Zero Knowledge for NP

APR 8
abhi shelat
3-COLORING OF A GRAPH

NP-COMPLETE
ZK-PROOF OF 3-COLORABILITY

Alice(G, C)  Bob(G)
ZK-PROOF OF 3-COLORABILITY

Alice(G, C)
PICK A COLOR PERM

Bob(G)
ZK-PROOF OF 3-COLORABILITY

\[ \text{Alice}(G, C) \]

- Pick a color perm
- Color the graph with new perm

\[ \text{Bob}(G) \]
**ZK-PROOF OF 3-COLORABILITY**

Alice($G, C$)

- Pick a color perm
- Color the graph
- Place cups over nodes

Bob($G$)
ZK-PROOF OF 3-COLORABILITY

Alice(G, C)

PICK A COLOR PERM

COLOR THE GRAPH
PLACE CUPS OVER NODES

Bob(G)
ZK-PROOF OF 3-COLORABILITY

Alice(\(G, C\))

- Pick a color perm
- Color the graph
- Place cups over nodes

Bob(\(G\))

- Pick a random edge
ZK-PROOF OF 3-COLORABILITY

Alice(G, C)

- Pick a color perm
- Color the graph
- Place cups over nodes
- Reveals chosen edge

Bob(G)

- Pick a random edge
**ZK-PROOF OF 3-COLORABILITY**

*Alice*($G, C$)

- Pick a color perm
- Color the graph
- Place cups over nodes
- Reveals chosen edge

*Bob*($G$)

- Pick a random edge
- Check colors
HOW TO MAKE THIS PROTOCOL INTERNET-READY?

WHAT FUNCTION DID THE CUPS PLAY?
HOW TO MAKE THIS PROTOCOL INTERNET-READY?

WHAT FUNCTION DID THE CUPS PLAY?

(binding) prover could not change color
(hiding) verifier could not see color
COMMITMENT SCHEMES

TWO PROTOCOLS:  
\text{\underline{commit}()}  \quad \text{\underline{open}()}
COMMITMENT SCHEMES

TWO PROTOCOLS: COMMIT() OPEN()

\[(c, s) \leftarrow \text{COMMIT}(\mathbf{m}, r)\]

Sender \hspace{1cm} Receiver
COMMITMENT SCHEMES

TWO PROTOCOLS: COMMIT() OPEN()

\((c, s) \leftarrow \text{COMMIT}(m; r)\)

Sender \[\rightarrow\] Receiver

\(\text{OPEN}_s(m, s)\)

Sender \[\rightarrow\] Receiver

\(\text{OPEN}_r(c, m, s) = m\)


**SECURITY PROPERTIES**

**computational**

**HIDE** FOR ALL \( m_0, m_1 \)

\[
\{ \text{commit}(m_0, r) \}_n \equiv \{ \text{commit}(m_1, r) \}_n
\]

**Perfectly**

**BIND** \( \exists (c^*, m_0, m_1, s_0, s_1) \) S.T. \( m_0 + m_1 \) and

\[
\text{open}(c^*, m_0, s_0) = m_0 \quad \text{and} \quad \text{open}(c^*, m_1, s_1) = m_1
\]
SECURITY PROPERTIES

(COMPUTATIONAL)

HIDE FOR ALL \(m_0, m_1\)

BIND
SECURITY PROPERTIES

(COMPUTATIONAL)
HIDE FOR ALL $m_0, m_1$

(PERFECT)
BIND
SECURITY PROPERTIES

HIDE FOR ALL $m_0$, $m_1$

BIND
SECURITY PROPERTIES

HIDE FOR ALL \( m_0, m_1 \)

\[ \text{COMMIT}(m_0, r) \quad \text{COMMIT}(m_1, r) \]

BIND
SECURITY PROPERTIES

HIDE

FOR ALL $m_0, m_1$

\[
\{ \text{COMMIT}(m_0,r) \} \approx \{ \text{COMMIT}(m_1,r) \}
\]

BIND
SECURITY PROPERTIES

HIDE

FOR ALL $m_0$, $m_1$

\[
\{ \text{COMMIT}(m_0, r) \} \approx \{ \text{COMMIT}(m_1, r) \}
\]

BIND

IMPOSSIBLE FOR ANY SENDER TO PRODUCE $c^*, m_0, s_0, m_1, s_1$ SUCH THAT $m_0 \neq m_1$ AND

\[
\text{OPEN}_R(c^*, m_0, s_0) = m_0 \\
\text{OPEN}_R(c^*, m_1, s_1) = m_1
\]
SECURITY PROPERTIES

(COMPUTATIONAL)

**HIDE**  FOR ALL $m_0$, $m_1$

\[
\{ \text{COMMIT}(m_0, r) \} \approx \{ \text{COMMIT}(m_1, r) \}
\]

**BIND**  IMPOSSIBLE FOR ANY SENDER TO PRODUCE $c^*, m_0, s_0, m_1, s_1$ SUCH THAT $m_0 \neq m_1$ AND

\[
\text{OPEN}_R(c^*, m_0, s_0) = m_0 \\
\text{OPEN}_R(c^*, m_1, s_1) = m_1
\]
SECURITY PROPERTIES

(COMPUTATIONAL)

HIDE

FOR ALL $m_0, m_1$

\[
\{ \text{COMMIT}(m_0, r) \} \approx \{ \text{COMMIT}(m_1, r) \}
\]

(PERFECT)

BIND

IMPOSSIBLE FOR ANY SENDER TO PRODUCE $c^*, m_0, s_0, m_1, s_1$ SUCH THAT $m_0 \neq m_1$ AND

\[
\text{OPEN}_R(c^*, m_0, s_0) = m_0 \\
\text{OPEN}_R(c^*, m_1, s_1) = m_1
\]
CAN BE IMPLEMENTED

WITH OWF

\[ f : \mathbb{Z}_p \to \mathbb{Z}_p \]

\[
\text{commit} (m) : \quad r \leftarrow \$
\]
\[
c \leftarrow (f(r), b(r) \oplus m)
\]
\[
s \leftarrow r
\]

\[
\text{open}(c, m, s) : \quad \text{check that } c = (f(s), b(s) \oplus m)
\]
\[
\text{if yes, output } m,
\]
\[
\text{else } \perp
\]


**ZK-PROOF OF 3-COLORABILITY**

\[ Alice(G, C) \]

1. Picks a color perm \( p \)
2. Commits to \( p(C) \cdot G \)
3. \( S_i, S_j \)

\[ Bob(G) \]

1. \( v_i, v_j \)
ZK-PROOF OF 3-COLORABILITY

$\text{Alice}(G, C)$

$\text{Bob}(G)$
ZK-PROOF OF 3-COLORABILITY

Alice(G, C)
PICK A COLOR PERM p

Bob(G)
Alice \((G, C)\)  
PICK A COLOR PERM \(p\)  
COLOR GRAPH WITH \(p(C)\)

Bob \((G)\)
ZK-PROOF OF 3-COLORABILITY

Alice(G, C)

PICK A COLOR PERM p
COLOR GRAPH WITH p(C)
COMMIT TO COLORS:

(c_i, s_i) ← COMMIT(p(C)(v_i); r_i)

Bob(G)
ZK-PROOF OF 3-COLORABILITY

**Alice**($G, C$)
- Pick a color perm $p$
- Color graph with $p(C)$
- Commit to colors:
  $$(c_i, s_i) \leftarrow \text{COMMIT}(p(C)(v_i); r_i)$$
- Send commitments $c_1, \ldots, c_m$

**Bob**($G$)
**ZK-PROOF OF 3-COLORABILITY**

\[ \textbf{Alice}(G, C) \]

- **PICK A COLOR PERM** \( p \)
- **COLOR GRAPH WITH** \( p(C) \)
- **COMMIT TO COLORS:**
  \[ (c_i, s_i) \leftarrow \text{COMMIT}(p(C)(v_i); r_i) \]
- **SEND COMMITMENTS**

\[ c_1, \ldots, c_m \]

\[ v_i, v_j \]

\[ \textbf{Bob}(G) \]

- **PICK RANDOM EDGE**

---

Diagram shows a graph with vertices colored in green and red, and arrows indicating the flow of commitments and communication between Alice and Bob.
ZK-PROOF OF 3-COLORABILITY

Alice\((G, C)\)

**PICK A COLOR PERM** \(p\)

**COLOR GRAPH WITH** \(p(C)\)

**COMMIT TO COLORS:**

\[(c_i, s_i) \leftarrow \text{COMMIT}(p(C)(v_i); r_i)\]

**SEND COMMITMENTS**

\[c_1, \ldots, c_m\]

**PICK RANDOM EDGE**

Bob\((G)\)

**OPEN EDGE**

\[s_i, s_j, p(C)(v_i), p(C)(v_j)\]
ZK-PROOF OF 3-COLORABILITY

Alice \( (G, C) \)

- Pick a color permutation \( p \)
- Color graph with \( p(C) \)
- Commit to colors:
  \[ (c_i, s_i) \leftarrow \text{COMMIT}(p(C)(v_i); r_i) \]
- Send commitments: \( c_1, \ldots, c_m \)

Bob \( (G) \)

- Pick random edge
- Open edge
- Check colors

\( s_i, s_j, p(C)(v_i), p(C)(v_j) \)
**ZK-PROOF OF 3-COLORABILITY**

\[ \text{Alice}(G, C) \]

\[ (c_i, s_i) \leftarrow \text{COMMIT}(p(C')(v_i); r_i) \]

**SEND COMMITMENTS**

\[ c_1, \ldots, c_m \]

**PICK RANDOM EDGE**

\[ v_i, v_j \]

**OPEN EDGE**

\[ s_i, s_j, p(C)(v_i), p(C')(v_j) \]

**Bob(G)**

**COMPLETENESS:**

\[ \text{Inspection.} \]
ZK-PROOF OF 3-COLORABILITY

\[ Alice(G, C) \]
\[ (c_i, s_i) \leftarrow \text{COMMIT}(p(C')(v_i); r_i) \]

SEND COMMITMENTS \[ c_1, \ldots, c_m \]

PICK RANDOM EDGES

OPEN EDGE \[ v_i, v_j \]

\[ s_i, s_j, p(C)(v_i), p(C')(v_j) \]

SOUNDNESS:

must rely on the perfect binding property

We must be sound against even unbounded P*
ZK-PROOF OF 3-COLORABILITY

SOUNDNESS: Consider first $P^*$ msg. By perfect binding, each commitment $c_i$ has at most one opening color. If $G \not\in 3\text{col}$, then $\exists$ a pair $c_i, c_j$ s.t. $\nexists$ $s_i, s_j$

$$\text{open}(c_i, s_i) = \text{open}(c_j, s_j). \quad (\text{Rest as before})$$

or INVALID
ZK-PROOF OF 3-COLORABILITY

Alice\((G, C)\)
\( (c_i, s_i) \leftarrow \text{COMMIT}(p(C)(v_i); r_i) \)

Bob\((G)\)

SEND COMMITMENTS
\( c_1, \ldots, c_m \)

PICK RANDOM EDGE
\( v_i, v_j \)

\( s_i, s_j, p(C)(v_i), p(C)(v_j) \)

ZERO-KNOWLEDGE: (must rely on the hiding property)
We must exhibit a simulator \( S \)
ZK-PROOF OF 3-COLORABILITY

\[ Alice(G, C) \]

\[ (c_i, s_i) \leftarrow \text{COMMIT}(p(C')(v_i); r_i) \]

SEND COMMITMENTS

\[ c_1, \ldots, c_m \]

\[ v_i, v_j \]

PICK RANDOM EDGE

OPEN EDGE

\[ s_i, s_j, p(C)(v_i), p(C')(v_j) \]

ZERO-KNOWLEDGE:
**ZK-PROOF OF 3-COLORABILITY**

\[ Alice(G, C) \]
\[ (c_i, s_i) \leftarrow \text{COMMIT}(p(C')(v_i); r_i) \]

**SEND COMMITMENTS**
\[ c_1, \ldots, c_m \]

**PICK RANDOM EDGE**
\[ v_i, v_j \]

\[ Bob(G) \]

**OPEN EDGE**
\[ s_i, s_j, p(C)(v_i), p(C)(v_j) \]

**REPEAT** \[ m^2 \] **TIMES**

**ZERO-KNOWLEDGE:**
ZK-PROOF OF 3-COLORABILITY

\[ Alice(G, C) \]
\[ (c_i, s_i) \leftarrow \text{COMMIT}(p(C'(v_i); r_i) \]

SEND COMMITMENTS

\[ c_1, \ldots, c_m \]

PICK RANDOM EDGE

\[ v_i, v_j \]

\[ s_i, s_j, p(C)(v_i), p(C')(v_j) \]

ZERO-KNOWLEDGE:

\[ x \in L \]
zk-proof of 3-colorability

$\text{Alice}(G, C) \downarrow (c_i, s_i) \leftarrow \text{COMMIT}(p(C')(v_i); r_i)$

SEND COMMITMENTS

$c_1, \ldots, c_m$

$\text{Bob}(G)$

PICK RANDOM EDGI

$v_i, v_j$

OPEN EDGE

$s_i, s_j, p(C)(v_i), p(C')(v_j)$

ZERO-KNOWLEDGE:

$x \in L$
ZK-PROOF OF 3-COLORABILITY

Alice\((G, C)\)

\((c_i, s_i) \leftarrow \text{COMMIT}(p(C')(v_i); r_i)\)

SEND COMMITMENTS

Bob\((G)\)

\(c_1, \ldots, c_m\)

PICK RANDOM EDGE

\(v_i, v_j\)

OPEN EDGE

REPEAT \(m^2\) TIMES

ZERO-KNOWLEDGE:

\(x \in L\)
ZK-PROOF OF 3-COLORABILITY

Alice \((G, C)\):
\[(c_i, s_i) \leftarrow \text{COMMIT}(p(C'(v_i); r_i)\]
SEND COMMITMENTS
\[c_1, \ldots, c_m\]
OPEN EDGE
\[v_i, v_j\]
REPEAT \(m^2\) TIMES

Bob \(G)\):
PICK RANDOM EDGI
\[s_i, s_j, p(C)(v_i), p(C)(v_j)\]

ZERO-KNOWLEDGE:

MUST EXHIBIT A P.P.T. SIMULATOR S.T.
FOR ALL VERIFIERS \(V^*\) AND FOR ALL
\[\text{View}_{V^*}(\langle P(x, w), V^*(x) \rangle) \approx \frac{x \in L}{S^{V^*}(x)}\]
ZERO-KNOWLEDGE

\[ S^{V^*}(G) \]

1. pick edge e & color w/diff. colors.

2. commit all vertices to RED except \( e = (v_i, v_j) \) which we commit to according to 1.
   call this msg \( C^* = (c_1, \ldots, c_{|V|}) \)

3. Feed \( V(C^*) \rightarrow (i_j, j_i) = e \)

4. If \( e^1 = e \), then open \( (s_i, s_j) \)
   Else Repeat \( \frac{n|E|}{n} \) times
   Else FAIL
ZERO-KNOWLEDGE

$S^{V^*}(G)$
ZERO-KNOWLEDGE

$S^{V^*}(G)$

PICK & COLOR A RANDOM EDGE $e = v_i, v_j$
ZERO-KNOWLEDGE

\[ S^{V^*}(G) \]

PICK & COLOR A RANDOM EDGE \( e = v_i, v_j \)
COLOR THE REMAINING NODES WITH 0
ZERO-KNOWLEDGE

$S^V\ast(G)$

1. **Pick & Color a Random Edge**: $e = v_i, v_j$
2. **Color the Remaining Nodes with 0**
3. **Commit to Graph**
   
   $$(c_i, s_i) \leftarrow \text{COM}(\text{color}_i; r_i)$$
ZERO-KNOWLEDGE

\[ S^{V^*}(G) \]

Pick & Color a random edge \( e = v_i, v_j \)
Color the remaining nodes with 0
Commit to graph
\[
(c_i, s_i) \leftarrow \text{COM}(\text{color}_i; r_i)
\]
Send \( V^* \) first msg
\[ S^{V^*}(G) \]

1. **Pick & Color a Random Edge** \( e = v_i, v_j \)
2. **Color the Remaining Nodes with** \( 0 \)
3. **Commit to Graph**
   
   \[(c_i, s_i) \leftarrow \text{COM}(\text{color}_i; r_i)\]
4. **Send \( V^* \) First MSG**
5. **If Challenge** \( c = e \) **Output View**
6. **Else Repeat Up to** \( N|E| \) **Times**
ZERO-KNOWLEDGE

\[ S^{V^*}(G) \]

- PICK & COLOR A RANDOM EDGE \( e = v_i, v_j \)
- COLOR THE REMAINING NODES WITH \( \emptyset \)
- COMMIT TO GRAPH
  \[(c_i, s_i) \leftarrow \text{COM}(\text{color}_i; r_i)\]
- SEND \( V^* \) FIRST MSG \( \rightarrow c \)
- IF CHALLENGE \( c = e \) OUTPUT VIEW
- ELSE REPEAT UP TO \( N|E| \) TIMES

ARGUE: ALGORITHM TERMINATES IN POLY TIME.
(COMPUTATIONAL)

HIDE FOR ALL $m_0, m_1$

\[
\left\{ \text{COMMIT}(m_0, r) \right\} \approx \left\{ \text{COMMIT}(m_1, r) \right\}
\]
INTUITION

$\langle P(G, C), V^*(G) \rangle$

$\text{COM}(c_1) \text{COM}(c_2) \cdots \text{COM}(c_i) \cdots \text{COM}(c_j) \cdots \text{COM}(c_v) \quad e \quad c_i, c_j$

$\text{COM}(0) \text{COM}(c_2) \cdots \text{COM}(c_i) \cdots \text{COM}(c_j) \cdots \text{COM}(c_v) \quad e \quad c_i, c_j$

$\text{COM}(0) \text{COM}(0) \cdots \text{COM}(c_i) \cdots \text{COM}(c_j) \cdots \text{COM}(c_v) \quad e \quad c_i, c_j$

$\text{COM}(0) \text{COM}(0) \cdots \text{COM}(c_i) \cdots \text{COM}(c_j) \cdots \text{COM}(0) \quad e \quad c_i, c_j$

$S^*(G)$
ZERO-KNOWLEDGE

Proof: \( \exists x \in L \) s.t. for inf. many \( n \), \( \exists \) a dist. \( D \) that distinguishes \( \{ \text{View}_{x}^{*}(P(x), U(x)) \} \) from \( \{ S^{u_{*}}(x) \} \).

\[ S \]

Note:

\[ \exists \{ S^{u_{*}}(x) \} = \{ S^{u_{*}}(x) \} \]

\( \text{STEP} \)

\( S \) \( \Rightarrow \) \( S' \) \( \Rightarrow \) \( S' \) same as \( S \) except it also receives the coloring \( C \), and it picks a random \( \pi \) & colors \( C_{c} = \pi(v_{i}) \), \( C_{j} = \pi(v_{j}) \).
ZERO-KNOWLEDGE

\( \text{VIEW}_{\nu} (p(x) \oplus V^*(x)) \quad S''(x) \)

same as \( S \) except:
1. \( \nu \) also receives coloring \( C \)
2. \( \nu \) commits to \( p(C) \) random coloring
3. \( \nu \) also picks \( e \) like \( S \)
4. \( \nu \) rewinds if \( e' \neq e \) (just like \( S \))

Lemma: \( \textbf{\{VIEW}_{\nu} (p(x) \oplus V^*(x)) \} \approx_s \{ S''(x) \} \)

They only differ when \( S''(x) \) fails after \( n |E| \) times.

This only happens w/negligible probability.
ZERO-KNOWLEDGE

D distinguishes these 2

\( \frac{1}{2^{\text{poly}(n)}} \)

S’’

S’

S’

S’

S’

S’

S’

S’

S’

Si = runs S’ for the first i iterations,

Δ S’’ for the rest.

By hybrid lemma, exists a pair i, i+1 s.t. D distinguishes

\( S'k \) and \( S'k+1 \) w/probability \( \frac{1}{2^{(\text{poly}(n)) \cdot n \cdot \text{HE}}} \)
ZERO-KNOWLEDGE

Define \( G \) \( E \) \( \text{more} \) hybrids

\( S_{k+1}' \)

\( S_{k+1} \) : if \( (i, j, \tau) \) that are chosen are \( \leq \varepsilon \), then run \( S_{k+1}' \). Else run \( S_{k+1} \).

ORDER \( (i, j, \tau) \) tuples.

\( U \rightarrow \tau \)

\( |E| \rightarrow 6 \) permutations

\( \text{Com}(0) \ldots \text{Com}(v_i) \quad \text{Com}(v_j) \ldots \text{Com}(0) \)

(\text{FULL colors})
Why does S get the coloring??

\( \chi(u) \) can be given as non-uniform advice to break the commitment scheme.
ROUND COMPLEXITY OF ZK

ONE ITERATION
ROUND COMPLEXITY OF ZK

ONE ITERATION

REPEAT k TIMES

$\text{n \mid t}$

$3 \times n \mid t \mid \text{G}$
ROUND COMPLEXITY OF ZK

ONE ITERATION

REPEAT k TIMES

O(k) ROUNDS!
CONSTANT ROUND ZK?

ONE ITERATION