Computational Zero-Knowledge:

$$S^{(V^*)}(x) \overset{c}{\cong} [P \leftrightarrow V(x)]$$

View of the two are the same as msg of P and V^* with V^*'s randomness.

1 Commitment Protocol

Commitment Protocol: An interactive protocol that consists of the commitment phase and open phase satisfying two properties:

- Binding: After the commit protocol terminiates, the commiter can only open b in 1 way.
- Hiding: After the commit phases is over, any PPT receiver cannot predict bit better than $\frac{1}{2} + \text{negl}$

2 Graph 3-Coloring

Given a graph G, we say it is **3-colorable** if we can assign colors to each node such that for all $(u, v) \in E$, u and v are different colors.

This problem is NP-complete. The problem of Circuit satisfiability is NP-Complete. Using 3-coloring, we can create gadgets that represent Or, And, and Not gates. Since these gates are turing complete, we can convert any circuit into a corresponding graph, such that one color represents 0, another represents 1, and the third color is used to assist with the gates. This way we know if the graph is 3-colorable, then the circuit it represents is satisfiable.

2.1 ZK Proof for 3-Color

Given a graph G, we have the following protocol to demonstrate that $G \in$ 3-Color:

1. The prover P creates a random permutation shuffling the colors from one to another randomly. (exp. Red → Blue). The prover secretly commits to this new three coloring and sends commitments to V.
2. V selects a random edge $e = (u, v)$ and sends their selection back to P.
3. P decommits nodes u and v to V who can see that they are different colors.

4. This process is repeated k times.

Completeness: is clear from the inspection.
Soundness: Consider some $G \notin$ **3-Color**. Then, $\exists e = (u, v) \in E$ where (u, v) are the same color. Since V selects a random edge e', the probability that they choose this edge is $\frac{1}{|E|}$. Therefore the probability that P can successfully fool the verifier is:

$$(1 - \frac{1}{|E|})^k$$

Which for $k = |E|$ is $1/e^{|E|}$.

2.2 Simulator for Protocol

To demonstrate this protocol is Zero Knowledge, we will create simulator S which does the following:

1. The simulator will pick a random edge e and commit two different colors to the edge's nodes.

2. If the verifier asks to reveal any edge that is not e, rewind the tape and try again.

3. Repeat until verifier accepts, since each attempt has a $1/|E|$ chance of succeeding, it will take approximately $k \cdot |E|$ tries to succeed.

3 Blum's Protocol for Hamiltonian Cycle

Hamiltonian Cycle is a decision problem that asks if there exists a simple cycle of length n inside of graph G. This problem is NP-Complete as one can map an instance of 3-Sat to an instance of this problem. The following protocol is a ZK proof for verifying if $G \in$ Ham-Cylce:

1. Prover P picks a random permutation π and commits an adjacency matrix of $\pi(G)$ to V.

2. V selects a random $b \overset{\$}{\leftarrow} \{0, 1\}$ and sends b to P.

3. If P obtains 0, it opens the entire adjacency matrix and sends it to V, along with the permutation of π^{-1} that maps the adjacency matrix it received back to G. If P obtains 1, it decommits the permuted cycle to V.

4. V checks P's decommitments and verifies that if $b = 0$, it properly permutes back to G, and if $b = 1$, P correctly decommited a single 1 in every row and column.

Notes on Protocol:
An adjacency matrix is a 0/1 matrix that represents an outgoing edge from vertex i to vertex j, which would result in something like $adj[i][j] = 1$. Since a hamiltonian cycle visits each vertex in the graph exactly once other than the starting vertex, there are n edges that exist, each starting at a unique source vertex and ending at a unique destination vertex. Therefore, if there exists a hamiltonian cycle in a graph, the adjaceny matrix can illustrate this by presenting n entries of 1's such that each is the only 1 in its column and row.

Completeness: In the case that V returns 0, it is clear that P can decommit each entry and also send V π^{-1}. In the case that V sends 1, since there exists a Hamiltonian Path in G, there must also exists one in $\pi(G)$. Therefore, there must be some combination of n 1's such that each 1 is unique in its row and column. This is what P decommits to demonstrate there must have been a cycle in the original G.

Soundness: Consider a graph G that does not have a Hamiltonian cycle. This means that you cannot find n entries in the matrix that demonstrate a hamiltonian cycle. Therefore, whatever commitment you send to V can be either:

1. A permutation of G without a hamiltonian cycle
2. A adjacency matrix with a hamiltonian cycle that is not a permutation of G.

Therefore, P can only correctly respond to at most 1 b request from V, meaning that the probability it passes k rounds of this protocol is $1/2^k$.

4 Commitment Protocols

One Way Functions: Functions that are easy to compute and hard to invert.

$$x \overset{\$}{\leftarrow} X, f(x) = y \text{ is easy.}$$

$$\forall y, \text{ calculating } f^{-1}(y) \text{ is hard}$$

For example, we consider factoring to be a hard problem, so given two large primes p, q, calculating $N = p \cdot q$ is easy, but finding p, q given N is hard.

4.1 Implications of Existence

We have the following relation where each statement is implied by all following statements.

1. $P \neq NP$, or in other words, there exists some hard problems.
2. Average $P \neq$ Average NP, or there exists hard problems that are easy to sample.
3. $\exists$ One Way Functions

4.2 Definition of One Way Function

A one way function can be loosely defined as follows. There exists a challenger and adversary such that if the challenge sends the adversary $y = f(x)$ for a random $x \leftarrow X$, then the adversary wins if it can find an x' such that $y = f(x')$. We say a function f is one-way if the adversaries chances of winning quickly approaches 0 as $|x| = k$ increases.

4.2.1 Formal Definition

A function f is one way if

$$\forall k, \forall A^* \in \text{PPT}, \forall c, \Pr[A^*(f(x), 1^n) \in f^{-1}(f(x))] < \frac{1}{k^c}$$

This is saying, for all values of k (length of x), and for all adversaries A^*, the probability that A^* can find some x' where $f(x') = f(x)$ descreases faster than the inverse of any polynomial