ETH Zurich, Department of Computer Science
SS 2017

Dr. Martin Hirt
Chen-Da Liu Zhang

Cryptographic Protocols
Solution to Exercise 7

7.1 Homomorphic Commitments

Note that a blob committing to 0 is a quadratic residue, and, since t is a quadratic non-residue (with $\left(\frac{t}{m}\right) = +1$), a blob committing to 1 is a quadratic non-residue b (with $\left(\frac{b}{m}\right) = +1$). Thus, the scheme is of type B, where the computational hiding property relies on the QR assumption, which states that modulo an RSA prime m it is hard to distinguish quadratic residues from quadratic non-residues (with $\left(\frac{b}{m}\right) = +1$).

a) Denote by $b_0 = r_0^2 t^{x_0}$ and $b_1 = r_1^2 t^{x_1}$ two blobs to bits x_0 and x_1, respectively. By multiplying b_0 and b_1, one obtains

$$b = b_0 \cdot b_1 = r_0^2 \cdot r_1^2 \cdot t^{x_0+x_1}.$$

This is a commitment to $x_0 \oplus x_1$: If $x_0 = x_1$ (i.e., $x_0 \oplus x_1 = 0$), then b is a quadratic residue (with randomness $r_0 r_1$ if $x_0 = x_1 = 0$ and $r_0 r_1 t$ if $x_0 = x_1 = 1$). If $x_0 \neq x_1$ (i.e., $x_0 \oplus x_1 = 1$), then b is a quadratic non-residue (with $\left(\frac{b}{m}\right) = +1$), where the randomness is $r_0 r_1$.

b) Let $b = r^2 t^x$ be the blob to x. By multiplying b by t one obtains

$$b' = b \cdot t = r^2 \cdot t^{x+1}.$$

If $x = 0$, b' is a quadratic non-residue and thus a commitment to 1. If $x = 1$, b' is a quadratic residue and thus a commitment to 0. The randomness for b' is r.

c) If all binary operations could be implemented in such a fashion, the BCC protocol would become substantially simpler: At the beginning, Peggy would commit to the satisfying input, and then Vic would evaluate the circuit on the blobs. In the end, Peggy would prove that the resulting blob b is indeed a commitment to one by proving that bt is a quadratic residue using the Fiat-Shamir protocol.

d) As shown in **a)**, if $x_0 = x_1$, $b_0 \cdot b_1$ is a quadratic residue, a fact that Peggy can prove using the Fiat-Shamir protocol. Moreover, if $x_0 \neq x_1$, then $b := b_0 \cdot b_1$ is a quadratic non-residue (with $\left(\frac{b}{m}\right) = +1$) and thus $b_0 \cdot b_1 \cdot t$ is a quadratic residue, which, again, can be proved using the Fiat-Shamir protocol.

7.2 Permuted Truth Tables

a) Peggy chooses a random permuted truth table for the gate 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 permutation of the AND gate function 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 AND relation."

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 $c_i = d_i$ $(i = 1 \ldots 3)$, 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, similar to the AND gate) 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 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.

7.3 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 $\{1, \ldots, n\}$ 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.