Exercise Sheet #8

Advanced Cryptography 2021

Exercise 1 Interactive Proof Systems

1. Let us consider the following Square number proof:
   - Statement: $x = (x_1, \ldots, x_n, m)$ where $m, x_1, \ldots, x_n$ are positive integers and n is even.
   - Witness: $w = (y_1, \ldots, y_n)$ such that for all i we have $y_i^2 \equiv x_i \mod m$.
   - The verifier V chooses a subset $I \subseteq \{1, \ldots, n\}$ with $|I| = \frac{n}{2}$.
   - The prover P sends y_i to V for all $i \in I$.
   - The verifier checks whether $y_i^2 \equiv x_i \mod m$ for all $i \in I$.

   Specify a witness relation R_c for which the completness bound c is equal to 1, and a witness relation R_s for which the soundness bound s is equal to 0. In both cases, give a completness and soundness bound.

2. Show that every language in NP has an interactive proof system (with polynomial-time prover and verifier) with perfect completeness and with soundness 0. (More exactly, for every language $L \in$ NP, there is a relation R such that there is a proof system for R with perfect completeness and with soundness 0.)

3. Let (P, V) be an interactive proof system for some relation R with soundness s and completeness c. Let (P°, V°) be the following proof system:
   - On input (x, w), P° executes $P(x, w)$ $|x|$ times sequentially. (That is, P° runs $P(x, w)$. When $P(x, w)$ terminates, $P(x, w)$ is run again, and so on. Each execution of $P(x, w)$ uses independent randomness, i.e., the different executions of $P(x, w)$ do not have any common data except x and w.)
   - On input x, V° executes $V(x)$ $|x|$ times sequentially. V° outputs 1 if and only if all invocations of V have output 1.

   Prove that (P°, V°) is a proof system for R with soundness $s^{|x|}$ and with completeness $c^{|x|}$

Exercise 2 Σ-Protocol for $\mathcal{P}$ (final 2011)

We consider an alphabet Z, a polynomial P, and a predicate R. We assume that R can be computed in polynomial time. Given $x \in Z^*$, we let

$$R_x = \{w \in Z^*; R(x, w) \text{ and } |w| \leq P(|x|)\}$$

where $|x|$ denotes the length of x. We define the language L from R by

$$L = \{x \in Z^*; R_x \neq \emptyset\}$$

1. In this question, we assume that there is an algorithm $\mathcal{A}$ such that for any $x \in L$, we obtain $\mathcal{A}(x) \in R_x$ and that for any $x \in Z^*$, the running time of $\mathcal{A}(x)$ is bounded by $P(|x|)$.

   Construct a Σ-protocol for L. Carefully specify all protocol elements and prove all properties which must be satisfied.

Exercise 3 Combined Proofs (final 2011)

Let $Z = \{0, 1\}$ be an alphabet. We consider two Σ-protocols Σ_1 and Σ_2 for two languages L_1 and L_2 over the alphabet Z defined by two predicates R_1 and R_2. We assume that Σ_1 and Σ_2 use the same challenge set E which is given a group structure with a law $+$. For Σ_i, $i \in \{1, 2\}$, we denote $\mathcal{P}_i$ the prover algorithm, V_i the verification predicate, $\mathcal{E}_i$ the extractor, and $\mathcal{S}_i$ the simulator.

2. (**AND proof**) Construct a Σ protocol $\Sigma = \Sigma_1$ AND Σ_2 for the language defined by

$$R((x_1, x_2), (w_1, w_2)) \iff R_1(x_1, w_1) \text{ AND } R_2(x_2, w_2)$$

(**OR proof**) In the remaining of the exercise, we now let

$$R((x_1, x_2), w) \iff R_1(x_1, w) \text{ OR } R_2(x_2, w)$$

This predicate defines a new language L. We construct a new Σ-protocol $\Sigma = \Sigma_1$ OR Σ_2 for L by

- $\mathcal{P}((x_1, x_2), w; r_1, r_2)$ finds out i such that $R_i(x_i, w)$ holds, sets $j = 3 - i$, then picks a random $e_j \in E$ and runs $\mathcal{S}_j(x_j, e_j; r_1) = (a_j, e_j, z_j)$. Then, it runs $\mathcal{P}(x_i, w; r_2) = a_i$ and yield (a_1, a_2).
- Upon receiving e, $\mathcal{P}((x_1, x_2), w, e; r_1, r_2)$ sets $e_i = e - e_j$, runs $\mathcal{P}(x_i, w, e_i; r_2) = z_i$ and yields (e_1, e_2, z_1, z_2).

The verification predicate is

$$V((x_1, x_2), (a_1, a_2), e, (e_1, e_2, z_1, z_2)) \iff \begin{cases} e = e_1 + e_2 \text{ AND} \\ V_1(x_1, a_1, e_1, z_1) \text{ AND} \\ V_2(x_2, a_2, e_2, z_2) \end{cases}$$

3. Show that Σ is complete and works in polynomial time.
4. Construct an extractor $\mathcal{E}$ for Σ and show that is works, in polynomial time.
5. Construct a simulator $\mathcal{S}$ for Σ and show that is works, in polynomial time.