1 3-Colorability

The Prover, $P$, picks a coloring $c : V \rightarrow [3]$. In each round $P$ randomly permutes the three colors and commits the graph and its coloring. The Verifier, $V$, is then permitted to (randomly) query the colors of the endpoints of a single edge, accepting iff the two colors are different. We first prove that the probability of successful cheating is exponentially small.

**Partial Soundness:** Suppose there does not exist a 3-coloring. Then there must be some monochromatic edge $e$ in the coloring committed by $P$. So $Pr(V$ chooses $e) \geq \frac{1}{m}$, and thus $P$ will cheat successfully with probability at most $1 - \frac{1}{m}$. By repeating the test $k$ times, this probability becomes exponentially small in $k$ and $m$. By repeating $km$ times, we drive the probability of successful cheating down to exponentially small in $k$.

**Completeness:** Is obvious. If $P$ knows a 3-coloring, then $V$ will always accept.

**Zero-Knowledge:** We give the following simulator for $P$ to show that this protocol is Zero-Knowledge. Enumerate the 6 ordered pairs of distinct elements of $[3]$ and pick one uniformly at random. Let this be the response of the Prover to the Verifier’s query $e$. If the map does have a 3-coloring, then each of the six pairs appears as the endpoints of $e$ in one of the permutations selected by the Prover. Since the prover permutes uniformly at random, the simulator behaves identically to the Prover.

2 Edge-Colorability

First convert the input graph $G = (V, E)$ into its line graph

$$L(G) = (E, \{(e, f)| e, f \text{ share an endpoint}\}).$$

Vertex-coloring $L(G)$ is the same as edge-coloring $G$. Just as in the previous question, the Prover picks a permutation $\pi$ on the color set $[d]$ uniformly at random. The Verifier queries an edge of $L(G)$ and the Prover reveals the colors of that edge’s endpoints. If $m = |E|$ then $m' = |E(L(G))| = O(m^2)$ so we only have a polynomial increase in the number of edges.

**Partial Soundness:** As before the probability of successful cheating is at most $1 - \frac{1}{m'}$. We repeat $km^2$ times and the probability of successful cheating is driven down to be exponentially small in $k$.

**Zero-Knowledge:** The simulator works identically to the one in the previous question except that it chooses from possible colors in $[d]$ rather than
[3]. Again, each possible pair \((a, b) \in [d] \times [d]\) appears with equal probability over the prover’s choice of color permutation so choosing uniformly at random is a good simulator.

### 3 Vertex Cover

The Prover chooses a permutation \(\pi : [n] \to [n]\) uniformly at random. Then, \(P\) commits the following information,

1. The permuted Adjacency matrix \(\pi(A)\)
2. The permutation \(\pi\)
3. The permuted vertex cover \(\pi(C)\) where \(C\) is expressed as a characteristic function \(C : V(G) \to \{0, 1\}\).
4. a randomly sorted list of the edges with vertices listed in random order.
5. a designated endpoint \(D_{\pi(i)\pi(j)}\) for each \((\pi(i), \pi(j))\).

The Verifier reveals one of three pieces of information.

1. \(\pi(A)\) and \(\pi\)
2. \(\pi(C)\)
3. an edge \((\pi(i), \pi(j))\) from the list and its designated endpoint \(D_{\pi(i)\pi(j)}, \pi(A)_{\pi(i)\pi(j)}, \) and \(\pi(C_{\pi(i)\pi(j)})\)

The first test allows \(V\) to see that the graph is correct. The second test allows \(V\) to see that the cover is the correct size. The third test allows \(V\) to see if a random edge (that is indeed in the graph) is covered by \(C\).

**Completeness:** If the tests all pass then every edge is covered by at least one vertex in the cover. Thus a prover that knows a vertex cover will always be accepted by \(V\).

**Soundness:** Again we have \(O(m)\) possible tests, where at least one must fail if there is cheating, so repeating \(km\) times yields a probability of mistake that is exponentially small in \(k\).

**Zero-Knowledge:** The simulator for this protocol works as follows. Choose a random permutation \(\pi : [n] \to [n]\). For test 1, return \(\pi(A)\) and \(\pi\). For test 2, return \(C \subset [n]\) with \(|C| = B\) chosen uniformly at random. For
test three, randomly select an edge \((i, j)\) of the graph and return \((\pi(i), \pi(j))\), one of its endpoints \(\pi(i)\) or \(\pi(j)\) each with probability \(\frac{1}{2}\), a 1 indicating that the edge is in the adjacency matrix, and another 1 indicating that the given endpoint is in the cover. In all cases the distribution of responses from the true prover is uniformly distributed with the choice of \(\pi\) so the simulator function identically.

**Extra Credit**

NOTE: All addition is modulo the bit-length of the longer addend \((2^n \text{ or } 2^{2n})\) even where it is not explicitly mentioned.

Let \(N\) be the positive integer and let \(p\) and \(q\) be the \(n\)-bit factors so \(N = pq\). The Prover \(P\) chooses two \(n\)-bit numbers \(a\) and \(c\) uniformly at random. Let \(b = p - a\ (\text{mod } 2^n)\) and \(d = q - c\ (\text{mod } 2^n)\). Now the prover also selects \(2n\)-bit numbers \(s_1, s_2, s_3, s_4\). The prover commits the following numbers.

1. \(a, b, c, d\),
2. \(s_1, s_2, s_3, s_4\),
3. \(ac + s_1, ad + s_2, bc + s_3, bd + s_4\), (all \(\text{mod } 2^{2n}\)) and
4. \(N + \sum_{i=1}^{4} s_i\ (\text{mod } 2^{2n})\).

Since \(a\) and \(c\) as well as \(s_1, \ldots, s_4\) were chosen uniformly and independently at random, the committed number are each random (though not independently). The Verifier \(V\) then asks to reveal one of 6 pieces of information.

1. \(a, c, s_4, ac + s_1\)
2. \(a, d, s_2, ad + s_2\)
3. \(b, c, s_3, bc + s_3\)
4. \(b, d, s_4, bd + s_4\)
5. \(s_1, s_2, s_3, s_4, N + \sum_{i=1}^{4} s_i\)
6. \(ac + s_1, ad + s_2, bc + s_3, bd + s_4, N + \sum_{i=1}^{4} s_i\)
In the first case, $V$ checks that indeed the values of $a, c$, and $s_1$ give the committed value for $ac + s_1$. The next three cases are handled similarly. In the fifth case, $V$ checks that the sum of $N$ and the $s_i$'s is the value given. In the last case, $V$ checks that the sum of the first 4 terms is equal to the last. We want to show that if all 6 tests are passed then indeed $p = a + b \pmod{2^n}$ and $q = c + d \pmod{2^n}$ is necessarily a valid solution for $N = pq$.

Test six ensures the following.

\[(a + b)(c + d) + \sum s_i = (ac + s_1) + (ad + s_2) + (bc + s_3) + (bd + s_4) \quad (1)\]
\[= N + \sum s_i \quad (2)\]

So, it follows that $(a+b)(c+d) = N$. However, this is impossible because then $(a+b)(c+d)$ is a solution. Therefore, it must be that one of the committed numbers in set 3 or 4 is not what it claims to be. Thus, one of tests 1-5 must fail. Therefore, we see that not all the tests can pass if $P$ does not have a valid solution.

**Zero-Knowledge:** To prove that this protocol is zero knowledge, we give a simulator for the prover as follows. Pick a test 1-6 at random. For tests 1-4, choose $n$-bit numbers $x, y$ and $2n$-bit number $z$ uniformly at random. The simulated prover returns $x, y, z$, and $xy + z$. For test 5, the simulator chooses $2n$-bit numbers $w, x, y, z$ uniformly at random and returns $w, x, y, z$, and $N + w + x + y + z$. For test 6, the simulator chooses $2n$-bit numbers $w, x, y, z$ uniformly at random and returns $w, x, y, z$, and $w + x + y + z$. In all cases the distribution of responses of the simulator exactly matches the distribution from the true prover.