ETH Zurich, Department of Computer Science
SS 2020

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

Cryptographic Protocols
Solution to Exercise 2

2.1 Definition of Interactive Proofs

a) As an extreme example, the "halting problem" is known to be undecidable and therefore not in **IP**. There are also decidable problems that are not in **IP**. For example, some problems related to the game of Go are **EXPSPACE**-complete.

b) Consider an interactive proof (P, V) for a language L, where P and V are probabilistic. We want to construct a deterministic $\widehat{P}$ so that $(\widehat{P}, V)$ is an interactive proof that accepts the same language.

In the random experiment between the probabilistic P and the probabilistic V, denote by $p^{\mathsf{acc},x}$ the probability that V accepts x. Moreover, let $p_r^{\mathsf{acc},x}$ be the probability that V accepts if P's randomness is fixed to r and p_r that r is chosen as P's randomness.

On input x, $\widehat{P}$ does as follows: It runs the protocol (P, V) with all possible random inputs for both Peggy and Vic and computes for each fixed randomness r of Peggy, the set of fixed randomness s of Vic that are accepted in the protocol (P, V).[1] Then, $\widehat{P}$ chooses to run P with the randomness r' that maximizes $p_{r'}^{\mathsf{acc},x}$. Note that $p_{r'}^{\mathsf{acc},x}$ is the probability that V accepts in an interaction with $\widehat{P}$. Observe that

$$p \leq p^{\mathsf{acc},x} = \sum_r p_r^{\mathsf{acc},x} p_r \leq \sum_r p_{r'}^{\mathsf{acc},x} p_r = p_{r'}^{\mathsf{acc},x} \sum_r p_r = p_{r'}^{\mathsf{acc},x}.$$

Thus, if $x \in L$, the probability that $\widehat{P}$ convinces V is at least p. Conversely, since V is such that it accepts a proof for a word $x \notin L$ with probability at most q no matter which prover it interacts with, $(\widehat{P}, V)$ is trivially sound.

Given the considerations in **b)**, the prover's algorithm is assumed to be deterministic for the remainder of this task.

c) If both P and V are deterministic, for every x there is but a single transcript between P and V. Since (P, V) is an interactive proof, this transcript is accepted by V if and only if $x \in L$. Thus, the transcript serves as an efficiently verifiable witness if $x \in L$ and if $x \notin L$, no transcript can convince V. Thus, $L \in \mathbf{NP}$.

d) Let (P, V) be an interactive-proof protocol with $q = 0$, i.e., V never accepts some $x \notin L$. The situation is similar to that in **c)**: If $x \in L$, the fact that $p > q = 0$ implies that there exists an accepting transcript between P and V, which is a witness for x. If $x \notin L$, $q = 0$ implies that no such transcript exists. Thus, $L \in \mathbf{NP}$.

e) For $n \geq 1$ we define the protocol (P', V') as follows: For input x, the protocol (P, V) is repeated sequentially n times. V' accepts x if and only if V accepted x at least $p^* \cdot n$ times. We show now that for n large enough (P', V') meets the definition of an interactive proof with parameters p', q'. To do that, let us fix $p^* = \frac{p+q}{2}$, and $\epsilon = \frac{p-q}{2}$.

[1] Recall that the prover's algorithm need not be efficient.

For $i = 1, \ldots, n$, let X_i be the random variable that is 1 if V accepts x in the i^{th} round and 0 otherwise, and set $\overline{X} := \frac{1}{n}\sum X_i$ and $\mu := E[\overline{X}]$. Note that $\mu = \mathsf{P}[X_i = 1]$ for any i.

Consider now $x \in L$. In that case $\mu = \mathsf{P}[X_i = 1] \geq p^* + \varepsilon$. Hence,

$$\begin{aligned}
\mathsf{P}[V' \text{ rejects } x] &\leq \mathsf{P}[\sum X_i \leq p^* n] \\
&= \mathsf{P}[\sum X_i \leq (p^* + \varepsilon)n - \varepsilon n] \\
&= \mathsf{P}[\overline{X} \leq (p^* + \varepsilon) - \varepsilon] \\
&\leq \mathsf{P}[\overline{X} \leq \mu - \varepsilon] \\
&\leq e^{-2n\varepsilon^2}.
\end{aligned}$$

Consider now $x \notin L$. In that case $\mu = \mathsf{P}[X_i = 1] \leq p^* - \varepsilon$. Hence,

$$\begin{aligned}
\mathsf{P}[V' \text{ accepts } x] &\leq \mathsf{P}[\sum X_i \geq p^* n] \\
&= \mathsf{P}[\sum X_i \geq (p^* - \varepsilon)n + \varepsilon n] \\
&= \mathsf{P}[\overline{X} \geq (p^* - \varepsilon) + \varepsilon] \\
&\leq \mathsf{P}[\overline{X} \geq \mu + \varepsilon] \\
&\leq e^{-2n\varepsilon^2}.
\end{aligned}$$

Concerning the number n of repetitions, note that if for example $p' = 1 - \delta$ and $q' = \delta$ for $\delta > 0$, then completeness and soundness are satisfied if $e^{-2n\varepsilon^2} \leq \delta$. This is the case if and only if $n \geq \frac{1}{2}\varepsilon^{-2}\ln(\delta^{-1})$. This means that δ can be made *negligible*, whereas ε needs to be *noticeable* (asymptotically in the length of the input to P and V) in order for n to be *polynomial*.[2]

2.2 Discrete Logarithms and Interactive Proofs

a) Consider the following interactive protocol:

Peggy		**Vic**
knows x		knows $z_1 = g^x$, $z_2 = h^x$
choose $k \in_R \mathbb{Z}_p$ compute $t_1 = g^k$, $t_2 = h^k$	$\xrightarrow{(t_1, t_2)}$	
	$\xleftarrow{c}$	let $c \in_R \mathcal{C} \subseteq \mathbb{Z}_p$
$r = k + cx$ in $\mathbb{Z}_p$	$\xrightarrow{r}$	check if $(g^r, h^r) \stackrel{?}{=} (t_1 z_1^c, t_2 z_2^c)$

b) The protocol in **a)** can be seen both as a proof of the statement that $\log_g z_1 = \log_h z_2$ as well as proof of knowledge of the exponent x such that $z_1 = g^x$ and $z_2 = h^x$. We do the analysis as a proof of statement.

COMPLETENESS: It is easily seen that Vic always accepts if Peggy knows x and follows the protocol.

SOUNDNESS: Suppose $z_1 = g^{x_1}$ and $z_2 = h^{x_2}$ for $x_1 \neq x_2$. Consider a cheating prover P' and assume her first message is (t_1, t_2). Such a message can be seen as

[2] See the the lecture notes, Section 1.6, for definitions of negligible, noticeable, and polynomial.

$(t_1, t_2) = (g^{k_1}, h^{k_2})$ where $k_1 = \log_g t_1, k_2 = \log_h t_2$. Consider a challenge $c \in \mathcal{C}$. A reply r causing V to accept must satisfy

$$(g^r, h^r) = (g^{k_1} g^{cx_1}, h^{k_2} h^{cx_2}),$$

which is equivalent to

$$(g^r, g^{\ell \cdot r}) = (g^{k_1} g^{cx_1}, g^{\ell \cdot k_2} g^{\ell \cdot cx_2}),$$

where $\ell \in \mathbb{Z}_p$ is such that $h = g^\ell$. Thus, Vic accepts if and only if

$$r = k_1 + cx_1$$
$$r = k_2 + cx_2,$$

or, equivalently, if $k_1 + cx_1 = k_2 + cx_2$. This is satisfied only by a single $c \in \mathbb{Z}_p$, namely by $c = (k_2 - k_1)/(x_1 - x_2)$ (where the denominator is non-zero since $x_1 \neq x_2$). Thus, Vic accepts only if said c is chosen, i.e., with probability $1/|\mathcal{C}|$.

Honest-Verifier Zero-Knowledge: For the honest verifier V, the zero-knowledge property is proved by showing that for any given challenge c, one can sample transcript triples with the correct conditional distribution. For a given c, simply sample a random r and set $t_1 := g^r z_1^{-c}$ and $t_2 := h^r z_2^{-c}$. The reader can verify that this results in the proper distribution.

c) Consider the mapping $\phi : \mathbb{Z}_p \to G$, $x \mapsto g^x$. For a group element $z \in G$, Schnorr's protocol allows to prove knowledge of a preimage x of z (w.r.t. to ϕ). The protocol in **a)** proceeds exactly like Schnorr's except that it works for the mapping $\phi' : \mathbb{Z}_q \to G \times G$, $x \mapsto (g^x, h^x)$. We will see in the next weeks how this can be generalized to any mapping that is a so-called *one-way homomorphism* between two groups.

2.3 A Modification of the Schnorr Protocol

Completeness: It is easily verified that if Peggy is honest and knows x, then Vic always accepts.

Soundness: From the prover's replies to two different challenges for the same first message t, one can compute x such that $h^x = z$: Let (t, c, r) and (t, c', r') be two accepting transcripts with $c \neq c'$. That is, $h^r = t^c z$ and $h^{r'} = t^{c'} z$. By dividing the first equation by the second one we get:

$$h^{r-r'} = t^{c-c'} = h^{k(c-c')},$$

which implies that $k \equiv_q \frac{r-r'}{c-c'}$. Since $h^r = t^c z = h^{kc+x}$, one can compute $x \equiv_q r - kc$. Note that since $|H| = q$ is prime, $c - c' \neq 0$ has an inverse modulo q.

Zero-Knowledge: This protocol is not zero-knowledge, since on challenge 0, r is the discrete log of z, which is unknown to the simulator.

2.4 IP and PSPACE

In order to prove that **IP** $\subseteq$ **PSPACE**, let (P, V) be an interactive proof for a language L. We argue that $L \in$ **PSPACE**. Given an input $x \in \{0,1\}^n$, we compute exactly (using polynomial space) the maximum probability with which a prover can make V accept.

Although the prover is allowed to be all-powerful, we will see that the optimal strategy can be computed in **PSPACE** and so it suffices to consider a **PSPACE** prover in general. Imagine a tree where each node at level i (with the root at level 0) corresponds to some sequence of i messages exchanged between the prover and verifier. This tree has polynomial depth (since V can only run for polynomially many rounds), and each node has at most 2^{n^c} children (for some constant c), since messages in the protocol have polynomial

length. We recursively assign values to each node of this tree in the following way: a leaf node is assigned 0 if the verifier rejects, and 1 if the verifier accepts. The value of an internal node where the prover sends the next message is the maximum over the values of that node's children. The value of an internal node where the verifier sends the next message is the (weighted) average over the values of that node's children. The value of the root determines the maximum probability with which a prover can make the verifier accept on the given input x, and this value can be computed in polynomial space. If this value is greater than $3/4$, then $x \in L$; and if it is less than $1/2$ then $x \notin L$.