ETH Zurich, Department of Computer Science
SS 2020

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

Cryptographic Protocols
Exercise 8

8.1 Impossibility and Feasibility Proofs

In the context of two-party computation between P_1 and P_2, we saw in the lecture that if one of the parties is corrupted, it is impossible to compute securely the AND function $b_1 \wedge b_2$, where b_1, b_2 are the input values of P_1 and P_2 respectively. However, some functions can still be securely constructed.

a) Construct a protocol that securely computes the XOR of the two input bits $b_1 \oplus b_2$ in the presence of a passive adversary that corrupts one of the players.

More generally, we can describe any binary Boolean function $f : \{0,1\}^2 \to \{0,1\}$ by a vector $(o_{00}, o_{01}, o_{10}, o_{11})$, where $f(b_1, b_2) = o_{b_1 b_2}$. For example, the AND function corresponds to the vector $(0,0,0,1)$, and the OR function corresponds to the vector $(0,1,1,1)$.

b) Show that a binary function can be securely constructed in the presence of a passive adversary if it is specified by a vector with an even number of ones.

c) Show that it is impossible to securely construct a binary function specified by a vector with an odd number of ones in the presence of a passive adversary.

HINT: Reduce it to the AND function.

8.2 Not Sending Values

Consider the case where the at most $t < n/2$ corrupted players can withhold information (but do not send wrong values).

For a finite field $\mathbb{F}$, let $[a] = (a_1, \ldots, a_n)$ be a sharing of a value $a \in \mathbb{F}$ among the players $P_1, \ldots, P_n$. The share a_i of P_i is a point on some polynomial $f \in \mathbb{F}[X]$ of degree at most t, i.e., $a_i = f(\alpha_i)$, where $\alpha_1, \ldots, \alpha_n$ are distinct values in $\mathbb{F} \setminus \{0\}$.

Devise a protocol that allows the players to reconstruct a share of a corrupted player. Keep in mind that in your protocol up to $t < n/2$ players can be corrupted and may not send values they are supposed to send.

8.3 Perfectly Binding/Hiding Commitments

a) Prove that it is not possible that a commitment scheme is both perfectly hiding and perfectly binding.

For a string-commitment scheme of type H, let $C_H(x, r)$ denote the function that for a string $x \in \{0,1\}^*$ computes the corresponding blob b, where $b \in \{0,1\}^*$. Similarly, for a commitment scheme of type B, let $C_B(x, r)$ denote the function that for an $x \in \{0,1\}^*$ computes the corresponding blob $b \in \{0,1\}^*$. We combine these two schemes to design the following three schemes:

1. The blob b' corresponding to x is computed as $b' = (C_H(x, r_1), C_B(x, r_2))$.
2. The blob b' corresponding to x is computed as $b' = C_H(C_B(x, r_1), r_2)$.
3. The blob b' corresponding to x is computed as $b' = C_B(C_H(x, r_1), r_2)$.

b) Show that the three schemes are commitment schemes.

c) Which of these schemes are of type H/type B?