Cryptographic Protocols
Solution to Exercise 11

11.1 Information-Theoretic Commitment Transfer Protocol

a) In protocol \texttt{Commit} the state of the dealer \(D\) consists of commit polynomial \(g\), where the committed value is \(g(0) = s\). Every player \(P_i\) stores the commit-share \(s_i = g(\alpha_i)\).

b) The commitment transfer protocol \texttt{CTP} allows to transfer a commitment from a player \(P\) to a player \(P'\). The protocol works as follows:

1. \(P\) sends the polynomial \(g\) to \(P'\).
2. Each \(P_i\) sends \(s_i\) to \(P'\).
3. \(P'\) checks that all but at most \(t\) of the received \(s_i\)'s lie on \(g\). If so, he accepts \(g(0)\) as value for \(s\), otherwise he assumes that he did not receive any value for \(s\).

The above protocol is secure for \(t < n/3\):

\textbf{Privacy:} Straight-forward as only \(P'\) receives values in the protocol and he only obtains the values which he is supposed to receive.

\textbf{Correctness:} This can be argued along the lines of the correctness of the protocol \texttt{Open} from the lecture notes: Assume that \(P\) sends \(P'\) some wrong polynomial \(g' \neq g\). Then, at most \(t\) of the commit shares can lie on polynomial \(g'\). Hence the commit shares of at least \(n - t\) players do \textit{not} lie on \(g'\). As at most \(t\) of those players might be corrupted, there are at least \(n - 2t > t\) players who will send commit shares that do not lie on \(g'\) to \(P'\), and therefore \(P'\) will not accept \(g(0)\) as value for \(s\).

In the case that \(P'\) did not receive a valid value for \(s\), he can accuse \(P\) via broadcast and the whole protocol is repeated, using broadcast instead of sending values.

11.2 Information-Theoretic Commitment Multiplication Protocol

In the following we will use \(f_a\) and \(f_b\) to denote the polynomials used in the commitment sharing protocol (CSP) to share the values \(a\) and \(b\), respectively. Furthermore, let \(f_d := f_a \cdot f_b\).

a) We show that correctness and privacy are satisfied:

\textbf{Privacy:} In steps 1-3, privacy is guaranteed by the privacy of the CSP, i.e., no information on \(a, b,\) and \(d\) is revealed in these steps. In step 4, the players only see values they already know, namely \(d_i = a_i \cdot b_i\), hence again no information is revealed. Finally, the commitments to some \(a_i, b_i,\) and \(d_i\) are opened only if \(D\) or the player \(P_i\) is corrupted, which means that the adversary already knows them.

\textbf{Correctness:} Any dealer who is not disqualified must successfully complete the CSP for values \(a\) and \(b\). Thus, every player \(P_i\) ends up with shares \(a_i\) on \(f_a\) and \(b_i\) on \(f_b\). Suppose, \(D\) commits to a value \(d' \neq d\) and shares it using a polynomial \(f_{d'} \neq f_d = f_a \cdot f_b\) in protocol CSP.\(^1\) Since both \(f_d\) and \(f_{d'}\) have degree at most \(2t\),

\(^1\)Note that the dealer cannot share \(d'\) using \(f_d\) as can easily be seen by inspecting the CSP.
they can have at most $2t$ points in common. Thus, there exists at least one honest player $P_i$ for which $d'_i \neq a_ib_i$, where $d'_i$ is his share of $d'$.\footnote{The condition $t < n/3$ implies that there are at least $2t + 1$ honest players.} This player will accuse the dealer and prove that he is corrupted by opening $a_i$, $b_i$, and $d_i$.

b) Let $n = 3t$, and assume that the players $P_1, \ldots, P_t$ are corrupted, where $P_1$ plays the role of $D$. In order to achieve that at the end of the protocol the players accept a false $d' \neq ab$, the corrupted players have the following strategy:

1. In step 1, $D$ chooses $d'$ (instead of $d$) and is committed to it.
2. Step 2 is executed normally, i.e., $D$ invokes the CSP for $a$ and $b$.
3. In step 3, $D$ invokes the CSP for $d'$, with the (unique) degree-$2t$ polynomial $f_{d'}(x)$, such that $f_{d'}(0) = d'$ and
   $$f_{d'}(\alpha_i) = f_a(\alpha_i) \cdot f_b(\alpha_i)$$
   for $i = t + 1, \ldots, n$.
4. The corrupted players do not complain in step 4.

As $f_{d'}(x)$ is chosen such that it satisfies the consistency check for all honest players, no player will complain and the commitment to $d'$ will be accepted.

11.3 Commitment Multiplication Protocol for ElGamal

The solution is a particular instantiation of the protocol in Exercise 10.3d).

The commitment multiplication protocol CMP allows a player $P$ that is committed to some values $a$ and $b$, to commit to their product $d = ab$.

Let use denote $A = (g^a, \gamma^ah^\alpha)$, $B = (g^\beta, \gamma^bh^\beta)$ the blobs of $a$ and $b$.

The inputs of $P$ are $a, b, \alpha, \beta$ and the inputs of the other players $P_i$ are $A, B$.

In the first step, $P$ computes $D = (g^\delta, \gamma^d h^\delta)$, where $\delta \in R Z_q$ and broadcasts $D$. Then, $P$ proves in zero-knowledge (using the generic zero-knowledge proof of knowledge of a preimage of a one-way homomorphism we saw in the lecture) that he knows a pre-image of $(A, D)$ with respect to the homomorphism

$$\psi : Z_q \times Z_q \times Z_q \rightarrow G \times G, \quad (a, \alpha, \rho) \mapsto \left((g^\alpha, \gamma^ah^\alpha), (g^{\alpha\beta + \rho}, \gamma^{abh^{\alpha\beta + \rho}})\right).$$

The pre-image of $(A, D)$ that $P$ uses in the generic zero-knowledge protocol is $(a, \alpha, \delta - a\beta)$. Observe that knowledge of a pre-image of $(A, D)$ corresponds to knowing the information to open the blobs $A$ and $D$ in a way so that the value $d$ is the product of $a$ and $b$.

Observe that even without knowing $b$ and $\beta$, any party is able to evaluate the homomorphism using $B$ instead, since:

$$\psi(a, \alpha, \rho) = \left((g^\alpha, \gamma^ah^\alpha), (g^{\alpha\beta + \rho}, \gamma^{abh^{\alpha\beta + \rho}})\right).$$

It is easily seen that such a function is a homomorphism and the condition $\exists u, l$ such that:

1. $[u] = (A, D)^l$.
2. $\forall c_1, c_2 \in \mathcal{C}$ with $c_1 \neq c_2 \quad \gcd(c_1 - c_2, l) = 1$
is satisfied.

We saw that these conditions (plus a suitable challenge space) are sufficient to prove that the generic protocol is a zero-knowledge proof of knowledge.

A player $P_i$ accepts the protocol if and only if $P$ succeeds in the zero-knowledge proof. $P$ outputs the randomness $\delta$ of $D$, and the other players $P_i$ output $D$.

Observe that the zero-knowledge proof has to be done between $P$ and all other parties $P_i$. To execute such a distributed zero-knowledge proof, $P$ broadcasts all of his messages. The challenge has to be chosen between all players. To do that, each player commits to a random value. Then, the sum of these values is opened and used as the challenge.

Since $P$ broadcasts $D$ as well as all the messages in the zero-knowledge proof and since the challenge is chosen in a distributed fashion, the honest players agree (by broadcasting their output) on whether or not the protocol was successful.