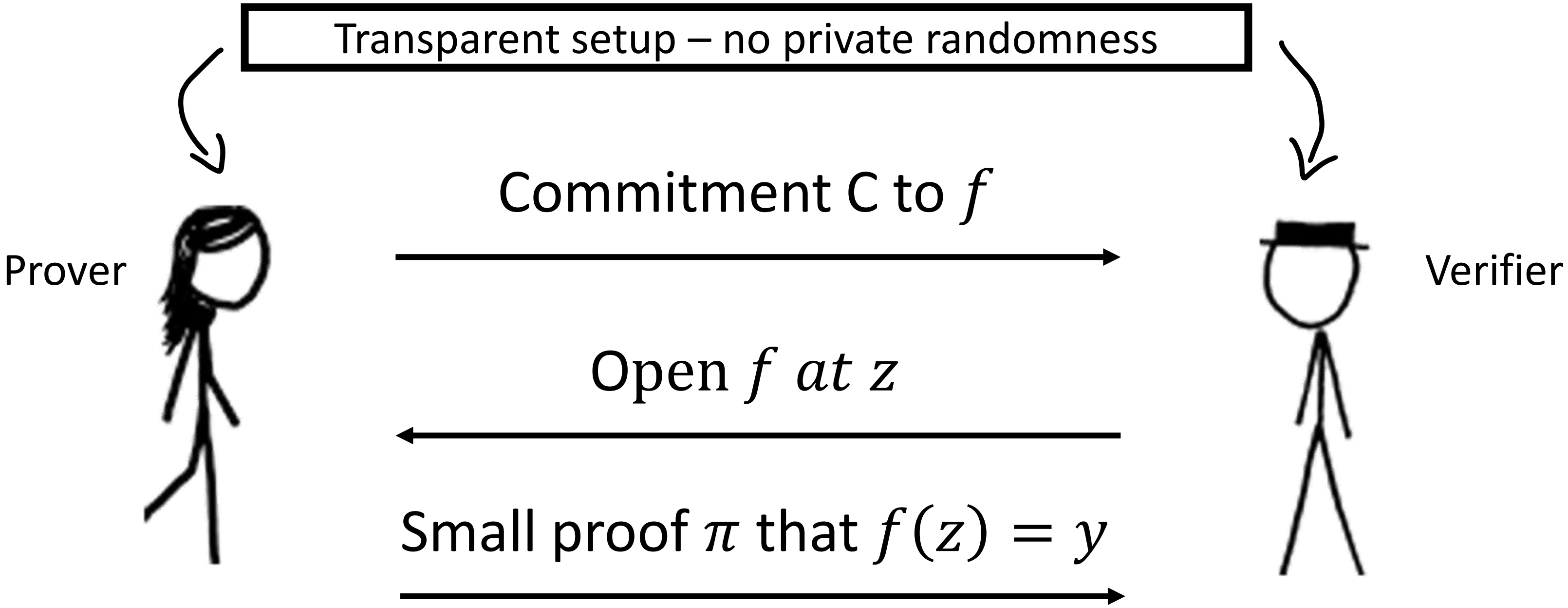
Dew: Transparent Constant-sized Polynomial Commitment Scheme

Arasu Arun	*New York University*
Chaya Ganesh	*Indian Institute of Science*
Satya Lokam	*Microsoft Research India*
Tushar Mopuri	*Indian Institute of Science*
Sriram Sridhar	*UC Berkeley*

Polynomial Commitment Schemes

$f(X) \in \mathbb{F}_p[X]$ s.t. $\deg(\mathrm{f}) \leq d$



Transparent setup – no private randomness

Prover

Verifier

Commitment C to f

Open f at z

Small proof π that $f(z) = y$

PCS – Properties

Completeness:

Extractability:
∃ efficient extractor that outputs a decommitment f to C that satisfies $f(z) = y$.

+ binding of the commitment scheme

Succinctness:
Commitment, proof size must be "small"
Verifier efficiency should be sublinear

(Hiding, ZK)

Main result

We construct a polynomial commitment scheme with

- Transparent setup
- Succinct commitments and opening proofs - $poly(\kappa)$
- Logarithmic verifier - $poly(\kappa) \cdot \log(\deg(f))$

feat. Groups of Unknown Order (Class groups)
Generic Group Model

$\Downarrow$

Proof of Knowledge of Exponent (PoKE) – BBF'19

$\Downarrow$ *

Extraction

Will focus on soundness

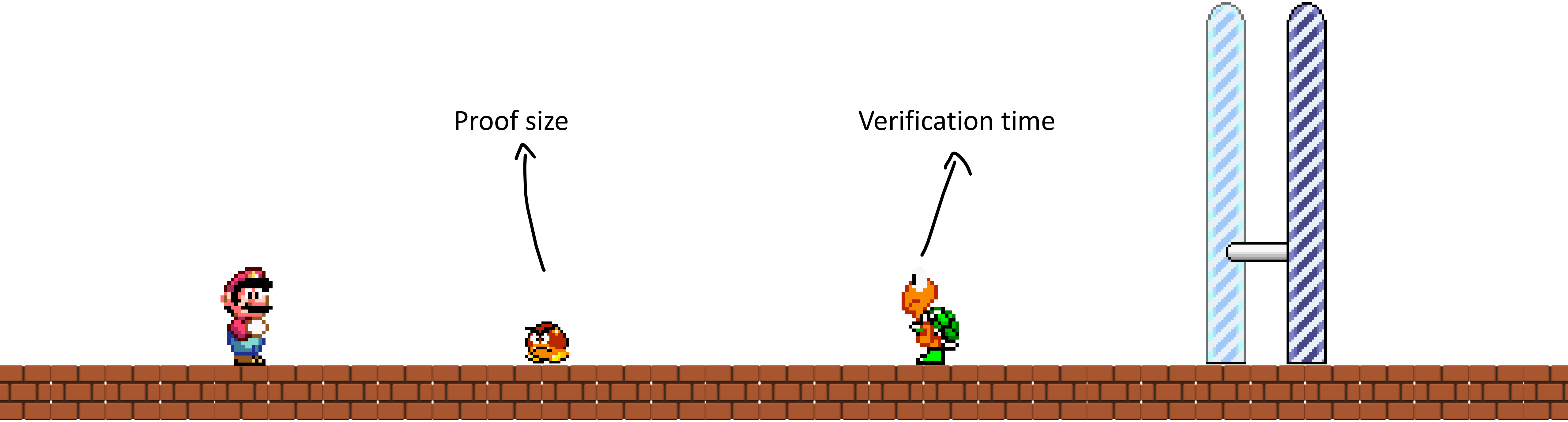
Other results

- Hiding and ZK variants of PCS
- Transparent Constant sized zkSNARKs
- DARK fix [BFS20] with increased prover time

  Also patched by [BHRRS21] as well as [eprint:BFS20]

(Details in paper)

Roadmap

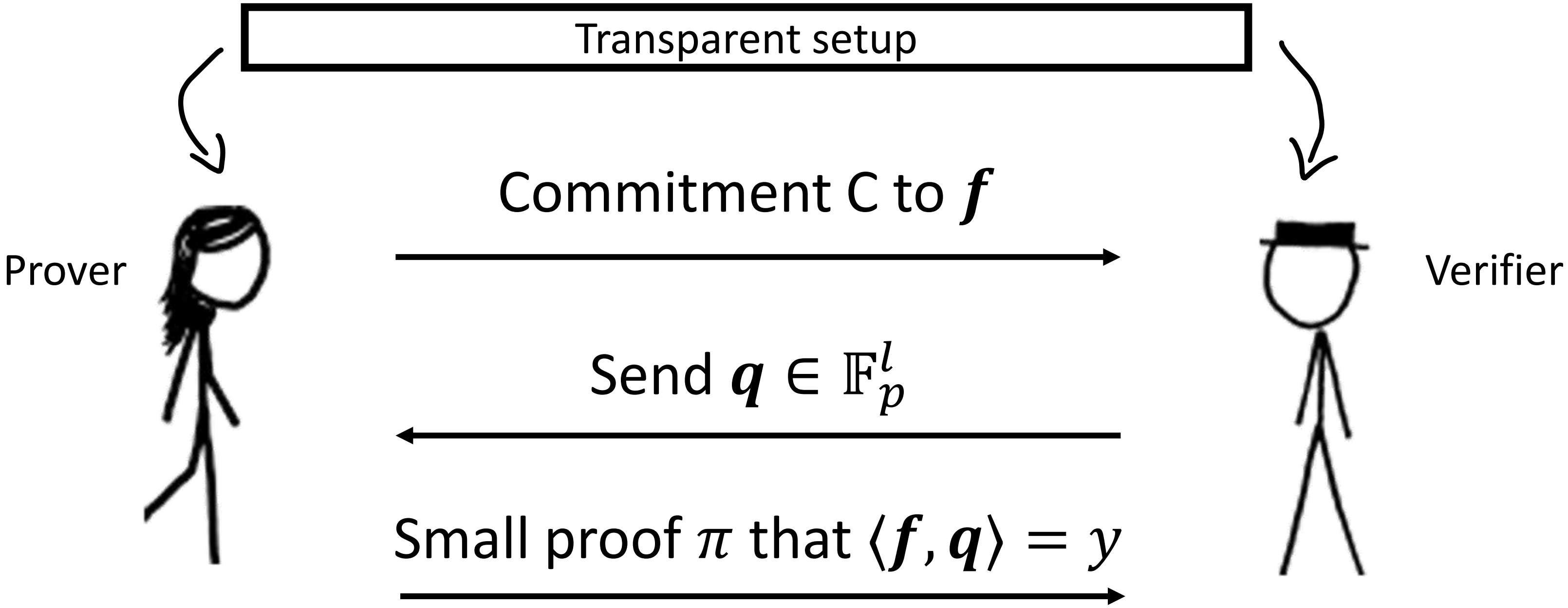


Proof size

Verification time

Inner Product Commitments $\Rightarrow \boldsymbol{PCS}$

$\boldsymbol{f} \in \mathbb{F}_p^l$



Transparent setup

Prover

Verifier

Commitment C to $\boldsymbol{f}$

Send $\boldsymbol{q} \in \mathbb{F}_p^l$

Small proof π that $\langle \boldsymbol{f}, \boldsymbol{q} \rangle = y$

Constant proof size?

(No constraints on verification time)

Encoding vectors/polynomials

Let $f(X) \equiv f_0 + f_1 X + f_2 X^2 + \cdots + f_\ell X^\ell \in \mathbb{F}_p[X]$

$$\boxed{\mathrm{int}_{\boldsymbol{\alpha}}(\boldsymbol{f}) \equiv f_0 + f_1\,\alpha + f_2\,\alpha^2 + \cdots + f_\ell\,\alpha^\ell \in \mathbb{Z}}$$

$$\boxed{\boldsymbol{f} \equiv (f_0, f_1, \cdots, f_\ell) \in [0, p-1)^{\ell+1}}$$

$\alpha \gg p$ & α is public

Encoding in base – α

$$\boxed{Com(f) := g^{int_\alpha(f)}}$$

(!) Groups of Unknown Order (GUOs) give us binding over integers; cannot open to both x and $x + n|G|$ as $|G|$ is unknown

Intuition – Inner products

$$\langle \boldsymbol{f}, \boldsymbol{q} \rangle = \sum_{i=0}^{\ell} f_i \, q_i$$

$int_\alpha(\boldsymbol{f})$

1	α			$\alpha^{\ell-1}$
f_0	f_1			$f_{\ell-1}$

$\cdot$

$int_\alpha(\boldsymbol{rev}(\boldsymbol{q}))$

1	α			$\alpha^{\ell-1}$
$q_{\ell-1}$	$q_{\ell-2}$			q_0

1	α			$\alpha^{\ell-1}$			$\alpha^{2\ell-2}$
$f_0 q_{\ell-1}$	$f_0 q_{\ell-2} + f_1 q_{\ell-1}$			$\sum f_i q_i$			$f_{l-1} q_0$

Intuition – Inner products

$$int_\alpha(f) \cdot \underbrace{int_\alpha(rev(q))}_{\substack{\sigma \\ \text{Verifier can compute}}} = L + \underbrace{\langle\langle f,q \rangle\rangle}_{\underbrace{\langle f,q \rangle}_{v\ \boxed{\leftarrow\ \text{Claimed inner product}}} + np} \cdot \alpha^\ell + H$$

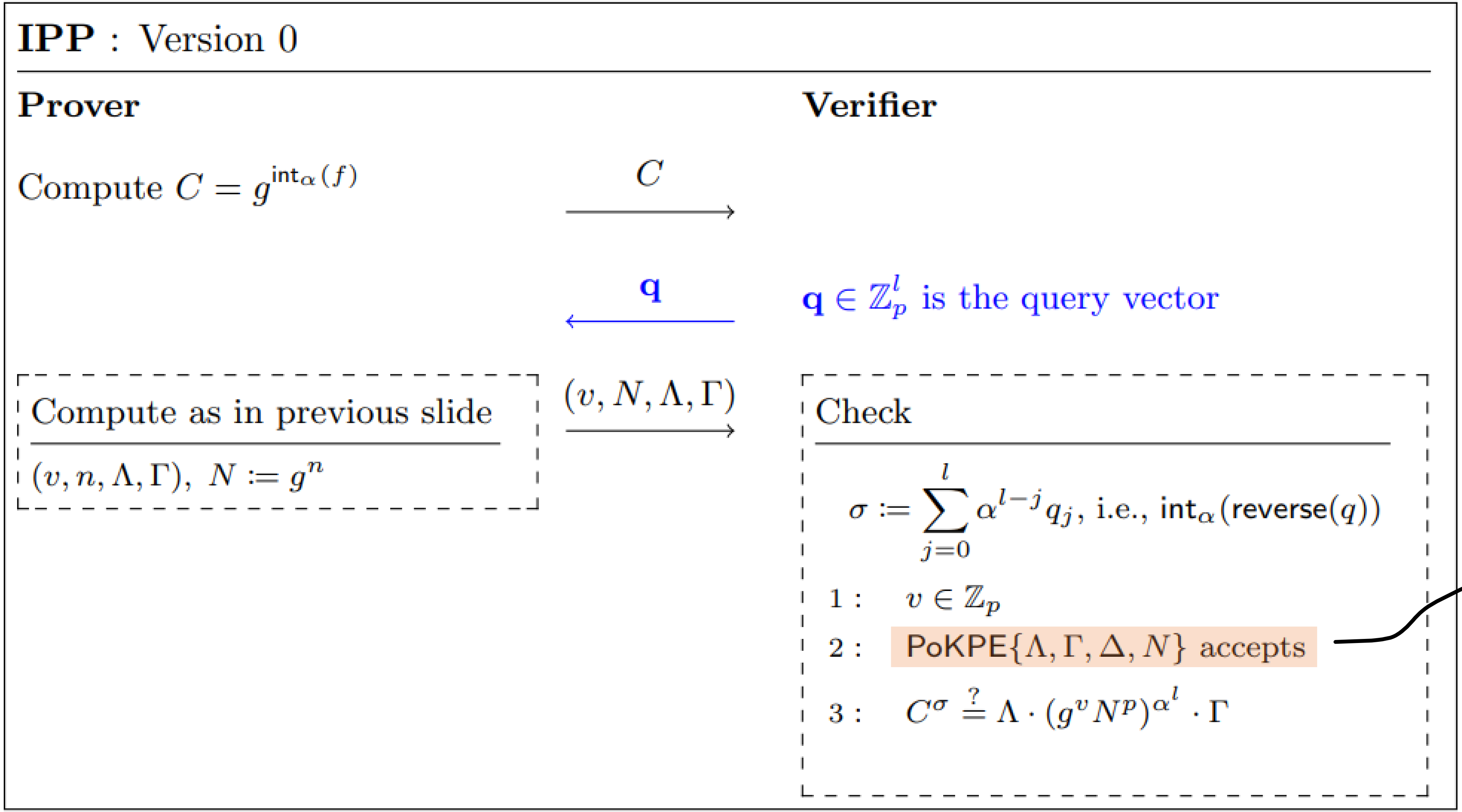
Putting both sides in the exponent of g, and since $C = g^{int_\alpha(f)}$

Verifier checks $$C^\sigma \overset{?}{=} g^L \cdot (g^v \cdot g^{np})^{\alpha^\ell} \cdot g^H$$

Λ N Γ

Prover's Commitment

Prover sends



IPP : Version 0

Prover

Verifier

Compute $C = g^{\mathsf{int}_\alpha(f)}$

$\xrightarrow{C}$

$\xleftarrow{\mathbf{q}}$

$\mathbf{q} \in \mathbb{Z}_p^l$ is the query vector

Compute as in previous slide

$(v, n, \Lambda, \Gamma),\ N := g^n$

$\xrightarrow{(v, N, \Lambda, \Gamma)}$

Check

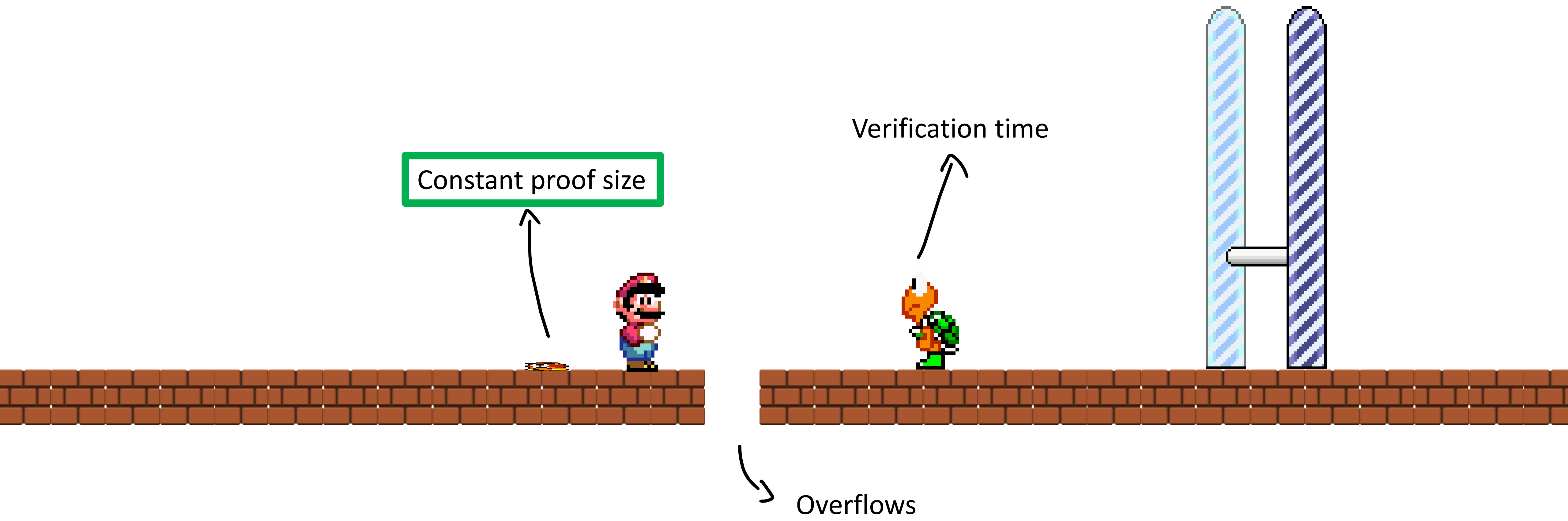
$$\sigma := \sum_{j=0}^{l} \alpha^{l-j} q_j, \text{ i.e., } \mathsf{int}_\alpha(\mathsf{reverse}(q))$$

1 : $v \in \mathbb{Z}_p$

2 : $\mathsf{PoKPE}\{\Lambda, \Gamma, \Delta, N\}$ accepts

3 : $C^\sigma \overset{?}{=} \Lambda \cdot (g^v N^p)^{\alpha^l} \cdot \Gamma$

Range proof using PoKE [BBF19]



Constant proof size

Verification time

Overflows

Overflow

A cheating prover can choose coefficients of $\boldsymbol{f}$ outside $\mathbb{Z}_p$.

This will cause “overflows” in the basic equation for inner product
(by violating the “sufficiently large” condition on α)

$$
\begin{aligned}
&int_\alpha(f) \cdot int_\alpha\big(rev(q)\big) \\
&= (f_0 + f_1\alpha + \cdots + f_\ell\alpha^\ell) \cdot (q_\ell + q_{\ell-1}\alpha + \cdots + q_0\alpha^\ell) \\
&= f_0 q_\ell + (f_0 q_{\ell-1} + f_1 q_\ell)\,\alpha + \cdots + \\
&\quad + (f_0 q_1 + f_1 q_2 + \cdots + f_{\ell-1} q_\ell)\,\alpha^{\ell-1} + (f_0 q_0 + f_1 q_1 + \cdots + f_\ell q_\ell)\,\alpha^\ell + \\
&\quad + \cdots + f_\ell q_0\,\alpha^{2\ell}
\end{aligned}
$$

Basic Equation

Controlling the overflow

Encoding in base – α

- Intersperse 0’s in $\boldsymbol{f}$: $\boxed{f_0 \quad 0 \quad f_1 \quad 0 \quad f_2 \quad 0 \quad \cdots \quad \cdots \quad \cdots \quad f_\ell \quad 0}$

  Query vector $\boldsymbol{q}$: $\boxed{q_0 \quad 0 \quad q_1 \quad 0 \quad q_2 \quad 0 \quad \cdots \quad \cdots \quad \cdots \quad q_\ell \quad 0}$

Honest prover

Commitment $C = g^{int_{\alpha^2}(f)}$, where $int_{\alpha^2}(f) = \sum f_i \, \alpha^{2i}$

Cheating Prover

- Test that the prover indeed used 0’s in odd positions

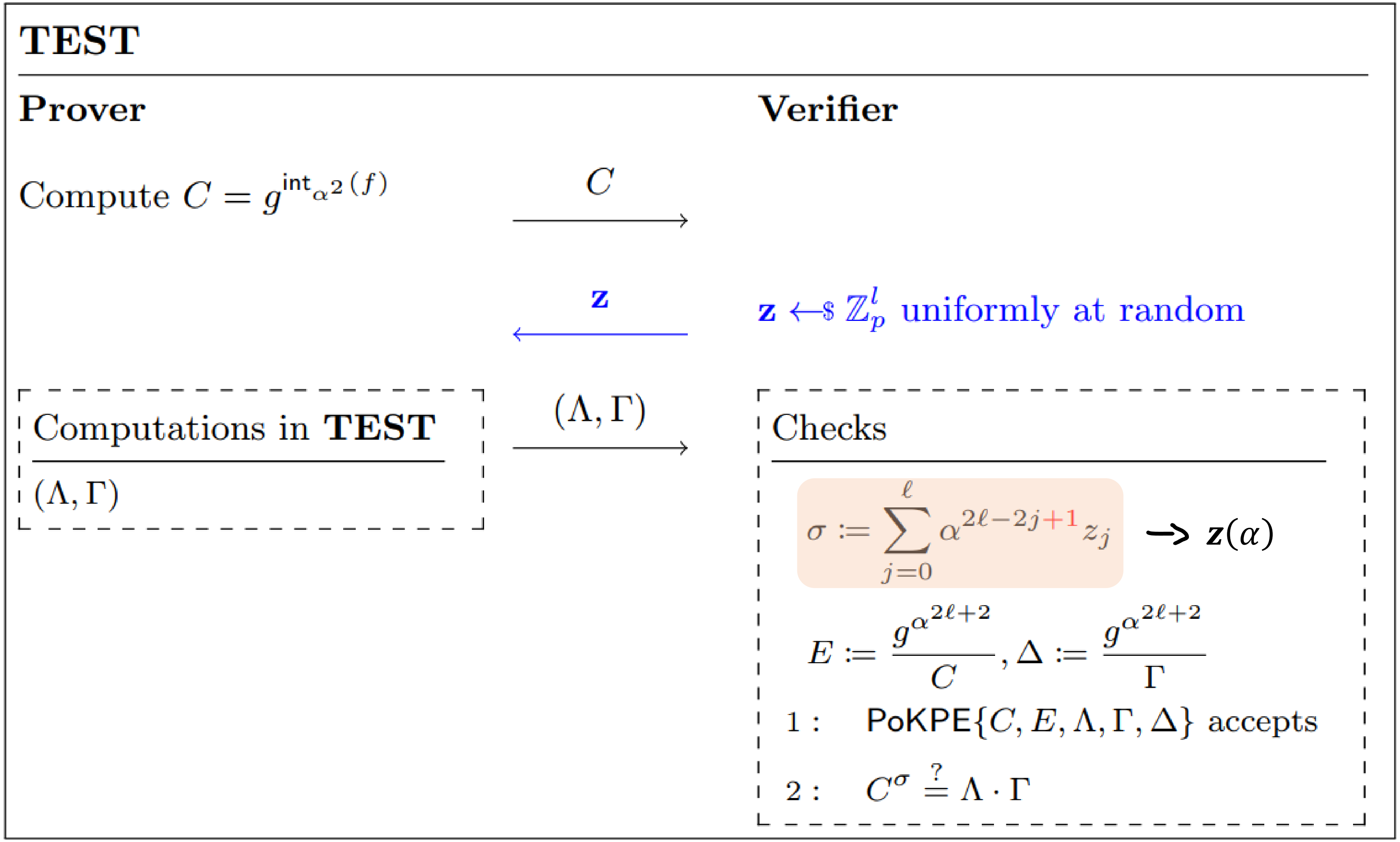
$(0 \leq d_i \leq \alpha - 1)$

$\boldsymbol{f}$: $\boxed{f_0 \quad d_0 \quad f_1 \quad d_1 \quad f_2 \quad d_2 \quad \cdots \quad \cdots \quad \cdots \quad f_\ell \quad d_\ell}$

$$int_{\alpha^2}(f) = \sum (f_i + \alpha d_i) \cdot \alpha^{2i}$$

Random query $\boldsymbol{z}$: $\boxed{0 \quad z_0 \quad 0 \quad z_1 \quad 0 \quad z_2 \quad \cdots \quad \cdots \quad \cdots \quad 0 \quad z_\ell}$

Inner product $\langle f, z \rangle$ “must” be 0



TEST

Prover

Verifier

Compute $C = g^{\mathsf{int}_{\alpha^2}(f)}$

$\xrightarrow{C}$

$\xleftarrow{\mathbf{z}}$

$\mathbf{z} \leftarrow_{\$} \mathbb{Z}_p^l$ uniformly at random

Computations in **TEST**

(Λ, Γ)

$\xrightarrow{(\Lambda, \Gamma)}$

Checks

$$\sigma := \sum_{j=0}^{\ell} \alpha^{2\ell-2j+1} z_j$$

$\rightarrow$ $z(\alpha)$

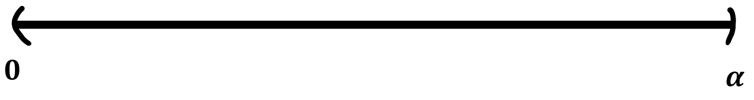
$$E := \frac{g^{\alpha^{2\ell+2}}}{C}, \Delta := \frac{g^{\alpha^{2\ell+2}}}{\Gamma}$$

1 : $\mathsf{PoKPE}\{C, E, \Lambda, \Gamma, \Delta\}$ accepts

2 : $C^{\sigma} \overset{?}{=} \Lambda \cdot \Gamma$

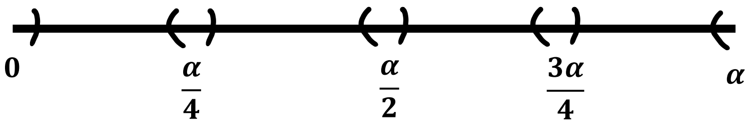
Structure on d_i

- Cannot show that $d_i = 0$
- But,



$\boxed{0 \le d_i \le \alpha - 1}$

vs.



etc.

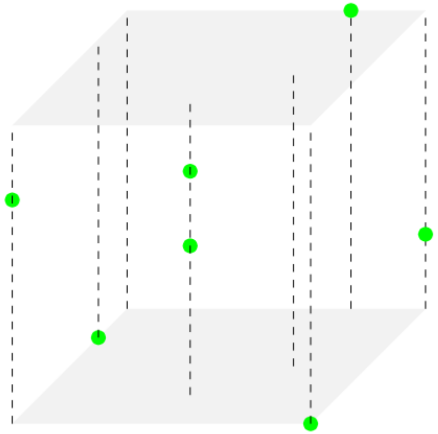
$\boxed{d_i = \frac{m_i\alpha - n_i}{k_i}}$ where $m_i, n_i, k_i \ll \alpha$

$\Rightarrow \frac{d_i}{\alpha}$ is very close to rationals with small denominators

This suffices!

Structure on d_i

$$\boxed{\underbrace{\langle \mathbf{d}, \mathbf{z} \rangle \mod \alpha}_{\text{Honest term}} + \underbrace{\left\lfloor \frac{\langle \mathbf{f}, \mathbf{z} \rangle}{\alpha} \right\rfloor + u \mod \alpha}_{\text{Overflow}} = 0 \mod \alpha}$$



Essentially,

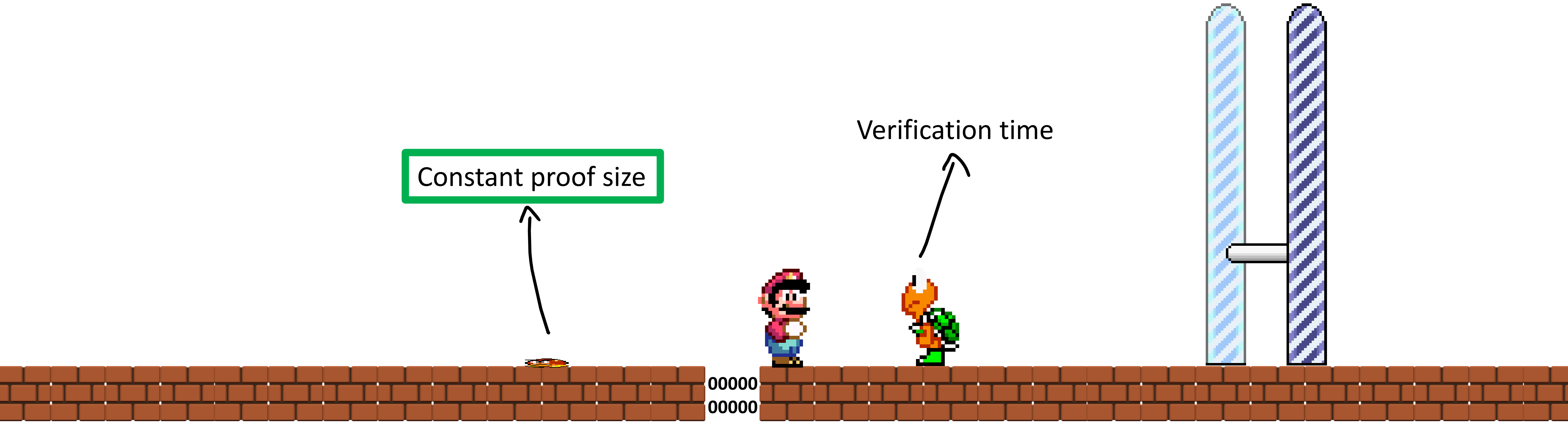
$$\sum_i d_i z_i = n \ \ mod\ \alpha \qquad \text{for } |n|, z \ll \alpha$$

Since all z_i are random and *independently* chosen from $\mathbb{Z}_p$,

If prover succeeds, can pick two satisfying assignments differing in one coordinate. $(r_0, r_1, \dots, r_l)$ and $(r_0', r_1, \dots, r_l)$

$$\Rightarrow \ d_i(r_0 - r_0') = (n - n') \ \ mod\ \alpha$$

$$\boxed{d_i = \frac{m_i \alpha - n_i}{k_i}} \qquad \text{where } m_i, n_i, k_i \ll \alpha$$



Constant proof size

Verification time

Verification time

All the z_i are independent and random

$\Rightarrow$ Takes linear time

$$\sigma := \sum_{j=0}^{\ell} \alpha^{2\ell-2j+1} z_j \longrightarrow \boldsymbol{z}(\alpha)$$

Choose $\boldsymbol{z} = \boldsymbol{x} \otimes \boldsymbol{y}$ for $\boldsymbol{x}, \boldsymbol{y} \in_R \mathbb{Z}_p^{\sqrt{\ell}}$, i.e., $z_{i\sqrt{\ell}+j} := x_i \cdot y_j$, for $0 \leq i, j \leq \sqrt{\ell}$.

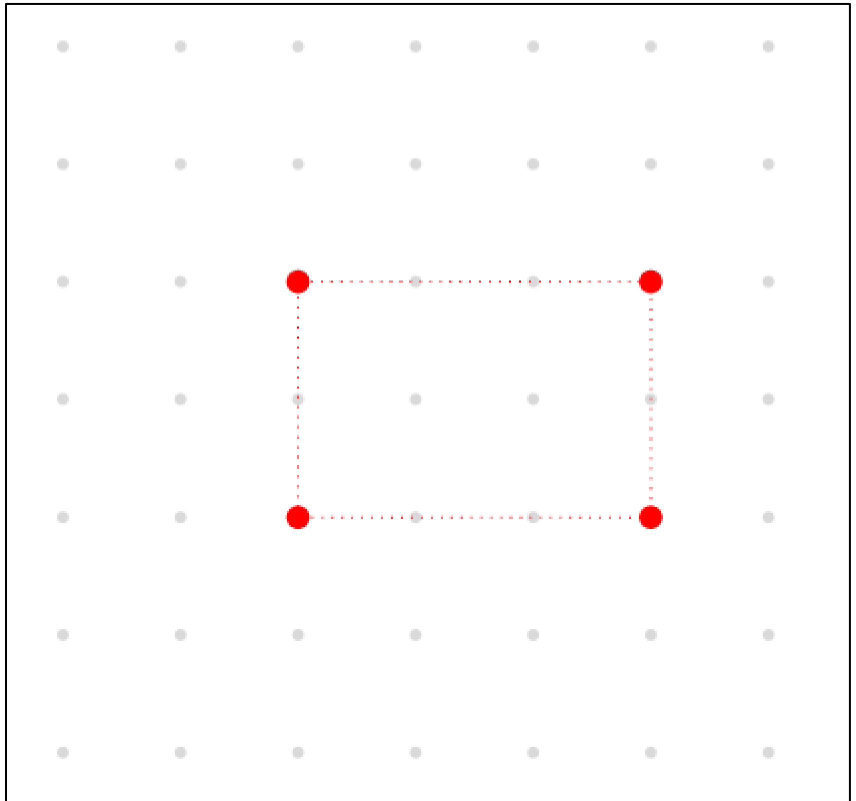
$$\sigma_{TEST} = \sum_k \alpha^{2\ell+1-2k} z^k = \alpha^{2\ell+1} \sum_{i,j} (\alpha^{-2})^{i\sqrt{\ell}+j} x_i y_j = \alpha^{2\ell+1} \cdot \left(\sum_i (\alpha^{-2\sqrt{\ell}})^i x_i \right) \cdot \left(\sum_j (\alpha^{-2})^j y_j \right)$$

Can be computed in $O(\sqrt{\ell})$ time.

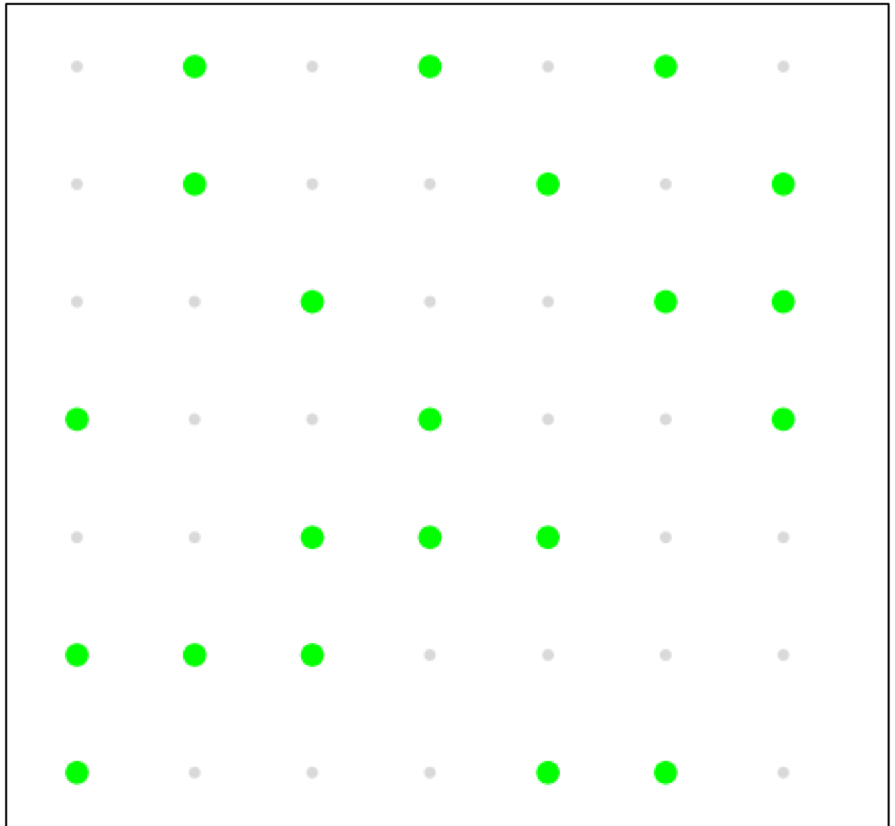
Soundness argument no longer works ☹

A question

Find the maximum number of points in an $n \times n$ grid that do *not* contain corners of a rectangle.



A Forbidden Configuration



NO Forbidden Configuration

Cancellation from rectangles

Answer: $\sim n\sqrt{n}$ points

In general, $\sim n^{d-2^{-d+1}}$ [Ros16]

Each coordinate $= i\sqrt{l} + j$ for some $i, j \leq \sqrt{l}$

Pick *four* accepting random choices of x_i, y_j such that they differ only in the i^{th} and j^{th} coordinates –

$$\underset{\boldsymbol{z_1}}{(x_i, y_j)}, \underset{\boldsymbol{z_2}}{(x_i, +h, y_j)}, \underset{\boldsymbol{z_3}}{(x_i, y_j + t)}, \underset{\boldsymbol{z_4}}{(x_i, +h, y_j + t)}$$

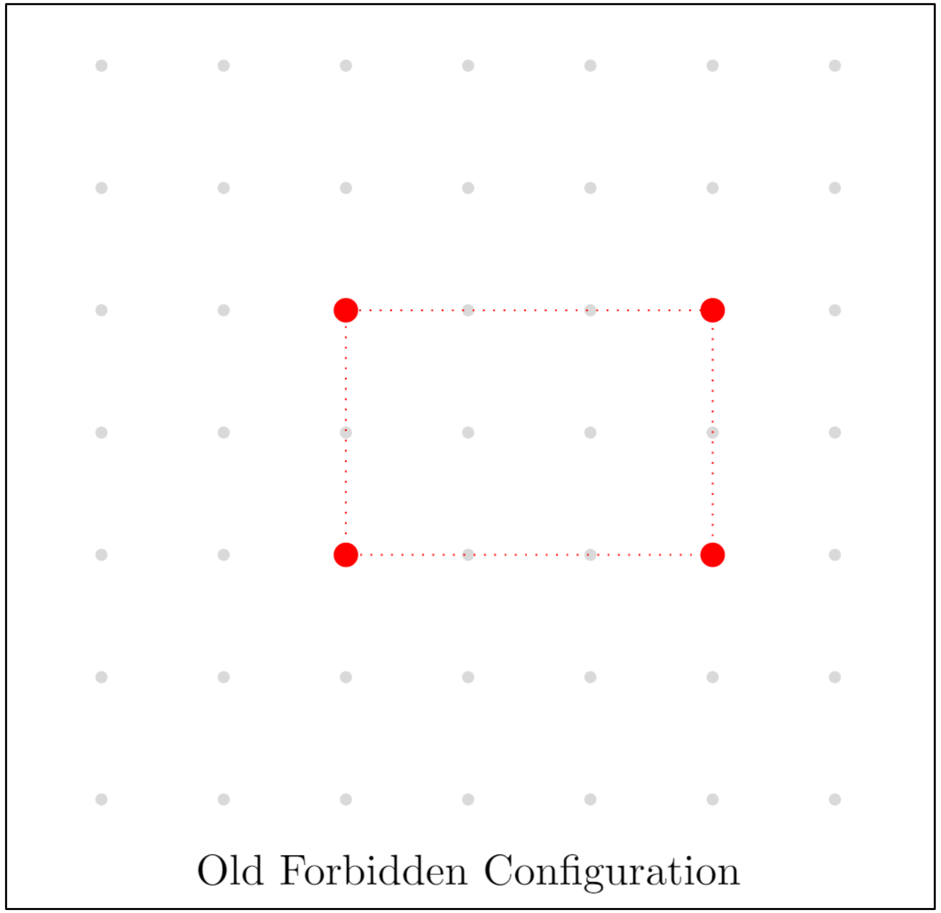
$$\sum_{i,j} d_{i,j} x_i y_j = n \ mod \ \alpha$$

Can isolate $d_{i,j}$ with four equations

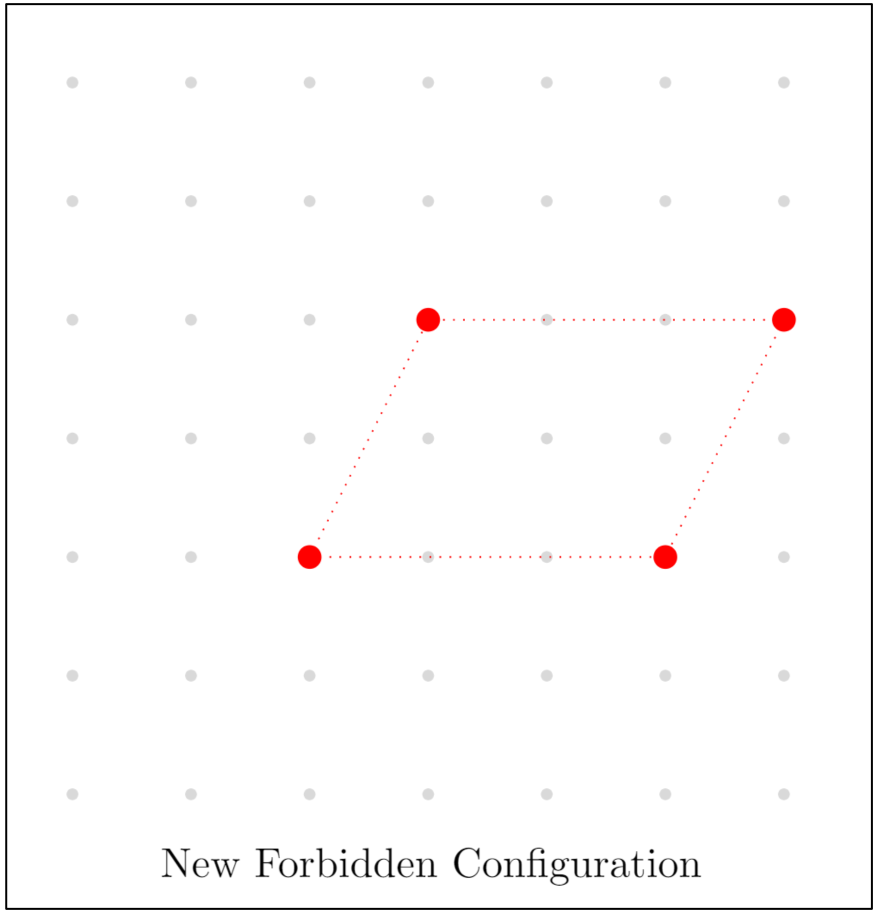
Soundness error $\sim \dfrac{\mathbf{1}}{\sqrt{\boldsymbol{n}}}$

For higher dim. $\sim \dfrac{1}{n^{2^{-d+1}}}$

Rectangles vs Parallelograms



Old Forbidden Configuration



New Forbidden Configuration

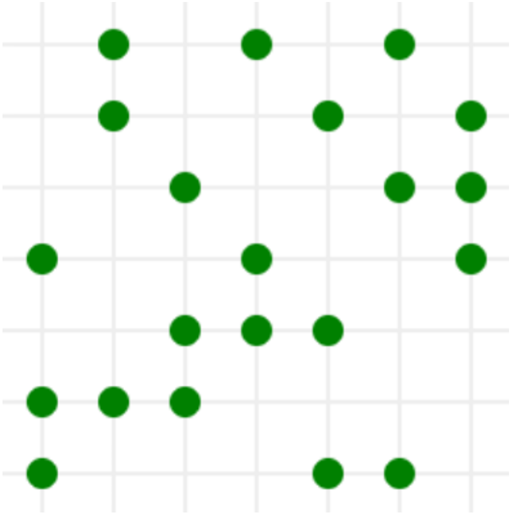
Are these easier to find?

Better bounds

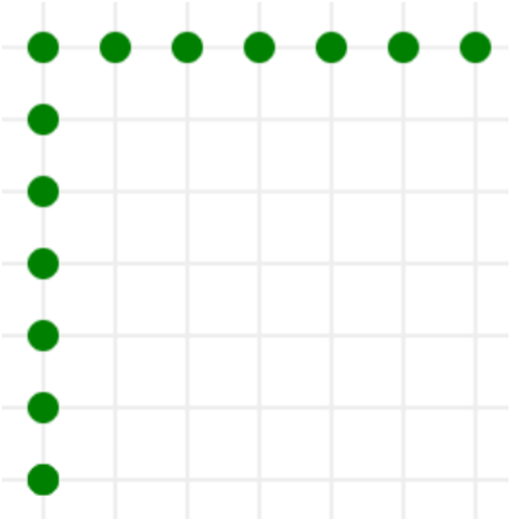
For dimension d, to find at least one

Box $\sim Cn^{d-2^{-d+1}}$ out of n^d points

d-cancellation structure $\sim dn^{d-1}$ out of n^d points



No rectangles



No parallelograms

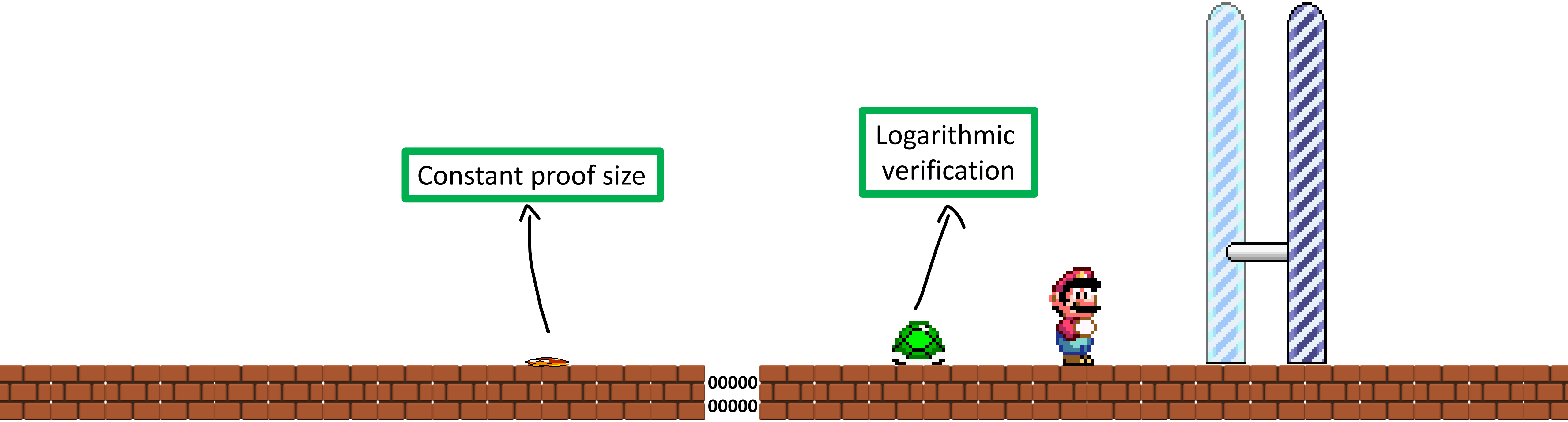
Logarithmic verification

Pick random $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_{\log l}$ from $\mathbb{Z}_p^2$ where $\boldsymbol{x}_j = (x_{j,0}, x_{j,1})$

Random query vector of the form

$$z_k \equiv z_{k_0,\ldots,k_{\log l-1}} := \prod_{j=1}^{\log l} x_{j,k_{j-1}}.$$

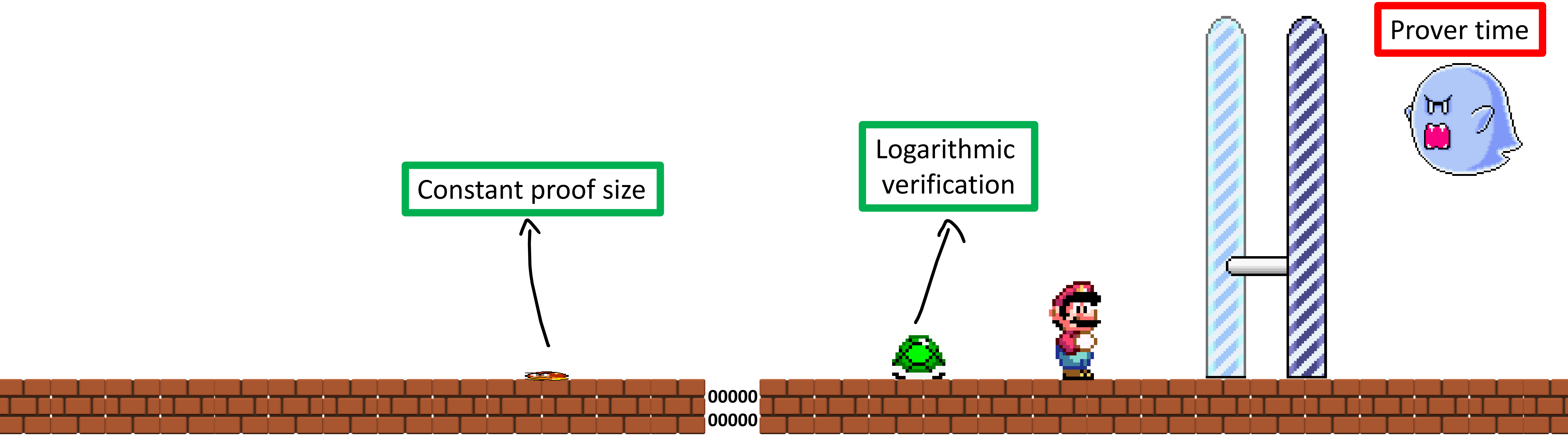
Soundness error $\sim \dfrac{\boldsymbol{log\ l}}{\boldsymbol{n}}$ = negl.



Constant proof size

Logarithmic verification

Open problems



Constant proof size

Logarithmic verification

Prover time

Thanks!

https://ia.cr/2022/419