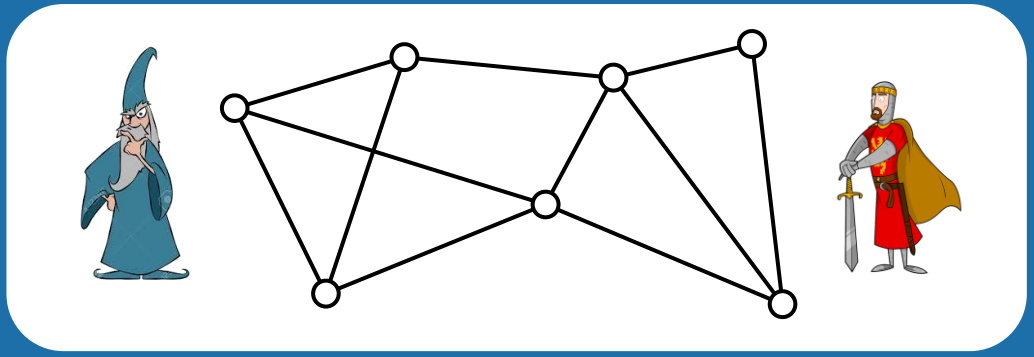
Zero-Knowledge Distributed Certification



Ami Paz

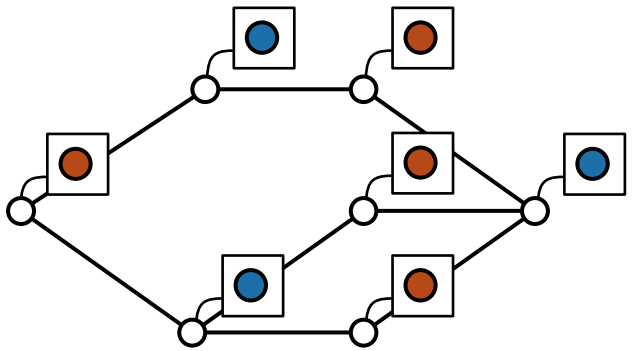
LISN: CNRS & Paris-Saclay University

Distributed Certification

- An n-node graph $G = (V, E)$ representing a network's topology

Goal:

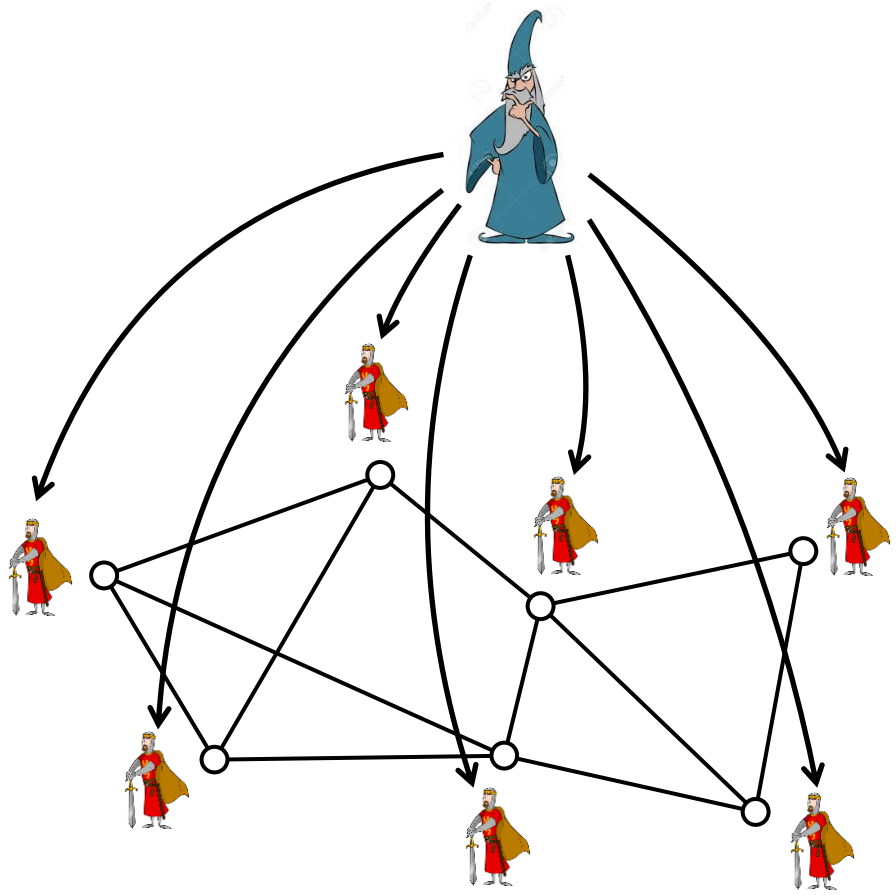
- Enable a network to verify its state
- Minimizing the interference to normal computation



Distributed Certification

1. Prover assigns certificates to nodes
2. Nodes communicate
3. Each node accepts/rejects

Repeat steps 2-3 periodically
to verify nothing “bad” happened

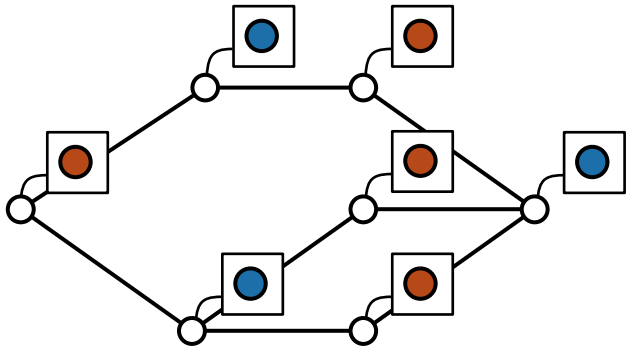


Distributed Certification

Given: Graph, predicate

Goal: Assign a label to each node such that:

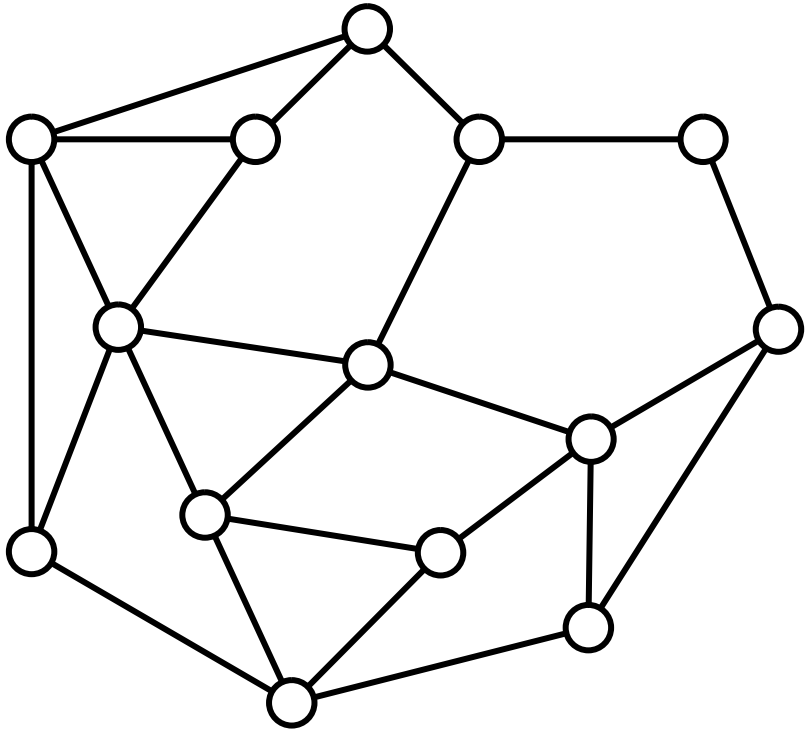
- Periodically, units exchange labels with neighbors
  - If ok, everyone continue as usual ✓ ✓ ✓ ✓
  - If there is a problem, someone detects it ✓ ✓ ✗ ✓



Goal: small labels

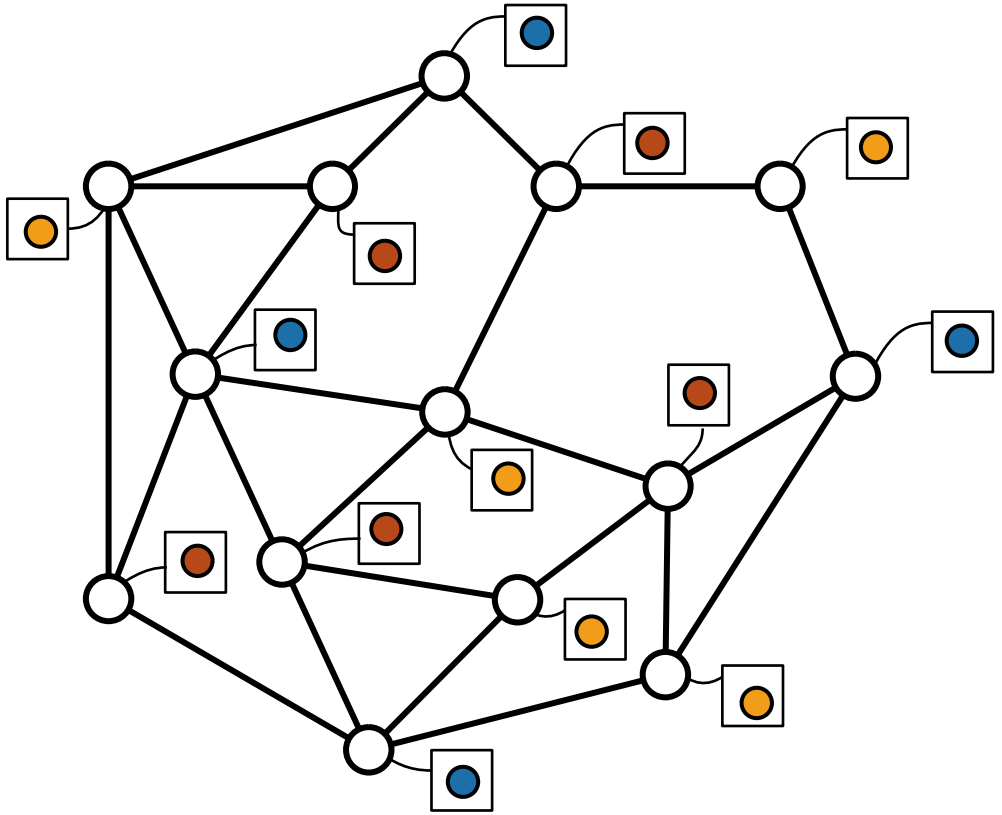
Example: c-colorability

- Predicate: the graph is c-colorable



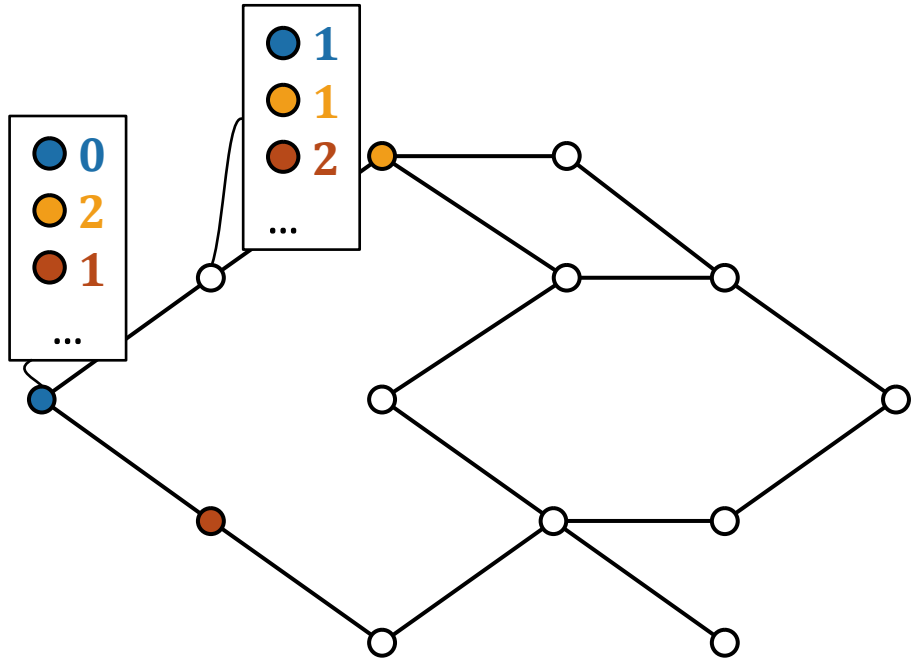
Example: c-colorability

- Predicate: the graph is c-colorable
- Labels: colors from $0, \ldots, c-1$



Example: Diameter

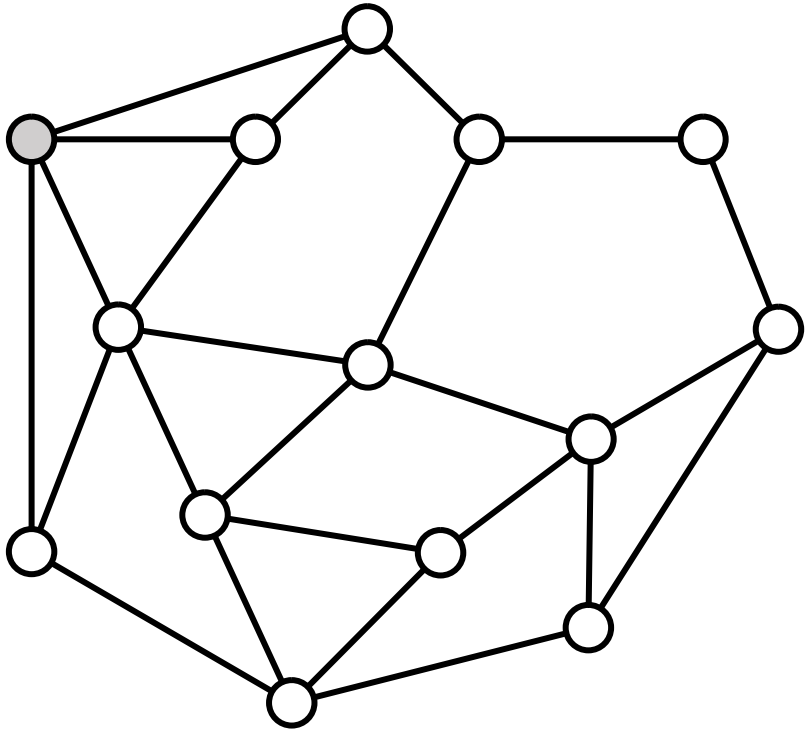
- $O(n \log n)$-bit certificates = all distances



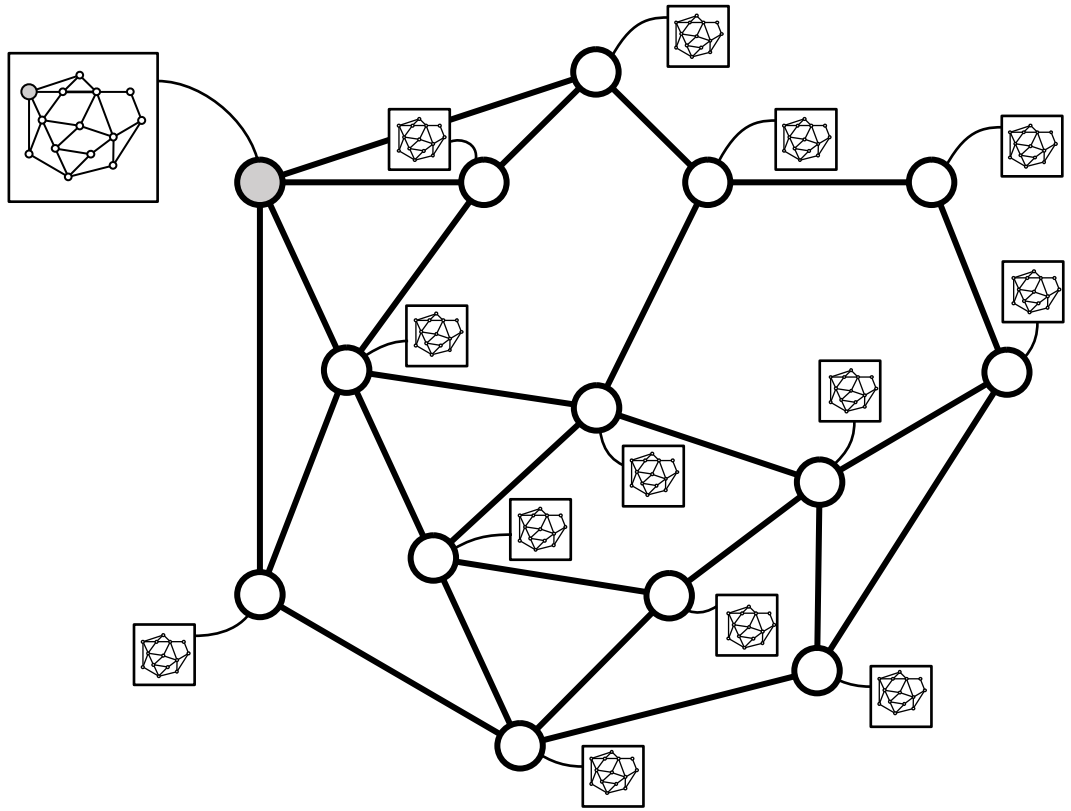
Certify **non** c-colorability

- Predicate: the graph is **not** c-colorable

?

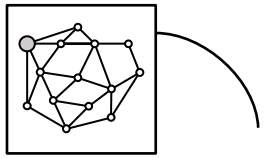


The Universal Scheme



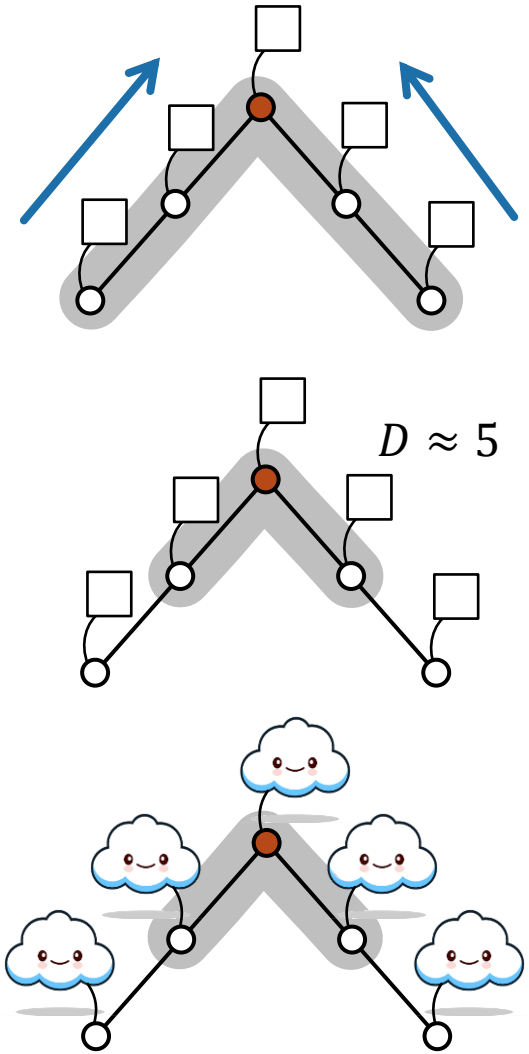
The Universal Scheme

- The universal scheme – All labels are the same
  - $O(n^2)$-*bit labels*
  - *Works for any predicate*
- Optimal for:
  - *Non 3-colorability*
  - *Symmetric graphs*
  - *…*



Variants

- Larger verification distance
- Approximate certification
- Quantum certification
- Interactive proofs
- **New**: Zero-Knowledge certification



Zero-Knowledge Proofs

Detour: Centralized Zero-Knowledge Proofs

Zero-Knowledge Proofs

Interactive proof system (P, V) for language L

- **Completeness**: $x \in L \Rightarrow$ honest prover convinces verifier
- **Soundness**: $x \notin L \Rightarrow$ cheating prover succeeds with negligible probability
- **Zero-knowledge**: verifier learns nothing beyond $x \in L$
  - Equivalent formulation: existence of an efficient **simulator**

ZK Protocol for 3-Colorability

1. Prover finds coloring $\mathbf{col}$ and applies random permutation π to colors
2. Prover commits to $\pi(\mathbf{col}(v))$ for every vertex v
3. Verifier selects random edge (u, v)
4. Prover opens commitments for u and v
5. Verifier checks opened colors differ

- Repeat $O(m)$ times for negligible soundness error

Example II: Graph Non-Isomorphism

Claim: graphs G_1 and G_2 are not isomorphic

Let's do something simpler...

Example II: Seeing Colors

- Two balls, red and green
- Colorblind verifier

Claim: Balls have different colors

1. Prover looks aside
2. Verifier does nothing / swaps the balls
3. Prover looks back, have to say if balls were switched

Distributed Zero-Knowledge Certification

Distributed Zero-Knowledge Certification

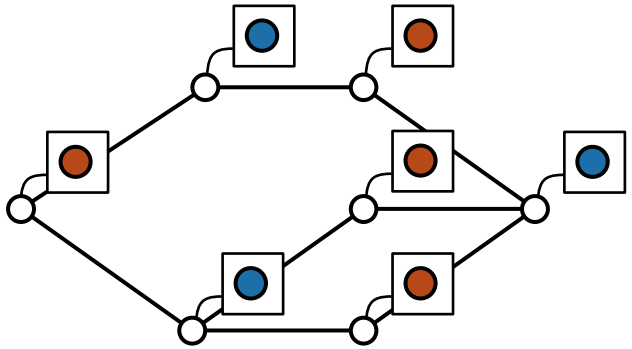
- Certify the network's states
- If the state is ok:
  - A node **does not learn anything** on network
  - E.g.: existence of edges between neighbors
- Extensions:
  - Coalitions of malicious nodes
  - A node is allowed to learn something it could learn by a local algorithm

Assumption: nodes have **shared randomness** (hidden from prover)

Example: 2-colorability

2-colorability – ZK by design

- Certificates: colors
- Verification: exchange colors
- **Perfect** completeness
- **Perfect** soundness
- **Perfect** ZK



“Distributed Zero-Knowledge Proofs Over Networks”

[Bick, Kol, Oshman 2022]

- Interactive proofs with zero-knowledge

$$\mathrm{dZK}[r, \ell, A, k]$$

Parameters:

- r: # rounds with prover
- ℓ: # bits with prover
- A: verifier/simulator power
- k: coalition size
- Weak/strong

Results

[Bick, Kol, Oshman 2022]

- Strong ZK for 3-Colorability

$$3-\mathrm{COL} \in \mathrm{dSZK}\left[3, O(\Delta), C\left(1, O(1)\right), 1\right]$$

- ZK for Spanning Tree Verification

$$\mathrm{STVer} \in \mathrm{dZK}[1, O(\log n), C(1, O(\log n)), 1]$$

- Generic compiler from PLS dZK, for $k = o(n)$

$$L \in \left[1, O\left(k^2(\Delta\ell + s)\log n\right), C\left(\tilde{O}(k), O\left(k^2(\Delta\ell + s)\log n\right)\right), k\right]$$

  - *For an ℓ-bit PLS with circuit size s*

“Distributed Non-Interactive Zero-Knowledge Proofs”

[Grilo, P., Perry 2026]

Non-interactive ZK proofs

$$\mathrm{dNIZK}[\ell, \mu] = \mathrm{dSZK}[r, \ell, A, k]$$

Parameters:

- $r = 1$: # rounds with prover
- ℓ: # bits with prover
- $A = CONGEST(1, \mu)$: verifier power
- $k = 1$: coalition size
- Always strong

Results

[Grilo, P., Perry 2026]

- Strong NIZK for 3-Colorability

$$3\text{−COL} \in \text{dNIZK}[\log n\,, \log n]$$

- Small subgraph freeness, $\forall 1 \leq \alpha \leq \sqrt{n}$

$$\Delta_{\text{free}}, \Diamond_{\text{free}} \in \text{dNIZK}\left[\frac{n}{\alpha}\log n\,, \alpha \log n\right]$$

- Generic compiler for any $L \in \text{NP}$, for any size of coalition

$$L \in [\text{ploy}\, n\,, \log n]$$

  - In the Random Oracle model

Interactive ZK Protocol for 3-Colorability

Why the Naïve Protocol Fails

Naïve protocol

- Prover sends u its color $\mathrm{col}(u) \in \{0,1,2\}$
- Neighbors exchange colors, check inequality

Problem: a node learns about it's neighbors (e.g., adjacency)

Interactive 3-Colorability Protocol

Theorem [Bick, Kol, Oshman 2022]

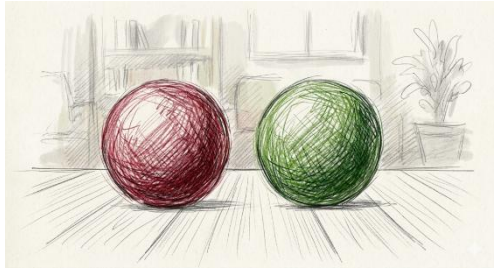
There exists a 3-round zero-knowledge protocol for 3-colorability using $O(\Delta)$-bit certificates and $O(1)$-bit messages

$$3\text{-col} \in \mathrm{dZKMAM}(\Delta, 1)$$

3-Round Protocol

- Prover assigns $\mathrm{col}(u)$ to each u
- Nodes need to verify $\mathrm{col}(u) \neq \mathrm{col}(v)$ for every edge (u, v)

Key idea: For every (u, v), prover proves it can **distinguish** $\mathrm{col}(u), \mathrm{col}(v)$



- Every two neighbors (u, v) sample $\boldsymbol{b_{u,v}} \in \{\mathbf{0}, \mathbf{1}\}$
- Two cases...

Protocol for an edge (u, v)

Case 1: $\boldsymbol{b_{u,v} = 0}$

- Nodes jointly sample a random permutation $\pi_{u,v} \in S_3$
- Send $\pi_{u,v}(\mathrm{col}(u))$ and $\pi_{u,v}(\mathrm{col}(v))$ (respectively) to the prover

If coloring is proper:

- Different colors: $\pi_{u,v}(\mathrm{col}(u)) \neq \pi_{u,v}(\mathrm{col}(v))$

Protocol for an edge (u, v)

Case 2: $\boldsymbol{b_{u,v} = 1}$

- Nodes jointly sample a random element $a_{u,v} \in \{0,1,2\}$
- Each sends $a_{u,v}$ to the prover

Now:

- Same color

Main Idea of the Protocol

- if $b_{u,v} = 0$: prover should receive **different colors**
- if $b_{u,v} = 1$: prover should receive **equal colors**

Prover must guess which case occurred

Soundness: If (u, v) violated, can only guess w.p. $1/2$

+ sequential repetition

Open Questions

[Bick, Kol, Oshman 2022]

- Prove that $\Omega(|E|)$ communication with prover is necessary
- Is interaction necessary?

Non-Interactive ZK Protocol for 3-Colorability

dNIZK for 3-Colorability

Theorem [Grilo, P., Perry 2026]

There exists a non-interactive zero-knowledge certification scheme for 3-colorability using $O(\log n)$-bit certificates and $O(\log n)$-bit messages

$$3\text{-col} \in \text{dNIZK}(\log n, \log n)$$

dNIZK 3-Colorability – Proof Idea

$$\text{3-col} \in \text{dNIZK}(\log n, \log n)$$

- Assigns $\text{col}(u) \in \{0,1,2\}$ to each node u
- Let C_u be an indicator polynomial of $\text{col}(u)$
  - *That is,* $C_u(i) = 1 \Leftrightarrow \text{col}(u) = i$
- **Key observation**: Edge (u, v) properly colored $\Leftrightarrow C_u(i)C_v(i) = 0 \ \forall i \in \{0,1,2\}$
- Sum to a single polynomial $P_u = \sum_{v \in N(u)} C_u C_v$
- Check $P_u(i) = 0 \ \forall i \in \{0,1,2\}$

ZK 3-Colorability Proof – Setting

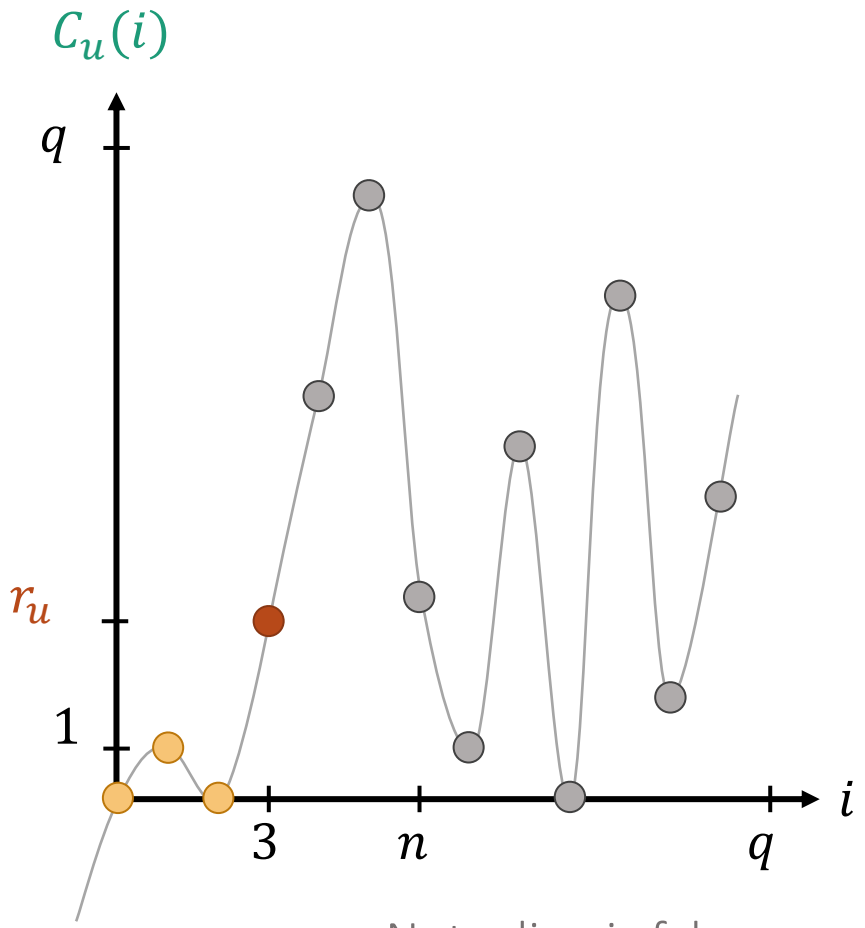
- Let $q \in [n+1, 2n]$ prime, known to all parties
  - $\mathbb{F}_q$ *the field of* q *elements*
- The prover chooses a random 3-coloring $\mathrm{col}: V \to \{0,1,2\}$
- Assigns $\mathrm{col}(u)$ to each node u
- Assigns a random $r_u \in \mathbb{F}_q$ to each node u

Defining C_u

For each u let $C_u : \mathbb{F}_q \to \mathbb{F}_q$ s.t.

- $C_u(i) = \begin{cases} 0 & \text{if } 0 \le i \le 2 \text{ and } i \neq \mathrm{col}(u) \\ 1 & \text{if } i = \mathrm{col}(u) \\ r_u & \text{if } i = 3 \end{cases}$
- degree $C_u \le 3$

Fig: C_u when $\mathrm{col}(u) = 1$



Note: line is fake

Defining P_u

- Let $P_u(i) = \sum_{v \in N(u)} C_u(i) C_v(i)$
  - *deg* $P_u \leq 6$
- Recall:

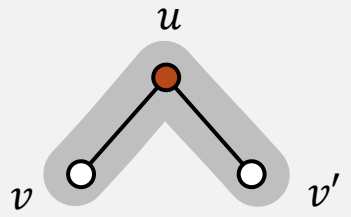
  $\text{col}(u) \neq \text{col}(v)\ \forall v \in N(u) \iff P_u(i) = 0\ \forall i \in \{0,1,2\}$
- u needs to verify this **without learning $\boldsymbol{P_u}$**

Detour: P_u Leaks Information

- Recall: $P_u(i) = \sum_{v \in N(u)} C_u(i) C_v(i)$

$$= C_u(i) \sum_{v \in N(u)} C_v(i)$$

- Using polynomial division, u learns $\sum_{v \in N(u)} C_v(i)$
- This is bad



- E.g., if $C_v(i) + C_{v'}(i) = 2$ for some i then $(v, v') \notin E$

Sharing P_u

- Split P_u among u and its neighbors:
  - Each v receives a random degree-6 polynomial H_v
  - u receives a degree-6 polynomial $P_v^{(0)}$ s.t.

$$P_u = P_u^{(0)} + \sum_{v \in N(u)} H_v$$

- To compute $P_u(i)$ for a given i:
  - Each neighbor $v \in N(u)$ sends $H_v(i)$ to u
  - u compute $P_u^{(0)}(i)$ and sums to get $P_u(i)$

So Far

- $\mathrm{col}: V \to \{0,1,2\}$, random value r_u
- C_u indicating $\mathrm{col}(u)$ on $i = 0,1,2$
- P_u checking col around u
- H_v random
- Sharing:

$$\sum_{v \in N(u)} C_u C_v = P_u = P_u^{(0)} + \sum_{v \in N(u)} H_v$$

What do we Need to Check?

1. Proper coloring: $P_u(i) = 0 \; \forall i = \{0,1,2\}$
2. Representation: P_u is indeed induced by col

$$\sum_{v \in N(u)} C_u C_v = P_u = P_u^{(0)} + \sum_{v \in N(u)} H_v$$

1. Checking Proper Coloring

For $i = 0,1,2$:

- Each neighbor $v \in N(u)$ sends $H_v(i)$ to v
- v compute $P_u^{(0)}(i)$ and verifies

$$P_u^{(0)}(i) + \sum_{v \in N(u)} H_v(i) \overset{?}{=} 0$$

- **ZK:** As H_v are random, $H_v(i)$ are random
  - *u gets random numbers summing to* 0

2. Checking Representation

Given

- $\mathrm{col}(u)$, $\{\mathrm{col}(v)\}_{v \in N(u)}$
- r_u, $\{r_v\}_{v \in N(u)}$
- $P_u^{(0)}$, $\{H_v\}_{v \in N(u)}$

} Define C_u, $\{C_v\}_{v \in N(u)}$ (by extrapolation)

Need to verify

$$\sum_{v \in N(u)} C_u C_v = P_u = P_u^{(0)} + \sum_{v \in N(u)} H_v$$

2. Checking Representation (cont.)

1. Each u uses $\mathrm{col}(u)$, r_u to extrapolate C_u; similarly for v
2. u and its neighbors choose $i^* \in \mathbb{F}_q \setminus \{0,1,2\}$ u.a.r.
3. Each $v \in N(u)$ sends $C_v(i^*)$ and $H_v(i^*)$ to u
4. u verifies

$$\sum_{v \in N(u)} C_u(i^*) C_v(i^*) \stackrel{?}{=} P_u^{(0)}(i^*) + \sum_{v \in N(u)} H_v(i^*)$$

- Soundness error: $\approx \frac{1}{q}$

2. Checking Representation (cont.)

1. Each u uses $\mathrm{col}(u)$, r_u to extrapolate C_u; similarly for v
2. u and its neighbors choose $i^* \in \mathbb{F}_q \backslash \{0,1,2\}$ u.a.r
3. Each $v \in N(u)$ sends $C_v(i^*)$ and $H_v(i^*)$ to u
4. u verifies

Zero-knowledge?

$$\sum_{v \in N(u)} C_u(i^*) C_v(i^*) \stackrel{?}{=} P_u^{(0)}(i^*) + \sum_{v \in N(u)} H_v(i^*)$$

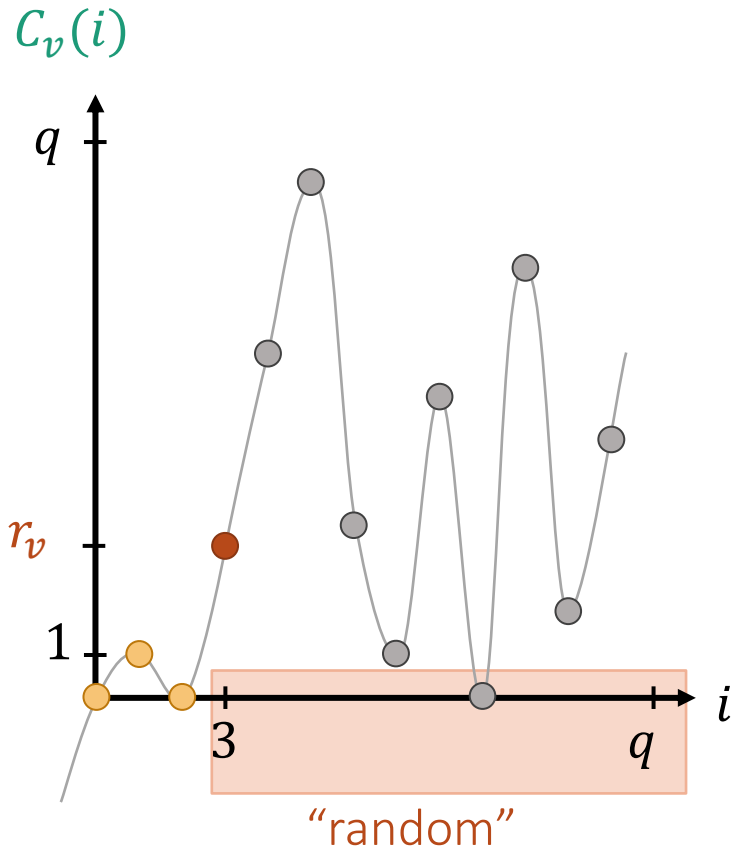
- Soundness error: $\approx \frac{1}{q}$

Zero Knowledge?

Each $v \in N(u)$ sends $C_v(i^*)$ and $H_v(i^*)$ to u

- Send $C_v(i^*)$?
  - Since $C_v(3) = r_v$ random, $C_v(i^*)$ random $\forall i^* \geq 3$
  - That is, independent of $\mathrm{col}(v)$
- Send $H_v(i^*)$?
  - H_v is a random polynomial

Fig: C_v when $\mathrm{col}(v) = 1$



Communication

- H_v is reused for all neighbors of v
  - *No dependency on* $|E|$*, only on* n
- Prover $\rightarrow u$
  - $\mathrm{col}(u)$, r_u, *6 coefficients of* P_u *and 6 of* $H_v(i)$
- $v \rightarrow u$
  - $H_v(i)$ *for* $i = 0,1,2, i^*$, $C_v(i^*)$

- In both cases, $O(1)$ elements of $\mathbb{F}_q$, i.e., $\boldsymbol{O(\log n)}$ **bits**

Conclusion & Extensions

- $3\text{-col} \in \mathrm{dNIZK}(\log n, \log n)$
- Easily extends to c-coloring and to maximum degree Δ:

$$c\text{-col} \in \mathrm{dNIZK}(c\log(c\Delta), c\log(c\Delta))$$

- Can use private randomness
  - *Needs 2 communication rounds among neighbors*
  - *Replace the global i^* by: Each u chooses i_u^* and sends to neighbors*
    - Works even with malicious parties

Zero-Knowledge Certification of Subgraph-Freeness

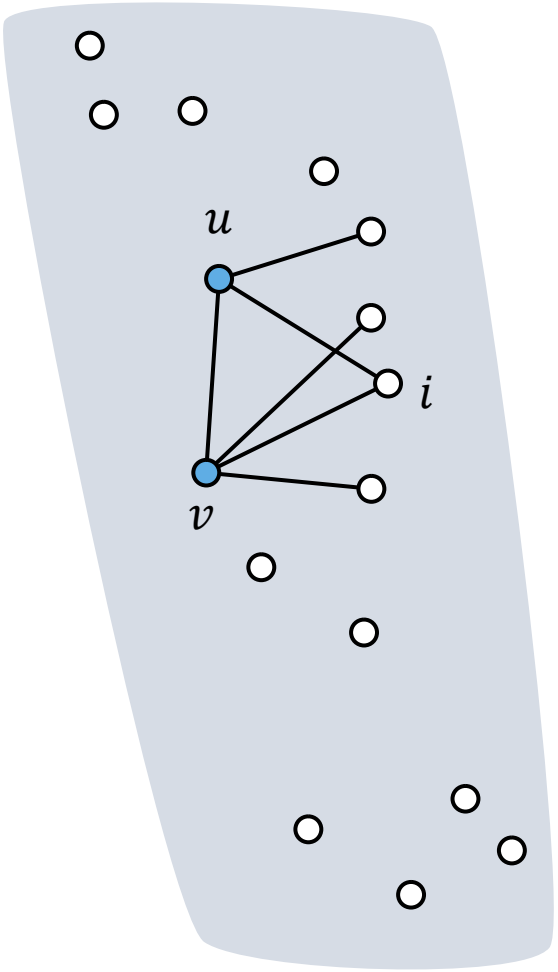
Triangle Freeness

Theorem

For every $\alpha \leq \sqrt{n}$, there exists a non-interactive zero-knowledge certification scheme for triangle-freeness with $\tilde{O}(n/\alpha)$-bit certificates and $\tilde{O}(\alpha)$-bit messages.

$$\forall \alpha \leq \sqrt{n}\colon\ \Delta_{\text{free}} \in \text{dNIZK}\left(\frac{n}{\alpha}\log n\,,\alpha \log n\right)$$

Triangle Freeness – Proof Idea



$\Delta_{\text{free}} \in \text{dNIZK}(n \log n, \log n)$

$V = [n]$, $q \in [n+1, 2n]$ prime

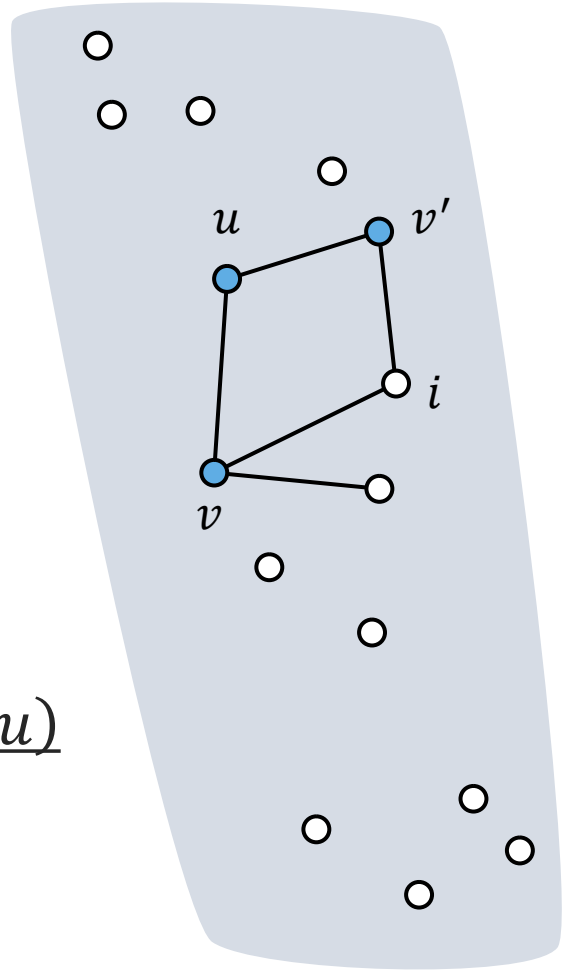
- Let C_u be an indicator polynomial of $N(u)$
  - *That is,* $C_u(i) = 1 \Leftrightarrow (u,i) \in E$
- Edge (u,v) not part of a triangle $\Leftrightarrow C_u(i)C_v(i) = 0 \; \forall i \in [n]$
- Sum to a single polynomial $P_u = \sum_{v \in N(u)} C_u C_v$
- Check $P_u(i) = 0 \; \forall i \in [n]$

C_4-Freeness

$\Diamond_{\text{free}} \in \text{dNIZK}(n \log n, \log n)$

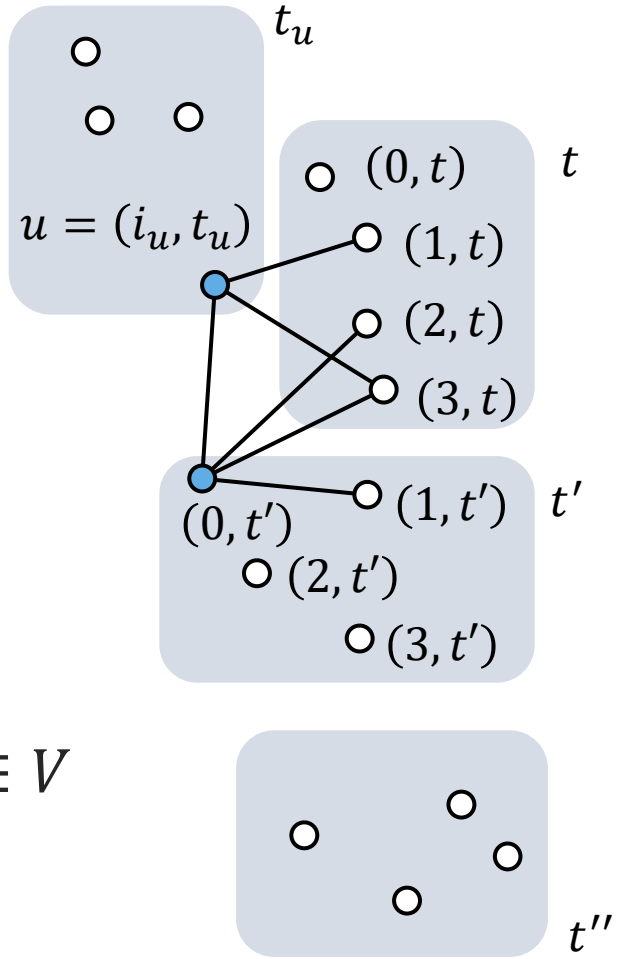
$V = [n]$, $q \in [n+1, 2n]$ prime

- Let C_u be an indicator polynomial of $N(u)$
- Node u not part of $C_4 \Leftrightarrow C_v(i)C_{v'}(i) = 0 \; \forall i \in [n], v, v' \in N(u)$
- Sum to a single polynomial $P_u = \sum_{v,v' \in N(u)} C_v C_{v'}$
- Check $P_u(i) = 0 \; \forall i \in [n]$



Tradeoff Results

$$\forall 1 \le \alpha \le \sqrt{n}\colon\ \Delta_{\text{free}}, \diamondsuit_{\text{free}} \in \text{dNIZK}\left(\frac{n}{\alpha}\log n\,, \alpha \log n\right)$$



Assume IDs in $[n/\alpha] \times [\alpha]$, $\alpha n < q \le 2\alpha n$

- For $t \in [\alpha]$ let $P_{u,t}$ be an indicator polynomial of $(i,t) \in N(u)$
- Edge (u,v) not part of a triangle $\Leftrightarrow C_{u,t}(i)C_{v,t}(i) = 0\ \forall (i,t) \in V$
- Sum to a single polynomial $P_u = \sum_{t\in[\alpha]} \sum_{v\in N(u)} P_{u,t}P_{v,t}$
- Check $P_u(i) = 0\ \forall i \in [n/\alpha]$

Universal Results

There are **universal compilers**

	Bick, Kol, Oshman 2022	Grilo, P., Perry 2026
Assumption	**None**	ROM or Cryptographic assumption
Type of distributed zero-knowledge	Weak	**Strong**
Max. size of coalition (k)	$k = o(n)$	$\boldsymbol{n}$
Communication with prover	1 round $O(k^2(\Delta\ell + s)\log n)$-bits	1 round $O(n^2 + \lvert\pi\rvert)$-bits
Communication with neighbors	$O(k)$ rounds $O(k^2(\Delta\ell + s)\log n)$-bits	**1 round** $\boldsymbol{O(\log n)}$**-bits**

Open Questions

- What problems have dNIZK with small certificates?
  - Logical characterization?
- Concrete problems
  - Colorability with more malicious parties
  - Planarity
- Lower bounds