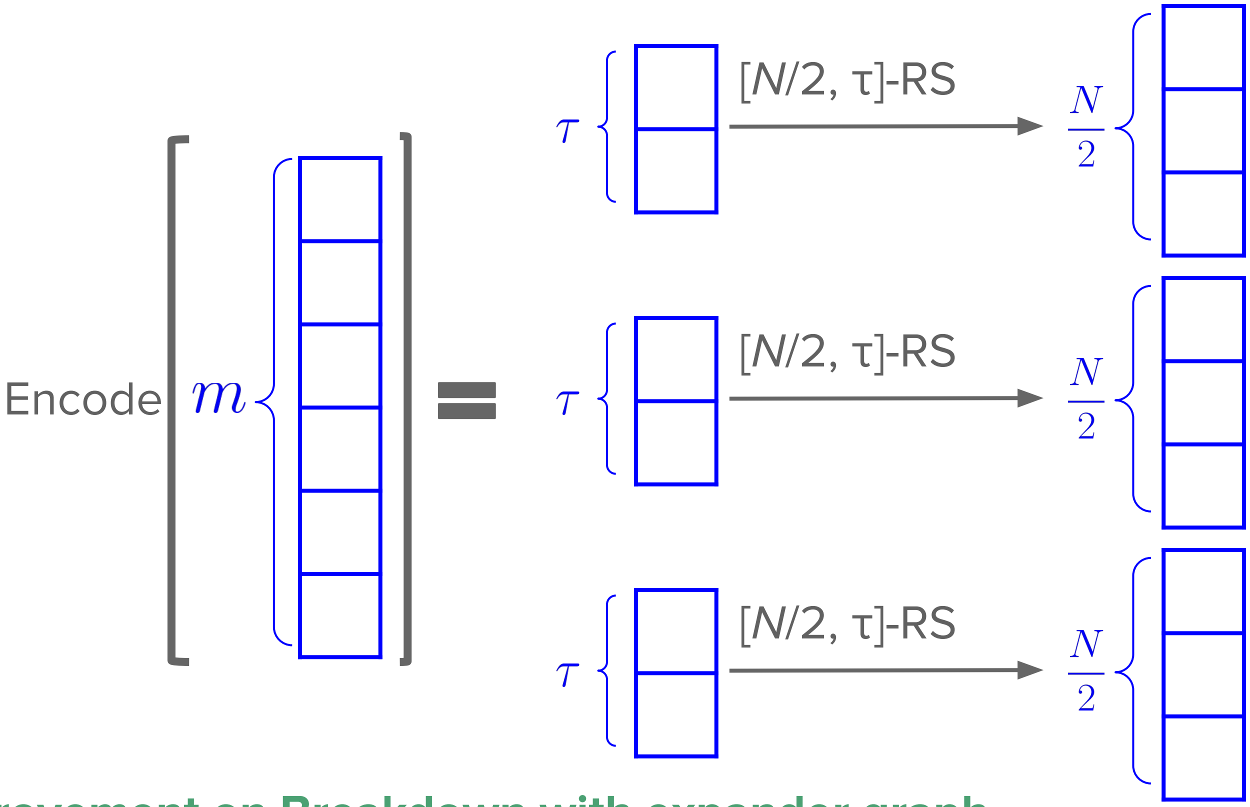
# Piecewise Reed-Solomon code



# Piecewise Reed-Solomon code



# Piecewise Reed-Solomon code




**x10 improvement on Breakdown with expander graph**

# Conclusions

# To summarize

- First practical VC for CKKS
  - Technique extend to FV/BGV
- Description of problem in a modular way (arithmetization)
  - AC satisfiability + range checks
- Design of proof-friendly CKKS
- Design of custom GKR to prove AC over rings
- Design of range proofs for polynomial rings
- Improved Brakedown for medium-sized fields
- Implemented all building blocks

https://eprint.iacr.org/2025/286

# Thank you!

This work is supported by the PICOCRYPT project that has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (Grant agreement No. 101001283), partially supported by projects PRODIGY (TED2021-132464B- I00) and ESPADA (PID2022-142290OB-I00) funded by MCIN/AEI/10.13039/501100011033/. This work is part of the grant JDC2023-050791-I, funded by MCIN/AEI/10.13039/501100011033 and the ESF+. This work is also supported by the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme in the scope of the CONFIDENTIAL6G project under Grant Agreement 101096435. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

# Images used in this presentation

- User faces: “Plump Interface Duotone Icons” by Streamline, Creative Commons Attribution 4.0 International, available at https://iconduck.com/sets/plump-interface-duotone-icons

# VC-HE

| | Verification | Arithmetic operations | Ciphertext maintenance | Bootstrapping | Supported HE schemes |
|---|---|---|---|---|---|
| Generic SNARK | Public | emulated | emulated | emulated | Any |
| Rinocchio | Private | Native | emulated | emulated | Any |
| HE-IOPs | Private **VO attacks** | Native | Native | Native | Exact (no CKKS) |
| **Our Work** | Public | Native | Efficiently emulated | ? | Any |