ETH Zurich, Department of Computer Science
SS 2017

Dr. Martin Hirt
Chen-Da Liu Zhang

Cryptographic Protocols
Exercise 9

9.1 Shamir Sharings

Let $\mathbb{F}$ be a finite field and $\alpha_1, \ldots, \alpha_n$ be fixed, distinct values in $\mathbb{F}\backslash\{0\}$.

a) Let $s_1, \ldots, s_n$ be arbitrary values in $\mathbb{F}$. Show that there exists a *unique* polynomial $f \in \mathbb{F}[X]$ of degree at most $n-1$ that goes through the points (α_i, s_i).

b) Show that any subset of at most t players have no information about a secret that is Shamir-shared with a polynomial of degree at most t.

c) Consider a 3-party setting with an adversary that passively corrupts P_2. Let $a \in \mathrm{GF}(5)$ be the input of P_1 and $b \in \mathrm{GF}(5)$ that of P_3. Assume a and b are shared via polynomials of degree at most $t = 1$ with $\alpha_1 = 1$, $\alpha_2 = 2$, and $\alpha_3 = 3$ as evaluation points.

Suppose that the players, to compute $c = ab$, locally multiply their shares and then open the product. Show that, given the shares of c (obtained when c was reconstructed) and the shares of player P_2, the adversary can determine a and b.

d) In an alternative sharing protocol, the dealer chooses a random sharing polynomial g with degree *exactly* t. Show that the alternative sharing protocol is not private, i.e., that it gives away information about the secret to the adversary.

Hint: Consider the case where the adversary corrupts t players.

9.2 Circuit Evaluation

In the lecture we have seen protocols for adding and multiplying shared values. Hence, players can evaluate circuits over a finite field $\mathbb{F}$ with input, output, addition, and multiplication gates. Let $|\mathbb{F}| = p$ for a prime p. Express the following tasks in terms of addition and multiplication:

a) Compute the multiplicative inverse x^{-1} of $x \in \mathbb{F}$.

b) Execute the instruction

$$z = \begin{cases} x & \text{if } c = 0 \\ y & \text{otherwise,} \end{cases}$$

where x, y, z, c are values in $\mathbb{F}$.

HINT: First, find a solution that works for $c \in \{0, 1\}$. Then, solve the general case.

9.3 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 inputs 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.