Zero Knowledge Round Complexity

abhi shelat
ZK-PROOF OF 3-COLORABILITY

\[ Alice(G, C) \quad \text{commit} \quad \text{edge} \quad \text{reveal edge} \quad Bob(G) \]

Soundness: \( \frac{1}{m} \Rightarrow m^2 \) repeating
ZK-PROOF OF 3-COLORABILITY

\[ Alice(G, C) \]

PICK A COLOR PERM \( p \)

\[ Bob(G) \]

Diagram:

- Alice and Bob engage in a protocol to prove the 3-colorability of a graph without revealing the coloring.
- The graph \( G \) is a set of vertices connected by edges.
- Alice picks a coloring \( C \) and a permutation \( p \) of the colors.
- Bob verifies the coloring without seeing the permutation.

The protocol involves exchanging messages to ensure both parties are satisfied with the proof of colorability.
**ZK-PROOF OF 3-COLORABILITY**

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

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

Alice$(G, C)$

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

Bob$(G)$

SEND COMMITMENTS

$c_1, \ldots, c_m$
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)
- Pick random edge

\( v_i, v_j \)
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)

- Pick random edge

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 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)**

**PICK RANDOM EDGE**

$v_i, v_j$

**OPEN EDGE**

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

**CHECK COLORS**
ROUND COMPLEXITY OF ZK

ONE ITERATION \{ \rightarrow \rightarrow \rightarrow \} \ 3
ROUND COMPLEXITY OF ZK

\[ O(k) \]

\( k \) times

\[ 2^{-k} \]

Goal: \( O(1) \) round ZK proofs with negligible soundness for NP.
ROUND COMPLEXITY OF ZK

ONE ITERATION

REPEAT k TIMES

O(k) rounds!
CONSTANT ROUND ZK?

ONE ITERATION
CONSTANT ROUND ZK?

ONE ITERATION

REPEAT $k$ TIMES IN PARALLEL
E.G. FOR GRAPH COLORING
E.G. FOR GRAPH COLORING
E.G. FOR GRAPH COLORING
GOOD PROTOCOL?

REPEAT $k$ TIMES IN PARALLEL

COMPLETENESS? ✓
GOOD PROTOCOL?

REPEAT $k$ TIMES IN PARALLEL

COMPLETENESS? ✓

SOUNDNESS? same as before. $\frac{1}{n^2}$ per instance.
GOOD PROTOCOL?

REPEAT \( k \) TIMES IN PARALLEL

COMPLETENESS? ✓

SOUNDNESS? \( 2^{-k} \)
GOOD PROTOCOL?

REPEAT $k$ TIMES IN PARALLEL

COMPLETENESS? ✓
SOUNDNESS? $2^{-k}$
ZERO KNOWLEDGE? $\Rightarrow$ We must exhibit a simulater.
OLD SIMULATOR?

$S^{V*}(G)$

PICK & COLOR A RANDOM EDGE $e = v_i, v_j$ $k$ TIMES IN PARALLEL

COLOR THE REMAINING NODES WITH 0

COMMIT TO GRAPH

$$(c_i, s_i) \leftarrow \text{COM}(color_i, r_i)$$

SEND $V^*$ FIRST MSG

IF CHALLENGE $c = e$ FOR ALL $k$ INSTANCES OUTPUT VIEW
ELSE REPEAT
OLD SIMULATOR?

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

PICK & COLOR A RANDOM EDGE $e = v_i, v_j$ $k$ TIMES IN PARALLEL

COLOR THE REMAINING NODES WITH 0

COMMIT TO GRAPH

$(c_i, s_i) \leftarrow \text{COM}(color_i; r_i)$

SEND $V^*$ FIRST MSG

IF CHALLENGE $c = e$ FOR ALL $k$ INSTANCES OUTPUT VIEW
ELSE REPEAT
OLD SIMULATOR?

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

PICK & COLOR A RANDOM EDGE \( e = v_i, v_j \) \( k \) TIMES IN PARALLEL

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

IF CHALLENGE \( c = e \) FOR ALL \( k \) INSTANCES OUTPUT VIEW
ELSE REPEAT

\[ \frac{1}{m^k} \]

PR SUCCESS?

Simulator runs in exponential time.

We do not know of a simulator that works!
NEW IDEA

\[
\text{commit (each edge # plans query)}
\]

\[
\text{open commitment}
\]

Answer the query
NEW IDEA

COMMIT TO ALL $k$ CHALLENGES
NEW IDEA

COMMIT TO ALL K CHALLENGES

What type of commitment is:

- perfectly hiding
- perfectly binding

b/c $A^*$ could be unbounded.
NEW IDEA

COMMIT TO ALL K CHALLENGES
NEW IDEA

COMMIT TO ALL K CHALLENGES

ANSWER IF COMMITMENTS OPENED PROPERLY
SOUNDNESS?

COMMIT TO ALL $k$ CHALLENGES
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

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

“IF A STATEMENT IS FALSE, THEN NO PROVER SHOULD EVER BE ABLE TO CONVINCE THE VERIFIER THAT IT IS SO.”

FOR A LANGUAGE L, AND FOR ALL $x \not\in L \ \forall P^*$

$$\Pr[ \langle P^*(x), V(x) \rangle = 1 ] < \epsilon(|x|)$$

WHERE $\epsilon$ IS A NEGligible FUNCTION
SOUNDNESS?

NEEDS TO BE A PERFECTLY HIDING COMMITMENT!

COMMIT TO ALL K CHALLENGES

CHALLENGE OPEN COM

CHALLENGE OPEN COM

CHALLENGE OPEN COM

CHALLENGE OPEN COM

CHALLENGE OPEN COM
ZERO KNOWLEDGE?

$S^V_*(G)$

1. Receive first message

2.
ZERO KNOWLEDGE?

$S^V_*(G)$

RUN $V^*$

COMMIT TO ALL $k$ CHALLENGES.

Send bogus msg

Send good msg using challenges

Open this challenge

Answer properly
ZERO KNOWLEDGE?

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

RUN $V^*$

SEND MSG

COMMIT TO ALL $k$ CHALLENGES
ZERO KNOWLEDGE?

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

**RUN $V^*$**

**SEND MSG**

**GET CHAL**

- COMMIT TO ALL $k$ CHALLENGES
- CHALLE OPEN
- CHALLE OPEN
- CHALLE OPEN
- CHALLE OPEN
- CHALLE OPEN
ZERO KNOWLEDGE?

$S^V^* (G')$

RUN $V^*$

COMMIT TO ALL $k$ CHALLENGES

SEND MSG

STOP! REWIND
ZER0 KNOWLEDGE?

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

RUN \( V^* \)

COMMIT TO ALL \( k \) CHALLENGES

SEND MSG

SEND NEW MSG

STOP! REWIND
**ZERO KNOWLEDGE?**

$S^V (G)$

**RUN** $V^*$

**COMMIT TO ALL $k$ CHALLENGES**

**SEND MSG**

**SEND NEW MSG**

**GET CHAL**

**STOP!REWIND**
ZERO KNOWLEDGE?

$S^V(G)$

RUN $V^*$

COMMIT TO ALL K CHALLENGES

SEND MSG
SEND NEW MSG

GET CHAL

CHALLE OPEN

CHALLE OPEN

CHALLE OPEN

CHALLE OPEN

ANSWER
PROBLEM

$S^V^* (G)$

RUN $V^*$

COMMIT TO ALL $k$ CHALLENGES

SEND MSG

SEND NEW MSG

GET CHAL

OPEN

STOP! REWIND

VERIFIER REFUSES TO OPEN THIS COMMIT 2ND TIME.
SOLUTION: 17 PAGE ANALYSIS

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

RUN $V^*$

COMMIT TO ALL $k$ CHALLENGES

$z$ rounds. (perfectly hiding)

SEND MSG
SEND NEW MSG

GET CHAL

STOP! REWIND

VERIFIER REFUSES TO OPEN THIS COMMIT 2ND TIME.
ZERO-KNOWLEDGE ENCRYPTION

(Definition)

\[ \exists \text{ simulator } S \text{ s.t. } \forall m \in \{0,1\}^n \]

\[ \{ \text{pk}, \text{sk} \leftarrow \text{Gen}(1^n) : \text{pk}, \text{Enc}(\text{pk}, m) \} \approx \]

\[ \{ S(1^n) \} \]

1) make a pk
2) ENC(pk, 0)
NEW PROBLEM

Alice

Eve

Bob
NEW PROBLEM

Alice

Eve

m

Bob
NEW PROBLEM

Alice

Eve

Bob

DID Alice REALLY SEND ME $m$?

Authentication
ПРЕЗИДЕНТ
РЕСПУБЛИКИ АБХАЗИЯ

В. АРДЗИБА

СПИКЕР
НАРОДНОГО СОБРАНИЯ
РЕСПУБЛИКИ АБХАЗИЯ

С. ДЖИФИДЖОЛЯ

ДЕПУТАТЫ НАРОДНОГО СОБРАНИЯ
РЕСПУБЛИКИ АБХАЗИЯ:
TWO VERSIONS OF DIGITAL SIGNATURES
MESSAGE AUTHENTICATION CODE

private key model

Alice

Eve $SK_e$

Bob $SK_e$

$SK_A$

$SK_A$
MESSAGE AUTHENTICATION CODE

Alice → Eve → Bob

Eve intercepts the communication between Alice and Bob.

Alice and Bob share a secret key k with a trusted third party, Gen.

The diagram illustrates the potential attack by Eve on the message authentication code.
MESSAGE AUTHENTICATION CODE

t=Tag_k(m)
MESSAGE AUTHENTICATION CODE

Alice

m,t

k

Bob

Eve

$t=\text{Tag}_k(m)$
MESSAGE AUTHENTICATION CODE

\[ t = \text{Tag}_k(m) \]

\[ \text{Ver}_k(m,t) \]
Alice really did send it.
MAC

MESSAGE SPACE \(\{M\}_n\)

\(\text{Gen}(1^n)\)

\(\text{Tag}_k(m)\)

\(m \in M_n\)

\(\text{Ver}_k(m, t)\)
MAC

MESSAGE SPACE \( \{M\}_n \)

\textbf{Gen}(1^n) \quad \text{GENERATES A KEY } k

\textbf{Tag}_k(m) \quad \text{ } \quad m \in M_n

\textbf{Ver}_k(m,t)
MAC

MESSAGE SPACE $\{\mathcal{M}\}_{n}$

$\text{Gen}(1^n)$ generates a key $k$

$\text{Tag}_k(m)$ generates a tag $t$ for $m \in \mathcal{M}_n$

$\text{Ver}_k(m,t)$
MAC

MESSAGE SPACE $\{\mathcal{M}\}_n$

$\text{Gen}(1^n)$ GENERATES A KEY $k$

$\text{Tag}_k(m)$ GENERATES A TAG $t$ FOR $m \in \mathcal{M}_n$

$\text{Ver}_k(m,t)$ ACCEPTS OR REJECTS A MSG, TAG
MAC

MESSAGE SPACE $\{\mathcal{M}\}_n$

$\text{Gen}(1^n)$ GENERATES A KEY $k$

$\text{Tag}_k(m)$ GENERATES A TAG $t$ FOR $m \in \mathcal{M}_n$

$\text{Ver}_k(m,t)$ ACCEPTS OR REJECTS A MSG, TAG

$\Pr[k \leftarrow \text{Gen}(1^n) : \text{Ver}_k(m, \text{Tag}_k(m)) = 1] = 1$