ETH Zurich, Department of Computer Science
SS 2020

Prof. Ueli Maurer
Dr. Martin Hirt
Chen-Da Liu Zhang

Cryptographic Protocols
Solution to Exercise 11

11.1 Graph Coloring

The protocol is a proof of statement, it shows that $\mathcal{G}$ has a 3-coloring. Let $V = \{1, \dots, n\}$, and the 3-coloring be defined as a function $f : V \to \{1, 2, 3\}$.

Peggy		**Vic**
knows a 3-coloring f for $\mathcal{G} := (V, E)$		knows $\mathcal{G}$
choose a random permutation of the colors π let $f' = \pi \circ f$ $\forall i \in V$, commit to $f'(i)$ as C_i	$\xrightarrow{C_1, \dots, C_n}$	
	$\xleftarrow{(i,j)}$	let $(i, j) \in_R E$
open colors of vertices i and j	$\xrightarrow{d_i, d_j}$	check if $f'(i), f'(j) \in \{1, 2, 3\}$ and $f'(i) \neq f'(j)$

Completeness: It is easily verified that if G has a 3-coloring, then Vic always accepts. Peggy can answer all the Vic's queries correctly such that Vic is convinced as long as the commitment scheme is binding.

Soundness: The scheme has soundness $\frac{1}{|E|}$: if $\mathcal{G}$ does not have a 3-coloring, a cheating prover must commit to a coloring that has at least one edge whose vertices have the same color, or to colors that are not in $\{1, 2, 3\}$. Hence, with probability $\frac{1}{|E|}$, the verifier catches him, assuming the commitments are perfectly binding. When doing $n|E|$ sequential repetitions of the protocol, the soundness error is down to $(1 - \frac{1}{|E|})^{n|E|} \leq e^{-n}$.

Zero-Knowledge: The protocol is c-simulatable: Given (i, j), choose random colors σ_i, σ_j, and compute the commitments C_i, C_j. Since $|E|$ is polynomially large the protocol is zero-knowledge., assuming that the commitments are perfectly hiding.

11.2 Sudoku

In the following we use a commitment scheme of Type B.

The following protocol is a possible solution for this task:

Phase 1: Peggy commits to every cell of the Sudoku solution. Peggy additionally commits for every row, column and subgrid, to the numbers $\{1, \dots, n\}$ uniformly at random.

Phase 2: Vic chooses a challenge uniformly at random $c \in_R \{0, 1\}$.

Phase 3: If $c = 0$ Peggy opens all additional commitments (rows, columns, subgrids) and also the preprinted values of the Sudoku solution. Vic checks that in each additional row, column and subgrid the numbers from $\{1, \ldots, n\}$ appear, and also checks that the preprinted values of the Sudoku solution are consistent. And if $c = 1$, Peggy proves (using the ZK proof for equality) that the blobs between each row (resp. column, subgrid) in the Sudoku solution and the additionally committed row (resp. column, subgrid) are commitments to equal values.

Completeness: If Peggy knows the Sudoku solution, she can answer both challenges, so completeness follows directly.

Proof of Knowledge: The protocol is 2-extractable. Let the triples $(t, c, r), (t, c', r')$ be two triples of messages accepted by Vic with $0 = c \neq c' = 1$. Here the message t is the set of blobs that Peggy commits to (the Sudoku solution and the additionally committed rows, columns and subgrids). From the first triple, we obtain r, the decommitment to open all preprinted values in the Sudoku solution and all additionally committed rows, columns and subgrids. From the second triple, we obtain r', the zero knowledge equality proofs between the blobs corresponding to rows/columns/subgrids of the Sudoku solution and the additionally committed rows/columns/subgrids. Since the commitments are of type B, we can recover the original values of the Sudoku solution with overwhelming probability.

Zero-Knowledge: The simulator S can produce a transcript as follows: First, S commits to a fake Sudoku solution with valid preprinted values, and also for each row, column and subgrid, S commits to the numbers $\{1, \ldots, n\}$ uniformly at random. If V' sends the challenge $c = 0$, the simulator opens the preprinted values and the additionally committed rows, columns and subgrids. If V' sends $c = 1$, S uses the simulator S' for the blob equality protocol to compute a transcript of the corresponding equality proofs. Note that, by the computational hiding property of the commitments, the transcript produced by S' is computationally indistinguishable from the real interaction.

11.3 Permuted Truth Tables

a) Peggy chooses a random permuted truth table for the $\wedge$-function and commits to its elements. Vic chooses a random challenge bit c and sends it to Peggy. If $c = 0$, then Peggy opens the whole table and Vic checks if it is a permuted $\wedge$-table. If $c = 1$, Peggy takes the blobs (d_1, d_2, d_3) from the row corresponding to the triple (b_1, b_2, b_3) and proves (using the ZK protocol for equality) that $\forall i \in \{1, 2, 3\}$ d_i and c_i are commitments of the same value.

Note that the commitments used in the above construction are of type B (i.e., perfectly binding). We show that the above protocol is a zero-knowledge proof of the statement "the committed values (b_1, b_2, b_3) corresponding to the commitments (c_1, c_2, c_3) satisfy the relation $b_1 \wedge b_2 = b_3$."

Completeness: Follows immediately from the completeness of the protocol for blob equality.

Soundness: Assume that $b_1 \wedge b_2 \neq b_3$. If Peggy commits to a valid permuted truth table in the first step, Peggy cannot answer the challenge $c = 1$ as there is no row in this table with with commitments corresponding to b_1, b_2, b_3. If Peggy commits to an invalid table, then she cannot answer the challenge $c = 0$, as the commitment is binding. Hence, the cheating probability of Peggy for each round is approximately $1/2$ (the "approximately" stems from the fact that, in case $c = 1$, Peggy might still be able, with some small probability, to cheat in the equality proof).

ZERO-KNOWLEDGE: We prove the (computational) zero-knowledge property only informally. We need to show that there exists an efficient simulation S producing a transcript which is (computationally) indistinguishable from the transcript resulting from a real protocol execution between the prover P and (a possibly dishonest) verifier V'.

The simulator S can produce a transcript as follows: First, S computes a valid permuted truth table and commits to it. If V' sends the challenge $c = 0$, the simulator opens the committed table. If V' sends $c = 1$, S uses the simulator S' for the blob equality protocol to compute a transcript of a proof of equality for $c_i = d_i$ ($i = 1 \dots 3$), where the d_i's are commitments corresponding to a randomly chosen row of the permuted truth table. Note that, by the computational hiding property of the commitments, the transcript produced by S' is computationally indistinguishable from the real interaction even if the d_i's are commitments to different values than those in the c_i's.

b) If Peggy knows the input to the circuit, then she can compute (by evaluating the circuit in a gate-by-gate manner) the bits on the wires. She commits to all those bits and sends the blobs to Vic. Subsequently, she uses the protocol from **a)** for each gate ($\neg$-gates are treated similarly to $\wedge$-gates) to prove that the committed values are consistent with the circuit. To convince Vic that the output of the circuit is in fact 1, Peggy and Vic use a fixed commitment of 1, i.e., a commitment that is hard-coded into the protocol.

c) In the BCC protocol from the lecture, when processing the circuit, Peggy blinds every wire using a random bit. In the protocol from **b)**, this is not necessary, but we need the additional zero-knowledge proofs of equality of committed values.