

## Solution of Exercise Sheet 1

Out: October 29, 2008

Due: November 7, 2008, before noon

**Problem 1: Some Interactive Proofs (9 Points)**

In the following, several (good and not so good) proof systems are given. For each, specify their witness relation  $R$ .<sup>1</sup> For each of the proofs, give (optimal) soundness and completeness bounds  $c$  and  $s$ .

(a) Square number proof:

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

**Solution.**

- We assume the given proof system proves that all  $x_i$  of  $x$  are squares. Then the corresponding witness relation is  $R := \{((x_1, \dots, x_n, m), (y_1, \dots, y_n)) \mid y_i^2 \equiv x_i \pmod{m} \text{ for } i = 1, \dots, n, n \text{ even}\}$ .
- Completeness: Given  $I$  by the verifier  $V$ , the prover  $P$  will always be able to send the square roots  $y_i$  for all  $i \in I$ , since all  $x_i$  are squares. All checks  $y_i^2 \equiv x_i \pmod{m}$  by  $V$  will then succeed and  $V$  will accept. Hence we have completeness bound  $c = 1$ .
- Soundness: We now compute the success probability of the best malicious prover  $P^*$  convincing the honest verifier  $V$  of a false statement. The best strategy for a malicious prover is to use exactly one non-square and  $n - 1$  squares for  $x$  (using more non-squares only increases the chance of  $V$  asking for the root of a non-square). Let  $x_j$  be the non-square. The verifier chooses the set  $I$ . Now there are two cases:

---

<sup>1</sup>Formally, you could specify any witness relation and then just say that the proof has terrible completeness and soundness. This, of course, is not the task. You should specify a *sensible* relation that matches the intention behind the proof.

- $j \in I$ :  $V$  wants to see the root of  $x_j$ . Since  $x_j$  is a non-square  $P^*$  cannot send a number  $z$ , such that  $z^2 \equiv x_j \pmod{m}$ .  $V$ 's check fails, it will output 0.
- $j \notin I$ :  $V$  does not want to see the root of  $x_j$ . Since all other  $x_i, i \neq j$  are squares,  $P^*$  is able to send those roots  $y_i$  for  $i \in I$ .  $V$ 's checks may succeed and it may output 1.

As  $j \in \{1, \dots, n\}$  is chosen independently of  $I$ , and  $|I| = \frac{n}{2}$ , the probability of  $j \in I$  is  $\frac{n/2}{n} = \frac{1}{2}$ . Hence  $V$  will output 0 with probability at least  $\frac{1}{2}$ , hence the soundness bound is  $s = \frac{1}{2}$ .

**Solution.**

- Alternatively we could assume that the given proof system proves that at least half of the  $x_i$  of  $x$  are squares. Then the corresponding witness relation is  $R := \{((x_1, \dots, x_n, m), (y_1, \dots, y_n)) \mid \exists I \subseteq \{1, \dots, n\} : |I| \geq \frac{n}{2} \wedge y_i^2 \equiv x_i \pmod{m} \text{ for all } i \in I, n \text{ even}\}$ .
  - Completeness: In the worst case, we have exactly  $\frac{n}{2}$  many squares and  $\frac{n}{2}$  many non-squares. Then  $P$  will only be able to send the roots  $y_i$  for  $i \in I$ , if  $V$  has chosen exactly the squares. Since there are  $\binom{n}{\frac{n}{2}}$  combinations the probability for  $V$  outputting 1 is  $\frac{1}{\binom{n}{\frac{n}{2}}}$ . Hence the completeness bound is  $c = \frac{1}{\binom{n}{\frac{n}{2}}}$ .
  - Soundness: There is no way for a cheating prover  $P^*$  to convince the honest verifier  $V$  of a false statement. If less than half of the  $x_i$  are squares, there will always be a  $j \in I$  such that  $x_j$  is not a square. The soundness bound is then  $s = 0$ .
- (b) A variant of the graph isomorphism proof. We try to use repetition to get better soundness.

- Statement:  $x = (G_1, G_2)$  where  $G_1 \approx G_2$ . Let  $n := |x|$ .
- Witness: An isomorphism  $\phi$  between  $G_1$  and  $G_2$ .
- The prover chooses random permutations  $\tau_1, \dots, \tau_n$  and computes  $H_k := \tau_k(G_2)$  for all  $k = 1, \dots, n$ . Then he sends  $H_1, \dots, H_n$  to the verifier.
- The verifier  $V$  chooses  $i \in \{1, 2\}$ .
- The prover  $P$  constructs isomorphisms  $\tilde{\phi}_1, \dots, \tilde{\phi}_n$  such that  $H_k = \tilde{\phi}_k(G_i)$  for all  $k$ . Then he sends  $\tilde{\phi}_1, \dots, \tilde{\phi}_n$  to  $V$ .
- The verifier  $V$  checks whether  $\tilde{\phi}_k(G_i) = H_k$  for all  $k$ .

**Solution.**

- For the witness relation we have in the case of graph isomorphism  $R := \{((G_1, G_2), \phi) \mid G_2 = \phi(G_1)\}$

- Completeness: similar to the original proof for graph isomorphism we have:
  - $i = 1$ :  $\tilde{\phi}_k(G_1) = \tau_k(\phi_k(G_1)) = \tau_k(G_2) = H_k$  for all  $k$
  - $i = 2$ :  $\tilde{\phi}_k(G_2) = \tau_k(G_2) = H_k$  for all  $k$

We have completeness bound  $c = 1$ .

- Soundness:
  - either  $H_k \not\approx G_1$  for all  $k$  or  $H_k \not\approx G_2$  for all  $k$
  - with probability  $\geq \frac{1}{2}$   $V$  sends  $i$  with  $H_k \not\approx G_i$  for all  $k$
  - there exist no  $\tilde{\phi}_k$  with  $H_k \not\approx G_i$  for all  $k$ , then  $V$  outputs 1

Hence we have soundness bound  $s = \frac{1}{2}$ .

(c) A trivial proof. Fix some language  $L$  in BPP. Let  $A$  be a polynomial-time algorithm such that on input  $A(x)$  it outputs whether  $x \in L$  and errs with probability at most  $2^{-|x|}$ .

- Statement:  $x \in L$ .
- Witness:  $w$  is any string.
- The prover sends *hello* to the verifier.<sup>2</sup>
- The verifier checks whether  $A(x)$  outputs yes.

### Solution.

- Here, a sensible witness relation is  $R := \{(x, w) | x \in L, w \in \{0, 1\}^*\}$
- Completeness: Since  $A(x)$  outputs whether  $x \in L$  (up to an error probability of  $2^{-|x|}$ ), the completeness bound is  $c = 1 - 2^{-|x|}$ .
- Soundness: Since for  $x \notin L$ ,  $A(x)$  outputs 0 with probability at least  $2^{-|x|}$ , the soundness bound is  $s = 2^{-|x|}$ .
- Note that this proof system is even statistical zero-knowledge (??): The simulator just has to simulate the prover honestly, this is possible without knowing the witness because the prover never uses the witness.

## Problem 2: Miscellaneous (11 Points)

(a) 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  $L \in \text{NP}$ , there is a relation  $R$  such that there is a proof system for  $R$  with perfect completeness and with soundness 0.)

---

<sup>2</sup>If German is your native language, you may alternatively investigate the case where the prover sends *hallo*.

**Solution.** We show that for every language in NP there exists an interactive proof system with completeness bound 1 and soundness bound 0.

- Let  $L$  be a language in NP. From the definition of NP it follows that there exists a relation  $R$ , such that  $x \in L \Leftrightarrow \exists w : (x, w) \in R$  and such that  $R$  can be decided by a deterministic polynomial-time Turing machine  $M$  and such that the  $|w|$  is polynomially-bounded in  $|x|$ .
  - In the interactive proof system the prover  $P$  sends the witness  $w$  to the verifier  $V$ .  $V$  then runs  $M(x, w)$  and outputs, what  $M$  outputs.
  - This proof system has completeness 1 and soundness 0.
- (b) Assume that  $\text{NP} \not\subseteq \text{BPP}$ . Show that there exists a relation  $R$  with  $L_R \in \text{NP}$  but such that  $R$  has no proof system with polynomial-time prover and verifier and with soundness  $\frac{1}{3}$  and completeness  $\frac{2}{3}$ .

**Hint:** Your relation should define witnesses in a such a way that knowing a witness is extremely unhelpful.

**Solution.**

- Let  $L \in \text{NP}$ .
  - We define  $R := \{(x, w) | x \in L, w \in \{0, 1\}^*\}$ : iff  $x \in L$ , then every string is a witness.
  - Then  $L_R = L \in \text{NP}$ .
  - Suppose there is a proof system for  $R$  with polynomial-time prover and verifier and with soundness  $\frac{1}{3}$  and completeness  $\frac{2}{3}$ .
  - Let  $M$  be the machine that on input  $x$  simulates the interaction of  $\langle P(x, w), V(x) \rangle$  for an arbitrary  $w$ .
  - If  $x \notin L$ , then  $\Pr[M(x) = 1] \leq \frac{1}{3}$ , otherwise  $\Pr[M(x) = 1] \geq \frac{2}{3}$ . Hence there is a polynomial-time probabilistic machine deciding  $L$ , hence  $L_R = L \in \text{BPP}$ .
  - Since this holds for all  $L \in \text{NP}$  this is a contradiction to the assumption  $\text{NP} \not\subseteq \text{BPP}$ .
  - Hence there is no proof system for  $R$  with polynomial-time prover and verifier and the given completeness and soundness bounds.
- (c) Let  $(P, V)$  be an interactive proof system for some relation  $R$  with soundness  $s$  and completeness  $c$ . Let  $(P^\circ, V^\circ)$  be the following proof system:
- On input  $(x, w)$ ,  $P^\circ$  executes  $P(x, w)$   $|x|$  times sequentially. (That is,  $P^\circ$  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  $x$ ,  $V^\circ$  executes  $V(x)$   $|x|$  times sequentially.  $V^\circ$  outputs 1 if and only if all invocations of  $V$  have output 1.

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

**Solution.**

- We will write  $M^n$  for the  $n$  times sequential composition of  $M$ . We prove by induction:
- Base case: For  $|x| = 1$  we have completeness  $c$  and soundness  $s$ .
- Induction hypothesis:  $(P^n, V^n)$  has completeness  $c^n$  and soundness  $s^n$ .
- Inductive step: Consider the case  $|x| = n+1$ . For completeness we have  $(\forall x \in L)$ :

$$\begin{aligned}
& \Pr[\langle P^\circ(x, w), V^\circ(x) \rangle = 1] \\
&= \Pr[\langle P^{n+1}(x, w), V^{n+1}(x) \rangle = 1] \\
&= \Pr[\langle (P(x, w), P^n(x, w)), (V(x), V^n(x)) \rangle = 1] \\
&= \Pr[\langle P(x, w), V(x) \rangle = 1] \cdot \Pr[\langle P^n(x, w), V^n(x) \rangle = 1] \\
&\geq c \cdot c^n = c^{n+1}
\end{aligned}$$

- Soundness: Here we deal with a malicious prover  $P^*$ . We assume we can decompose it into two malicious provers  $P_1^*$  and  $P_2^*$  running sequentially:  $P_1^*$  ends after sending the last message to the first invocation of  $V$  in  $V^\circ$  (we may assume, the number of rounds in the proof system  $(P, V)$  is known, so we know when the last message is sent). Both  $P_1^*$  and  $P_2^*$  output their internal state after termination.  $P_2^*$  gets as input the state  $s_1$  of  $P_1^*$  after its termination. We write  $(s, v) \leftarrow \langle P(\dots), V(\dots) \rangle$  for assigning to  $s$  the output of  $P$  and to  $v$  the output

of  $V$ . Then we have  $(\forall P^* \forall x \notin L)$ :

$$\begin{aligned}
& \Pr[\langle P^*, V^\circ(x) \rangle = 1] \\
&= \Pr[v_1 = v_2 = 1 : (s_1, v_1) \leftarrow \langle P_1^*, V(x) \rangle, v_2 \leftarrow \langle P_2^*(s_1), V^n(x) \rangle] \\
&= \sum_{s_0} \Pr[s_1 = s_0 \wedge v_1 = v_2 = 1 : (s_1, v_1) \leftarrow \langle P_1^*, V(x) \rangle, \\
&\quad (s_2, v_2) \leftarrow \langle P_2^*(s_0), V^n(x) \rangle] \\
&= \sum_{s_0} \Pr[v_2 = 1 : (s_2, v_2) \leftarrow \langle P_2^*(s_0), V^n(x) \rangle] \\
&\quad \cdot \Pr[s_1 = s_0 \wedge v_1 = 1 : (s_1, v_1) \leftarrow \langle P_1^*, V(x) \rangle] \\
&\leq \sum_{s_0} s^n \cdot \Pr[s_1 = s_0 \wedge v_1 = 1 : (s_1, v_1) \leftarrow \langle P_1^*, V(x) \rangle] \\
&= s^n \cdot \sum_{s_0} \Pr[s_1 = s_0 \wedge v_1 = 1 : (s_1, v_1) \leftarrow \langle P_1^*, V(x) \rangle] \\
&= s^n \cdot \Pr[v_1 = 1 : (s_1, v_1) \leftarrow \langle P_1^*, V(x) \rangle] \\
&\leq s^n \cdot s = s^{n+1}
\end{aligned}$$

$P_1^*$  and  $V$ ).

- (d) Explain why the proof system  $(P^\circ, V^\circ)$  from (c) is useless for proof systems with  $c = \frac{2}{3}$  and  $s = \frac{1}{3}$ . Explain (without a proof) how the method from (c) might be modified to deal with this case.

**Solution.**

- Using the proof system  $(P^\circ, V^\circ)$  for proof systems with  $c = \frac{2}{3}$  and  $s = \frac{1}{3}$  decreases not only the soundness bound, but also the completeness bound, since  $\lim_{n \rightarrow \infty} (\frac{2}{3})^n = \lim_{n \rightarrow \infty} (\frac{1}{3})^n = 0$ . With completeness  $\frac{2}{3}$  we would expect that with high probability approximately  $\frac{2}{3}$  of the  $V$ 's output 1, iff  $x \in L$  and with high probability approximately  $\frac{1}{3}$  of the  $V$ 's output 1, iff  $x \notin L$ . So we could modify  $V^\circ$ , such that it outputs 1, iff the majority of the  $V$ 's outputs 1, and 0 otherwise.

## Solution of Exercise Sheet 2

Out: Wed, Nov 5, 2008

Due: Fri, Nov 14, 2008, before noon

**Problem 1: Statistical distance**

Prove the following facts about the statistical distance SD (Definition 2 in the lecture notes).

- (a)  $SD(X; Y) = \frac{1}{2} \sum_a |\Pr[X = a] - \Pr[Y = a]|$  (assuming there are only countably many values  $a$  in the range of  $X$  and  $Y$ ).

**Hint:** First show that  $T := \{a : \Pr[X = a] > \Pr[Y = a]\}$  is a set that distinguishes best in the sense of Definition 2 in the lecture notes.

**Solution.** Let  $T' := \{a : \Pr[X = a] > \Pr[Y = a]\}$ . Fix a set  $T$ . Then

$$\begin{aligned} |\Pr[X \in T'] - \Pr[Y \in T']| &\geq \Pr[X \in T'] - \Pr[Y \in T'] \\ &\geq \sum_{a \in T'} \Pr[X = a] - \Pr[Y = a] \\ &\stackrel{(*)}{\geq} \sum_{a \in T} \Pr[X = a] - \Pr[Y = a] \\ &= \Pr[X \in T] - \Pr[Y \in T]. \end{aligned} \tag{1}$$

Here  $(*)$  holds because for all  $a \notin T'$ , we have  $\Pr[X = a] - \Pr[Y = a] \leq 0$ .

Since (1) holds for any set  $T$ , (1) also holds for  $T^c$ , the complement of  $T$ .

If  $\Pr[X \in T] - \Pr[Y \in T] \geq 0$ , from (1) we directly have  $|\Pr[X \in T'] - \Pr[Y \in T']| \geq \Pr[X \in T] - \Pr[Y \in T] = |\Pr[X \in T] - \Pr[Y \in T]|$ .

If  $\Pr[X \in T] - \Pr[Y \in T] \leq 0$ , we use the fact that (1) holds for all  $T$ , and hence also for  $T^c$ , the complement of  $T$ . We then have  $|\Pr[X \in T'] - \Pr[Y \in T']| \geq \Pr[X \in T^c] - \Pr[Y \in T^c] = -\Pr[X \in T] + \Pr[Y \in T] = |\Pr[X \in T] - \Pr[Y \in T]|$ .

Hence  $|\Pr[X \in T'] - \Pr[Y \in T']| = \max_T |\Pr[X \in T] - \Pr[Y \in T]| = SD(X; Y)$ .

Finally, we have

$$\begin{aligned}
2|\Pr[X \in T'] - \Pr[Y \in T']| &= \underbrace{\Pr[X \in T'] - \Pr[Y \in T']}_{\geq 0} + \underbrace{\Pr[Y \in T'^c] - \Pr[X \in T'^c]}_{\geq 0} \\
&= \sum_{a \in T'} \Pr[X = a] - \Pr[Y = a] + \sum_{a \in T'^c} \Pr[Y = a] - \Pr[X = a] \\
&= \sum_{a \in T'} |\Pr[X = a] - \Pr[Y = a]| + \sum_{a \in T'^c} |\Pr[Y = a] - \Pr[X = a]| \\
&= \sum_a |\Pr[X = a] - \Pr[Y = a]|.
\end{aligned}$$

Hence  $\text{SD}(X; Y) = \frac{1}{2} \sum_a |\Pr[X = a] - \Pr[Y = a]|$ .

(b)  $\text{SD}(f(X); f(Y)) \leq \text{SD}(X; Y)$ .

**Solution.**

$$\begin{aligned}
\text{SD}(f(X); f(Y)) &= \max_T |\Pr[f(X) \in T] - \Pr[f(Y) \in T]| \\
&= \max_T |\Pr[X \in f^{-1}(T)] - \Pr[Y \in f^{-1}(T)]| \\
&\leq \max_{\tilde{T}} |\Pr[X \in \tilde{T}] - \Pr[Y \in \tilde{T}]| \\
&= \text{SD}(X; Y).
\end{aligned}$$

(c)  $\text{SD}(f(X); f(Y)) = \text{SD}(X; Y)$  if  $f$  is injective.

**Solution.**

$$\text{SD}(X; Y) = \text{SD}(f^{-1}(f(X)); f^{-1}(f(Y))) \stackrel{(b)}{\leq} \text{SD}(f(X); f(Y)) \stackrel{(b)}{\leq} \text{SD}(X; Y).$$

(d) If  $Z$  is stochastically independent of  $X$  and of  $Y$ , then  $\text{SD}((X, Z); (Y, Z)) = \text{SD}(X; Y)$  (independent additional information does not change the statistical distance).

**Hint:** If  $A$  and  $B$  are stochastically independent, then  $\Pr[(A, B) = (a, b)] = \Pr[A = a] \cdot \Pr[B = b]$ . Use (a).

**Solution.**

$$\begin{aligned} \text{SD}((X, Z); (Y, Z)) &\stackrel{(a)}{=} \frac{1}{2} \cdot \sum_{a,b} |\Pr[(X, Z) = (a, b)] - \Pr[(Y, Z) = (a, b)]| \\ &= \frac{1}{2} \sum_{a,b} |\Pr[X = a] \cdot \Pr[Z = b] - \Pr[Y = a] \cdot \Pr[Z = b]| \\ &= \frac{1}{2} \sum_a \left( \sum_b |\Pr[X = a] - \Pr[Y = a]| \cdot |\Pr[Z = b]| \right) \\ &= \frac{1}{2} \sum_a \left( |\Pr[X = a] - \Pr[Y = a]| \cdot \sum_b |\Pr[Z = b]| \right) \\ &= \frac{1}{2} \sum_a |\Pr[X = a] - \Pr[Y = a]| \\ &= \text{SD}(X; Y). \end{aligned}$$

- (e) Let  $X$  be uniformly distributed over  $\{0, 1\}$ , and let  $Y$  be uniformly distributed over  $\{1, 2\}$ . What is  $\text{SD}(X; Y)$ ?

**Solution.**

$$\begin{aligned} \text{SD}(X; Y) &= \frac{1}{2} \sum_{a \in \{0, 1, 2\}} |\Pr[X = a] - \Pr[Y = a]| \\ &= \frac{1}{2} \left( \left| \frac{1}{2} - 0 \right| + \left| \frac{1}{2} - \frac{1}{2} \right| + \left| 0 - \frac{1}{2} \right| \right) \\ &= \frac{1}{2} \end{aligned}$$

- (f) Let  $X$  be uniformly distributed over  $\{0, 1\}^n$ , and let  $Y$  be  $0^n$  with probability 1. What is  $\text{SD}(X; Y)$ ?

**Solution.**

$$\begin{aligned}
\text{SD}(X;Y) &= \frac{1}{2} \sum_{a \in \{0,1\}^n} |\Pr[X = a] - \Pr[Y = a]| \\
&= \frac{1}{2} \left( \left| \frac{1}{2^n} - 1 \right| + \sum_{\substack{a \in \{0,1\}^n \\ a \neq 0^n}} |\Pr[X = a] - \Pr[Y = a]| \right) \\
&= \frac{1}{2} \left( \left| \frac{1}{2^n} - 1 \right| + \sum_{\substack{a \in \{0,1\}^n \\ a \neq 0^n}} \frac{1}{2^n} \right) \\
&= \frac{1}{2} \left( 1 - \frac{1}{2^n} + (2^n - 1) \cdot \frac{1}{2^n} \right) \\
&= \frac{1}{2} \left( 1 - \frac{1}{2^n} + 1 - \frac{1}{2^n} \right) \\
&= 1 - \frac{1}{2^n}
\end{aligned}$$

## Problem 2: Graph isomorphism proof is ZK

**Quantum of solace:** This problem looks more frightening than it is. The largest part of the work is probably to read the problem description.

**Notation:** Let  $P$  be the prover of the graph isomorphism proof as described in the lecture (see also Definition 4 in the lecture notes). Let  $V^*$  be a malicious verifier,  $S_1$  as described in the lecture (the simulator for graph isomorphism that tries only once to guess the message  $i^*$  sent by  $V^*$  and fails if  $i \neq i^*$  where  $i$  is the guess of  $S_1$ ). Let  $\text{perm}_n$  be the set of all permutations on  $\{1, \dots, n\}$  (i.e., on the vertices of a graph of size  $n$ ). Let  $n$  be the number of vertices of the graph  $G_1$  and assume that  $\phi(G_1) = G_2$ .

The goal of this exercise is to show that

$$\langle P(G_1, G_2, \phi), V^*(G_1, G_2) \rangle \quad \text{and} \quad S_1(G_1, G_2)|_{\text{success}} \quad (2)$$

have the same distribution. Here  $S_1(G_1, G_2)|_{\text{success}}$  denotes the distribution of the output of  $S_1$  under the condition that  $S_1$  does not fail (i.e., under the condition that  $i = i^*$  in the simulation).

Consider the following games:<sup>1</sup>

$$i \stackrel{R}{\leftarrow} \{1, 2\}, \quad \tilde{\phi} \stackrel{R}{\leftarrow} \text{perm}_n, \quad H := \tilde{\phi}(G_i), \\ i^* \leftarrow V^*(H), \quad \text{out} \leftarrow V^*(\tilde{\phi}), \quad \text{if } i = i^* \text{ return } \text{out} \text{ else return } \perp \quad (3)$$

$$i \stackrel{R}{\leftarrow} \{1, 2\}, \quad \tau \stackrel{R}{\leftarrow} \text{perm}_n, \quad \text{if } i = 1 \text{ then } \tilde{\phi} := \tau \circ \phi \text{ else } \tilde{\phi} := \tau, \quad H := \tilde{\phi}(G_i), \\ i^* \leftarrow V^*(H), \quad \text{out} \leftarrow V^*(\tilde{\phi}), \quad \text{if } i = i^* \text{ return } \text{out} \text{ else return } \perp \quad (4)$$

$$i \stackrel{R}{\leftarrow} \{1, 2\}, \quad \tau \stackrel{R}{\leftarrow} \text{perm}_n, \quad \text{if } i = 1 \text{ then } \tilde{\phi} := \tau \circ \phi \text{ else } \tilde{\phi} := \tau, \quad \mathbf{H := \tau(G_2)}, \\ i^* \leftarrow V^*(H), \quad \text{out} \leftarrow V^*(\tilde{\phi}), \quad \text{if } i = i^* \text{ return } \text{out} \text{ else return } \perp \quad (5)$$

$$i \stackrel{R}{\leftarrow} \{1, 2\}, \quad \tau \stackrel{R}{\leftarrow} \text{perm}_n, \quad H := \tau(G_2), \quad i^* \leftarrow V^*(H), \\ \text{if } \mathbf{i^* = 1} \text{ then } \tilde{\phi} := \tau \circ \phi \text{ else } \tilde{\phi} := \tau, \\ \text{out} \leftarrow V^*(\tilde{\phi}), \quad \text{if } i = i^* \text{ return } \text{out} \text{ else return } \perp \quad (6)$$

$$\tau \stackrel{R}{\leftarrow} \text{perm}_n, \quad H := \tau(G_2), \quad i^* \leftarrow V^*(H), \quad \text{if } i^* = 1 \text{ then } \tilde{\phi} := \tau \circ \phi \text{ else } \tilde{\phi} := \tau, \\ \text{out} \leftarrow V^*(\tilde{\phi}), \quad \mathbf{i \stackrel{R}{\leftarrow} \{1, 2\}}, \quad \text{if } i = i^* \text{ return } \text{out} \text{ else return } \perp \quad (7)$$

$$\tau \stackrel{R}{\leftarrow} \text{perm}_n, \quad H := \tau(G_2), \quad i^* \leftarrow V^*(H), \quad \text{if } i^* = 1 \text{ then } \tilde{\phi} := \tau \circ \phi \text{ else } \tilde{\phi} := \tau, \\ \text{out} \leftarrow V^*(\tilde{\phi}), \quad \mathbf{\text{with probability } \frac{1}{2} \text{ return } \text{out} \text{ else return } \perp} \quad (8)$$

(The parts that changed between two lines are highlighted in boldface.)

Here we wrote  $x \stackrel{R}{\leftarrow} Y$  for denoting that  $x$  is uniformly randomly chosen from the set  $Y$ , and  $x \leftarrow A$  to denote that  $x$  is chosen according to the algorithm  $A$ . We wrote  $i^* \leftarrow V^*(H)$  for the first invocation of the verifier  $V^*$  (with incoming message  $H$  and outgoing message  $i^*$ ), and  $\text{out} \leftarrow V^*(\tilde{\phi})$  for the second invocation of  $V^*$  (with incoming message  $\tilde{\phi}$  and final output  $\text{out}$ ).

Note that each of these games induces a probability distribution on the finally returned value (which may be  $\text{out}$  or  $\perp$ ). The distribution of the return value in game (3) has the same distribution as  $S_1(G_1, G_2)$ . The distribution of the return value in game (8) under the condition that it is not  $\perp$  has the same distribution as  $\langle P(G_1, G_2, \phi), V^*(G_1, G_2) \rangle$ . Hence, if we knew that the return values of (3) and (8) have the same distribution, (2) would follow.

**Your task:** For each  $i \in \{3, \dots, 7\}$ , show that the games  $(i)$  and  $(i + 1)$  are equivalent in the sense that the distribution of their outputs is the same.

**Solution.** We assume, that  $V^*(H)$  returns 1 or 2. We show the equivalences:

- (3)  $\leftrightarrow$  (4): We choose a random permutation  $\tau$ . For  $i = 1$  we assign  $\tau \circ \phi$  to  $\tilde{\phi}$ , which does not change the distribution of permutations, because of group properties and

---

<sup>1</sup>In cryptography, the word *game* often denotes some probabilistic experiment described by an algorithm, as in the equations below.

the fact, that  $\tau$  is randomly chosen. For  $i = 2$  we simply assign  $\tau$  to  $\tilde{\phi}$ , which yields the same distribution as before.

- (4)  $\leftrightarrow$  (5): For  $i = 1$  we have  $\tilde{\phi} = \tau \circ \phi, H = \tilde{\phi}(G_i) \Rightarrow \tilde{\phi}(G_i) = \tau(\phi(G_1)) = \tau(G_2)$ . For  $i = 2$  we have  $\tilde{\phi} = \tau, H = \tilde{\phi}(G_i) \Rightarrow \tilde{\phi}(G_i) = \tau(G_2)$ .
- (5)  $\leftrightarrow$  (6): Computing  $H := \tau(G_2)$  and the first invocation of  $V^*, i^* \leftarrow V^*(H)$ , do not depend on the choice of  $i$  or  $\tilde{\phi}$ , hence we can change the order of those steps. For the case  $i = i^*$  nothing has to be shown. For  $i \neq i^*$  we return  $\perp$  in both games.
- (6)  $\leftrightarrow$  (7): Since  $\tau \stackrel{R}{\leftarrow} \text{perm}_n, H := \tau(G_2), i^* \leftarrow V^*(H)$ , if  $i^* = 1$  then  $\tilde{\phi} := \tau \circ \phi$  else  $\tilde{\phi} := \tau, out \leftarrow V^*(\tilde{\phi})$  does not depend on the choice of  $i \stackrel{R}{\leftarrow} \{1, 2\}$ , we can change the order of those steps as well.
- (7)  $\leftrightarrow$  (8): Since  $i \stackrel{R}{\leftarrow} \{1, 2\}$  is chosen uniformly random, the probability of outputting  $out$  is  $\Pr[i = i^* : \dots, i^* \leftarrow V^*(H), \dots, i \stackrel{R}{\leftarrow} \{1, 2\}] = \frac{1}{2}$ .

### Solution of Exercise Sheet 3

Out: Fri, Nov 14, 2008

Due: Fri, Nov 21, 2008, before noon

#### Problem 1: Statistical distance (5=2+2+1 points)

Let  $X$  and  $Y$  be random variables. Let  $F$  be a probabilistic (i.e., randomized) function.

You may assume that the ranges of all random variables/functions are countable.

(a) Show that

$$\text{SD}(F(X); F(Y)) \leq \text{SD}(X; Y)$$

**Hint:** Use the fact that any probabilistic function  $F(\cdot)$  can be expressed as  $f(\cdot, R)$  where  $f$  is deterministic and  $R$  is a random variable stochastically independent from all other random variables (e.g.,  $F(X)$  and  $F(Y)$  are the same as  $f(X, R)$  and  $f(Y, R)$  where  $R$  is stochastically independent from  $X$  and  $Y$ ).

**Solution.**

$$\begin{aligned} \text{SD}(F(X); F(Y)) &= \text{SD}(f(X, R); f(Y, R)) \\ &\leq \text{SD}((X, R); (Y, R)) \\ &= \text{SD}(X; Y) \end{aligned}$$

Here we used Exercise Sheet 2, Problem 1(a) in the second and 1(d) in the third step.

(b) Show that

$$\text{SD}(X_Z; Y_Z) = \sum_z \Pr[Z = z] \text{SD}(X_z; Y_z)$$

By  $X_Z$  we denote the random variable resulting from first sampling  $z$  according to  $Z$  and then sampling  $x$  from  $X_z$ .

**Hint:** Use the representation of SD from Exercise Sheet 2, Problem 1(a).

**Solution.**

$$\begin{aligned}
\text{SD}(X_Z; Y_Z) &\stackrel{(*)}{=} \frac{1}{2} \sum_a |\Pr[X_Z = a] - \Pr[Y_Z = a]| \\
&= \frac{1}{2} \sum_a \left| \sum_z \Pr[X_z = a \text{ and } Z = z] - \sum_z \Pr[Y_z = a \text{ and } Z = z] \right| \\
&= \frac{1}{2} \sum_a \left| \sum_z \Pr[X_z = a] \Pr[Z = z] - \sum_z \Pr[Y_z = a] \Pr[Z = z] \right| \\
&= \sum_z \left( \Pr[Z = z] \frac{1}{2} \sum_a |\Pr[X_z = a] - \Pr[Y_z = a]| \right) \\
&= \sum_z \Pr[Z = z] \text{SD}(X_z; Y_z)
\end{aligned}$$

Here  $(*)$  uses Exercise Sheet 2, Problem 1(a).

(c) Show the following. If  $\text{SD}(X_z; Y_z) \leq \varepsilon$  for all  $z$ , then  $\text{SD}(X_Z; Y_Z) \leq \varepsilon$ .

**Solution.**

$$\text{SD}(X_Z; Y_Z) = \sum_z \Pr[Z = z] \text{SD}(X_z; Y_z) \leq \sum_z \Pr[Z = z] \varepsilon = \varepsilon$$

## Problem 2: On auxiliary input (7 Points)

Assume that an unkeyed family of collision-resistant hash functions exists. In the context of this problem, by an unkeyed family of collision-resistant hash functions we denote a sequence  $(h_n)_{n \in \mathbb{N}}$  of functions  $h_n : \{0, 1\}^* \rightarrow \{0, 1\}^n$  with the following property: For all polynomial-time (uniform!) machines  $A$ , the following probability is negligible in  $n$ :  $\Pr[h_n(x) = h_n(x') \wedge x \neq x' : (x, x') \leftarrow A(1^n)]$  (i.e.,  $A$  find a collision  $(x, x')$  for  $h_n$  only with negligible probability).

Under this assumption, show that there is a proof system  $(P, V)$  that is statistical zero-knowledge without auxiliary input (Definition 3 in the lecture notes), but not statistical zero-knowledge with auxiliary input (Definition 6 in the lecture notes).

For one bonus point, additionally ensure that the proof system  $(P, V)$  has perfect soundness (soundness bound 0) and perfect completeness (completeness bound 1).

**Hint:** Let the first step of the protocol be the following:  $V$  sends a pair  $(x, x')$ . (An honest  $V$  would send random values.) The prover  $P$  checks whether  $x \neq x'$  and  $h_n(x) = h_n(x')$ . The behaviour of the prover then depends on whether this check succeeds.

**Solution.** Let  $R = \{(x, w) | x \in \{0, 1\}^*, w \in \{0, 1\}^*\}$ . Let  $n = |x|$ .

- On input  $(x, w)$  the prover  $P$  checks after receiving  $(y, y')$ , whether  $(y, y')$  is a collision for  $h_n$  ( $y \neq y'$  and  $h_n(y) = h_n(y')$ ). If  $(y, y')$  is a collision  $P$  sends a witness  $w$ , otherwise it sends 1 indicating that no witness has been sent.
- The verifier  $V$  sends a randomly chosen pair  $(y, y')$  to  $P$ . After receiving the witness  $w$  or 1 it outputs 1.

The proof system has completeness bound 1 (since  $V$  always outputs 1) and soundness bound 0 (since  $\forall x : x \in L_R$ ).

The proof system  $(P, V)$  is zero-knowledge without auxiliary input: The interaction between  $P$  and  $V^*$  can be simulated by  $S$ . After receiving  $(y, y')$  from  $V^*$   $S$  sends 1 to  $V^*$  and outputs what  $V^*$  outputs.  $V^*$  can obtain a collision for  $h_n$  only with negligible probability, hence  $\text{SD}(\langle P(x, w), V^*(x) \rangle; S(x))$  is negligible.

However the proof system  $(P, V)$  is not zero-knowledge with auxiliary input: there is an auxiliary input  $z = (z, z')$  which is a collision for  $h_n$  (since collisions for  $h_n : \{0, 1\}^* \rightarrow \{0, 1\}^n$  must exist). A malicious verifier  $V^*$  will send  $z$  to  $P$ , expecting to receive the witness  $w$  for  $x$  and output  $w$ . But for witnesses  $w_1$  and  $w_2$  we then have

$$\text{SD}(\langle P(x, w_1), V^*(x, z) \rangle; \langle P(x, w_2), V^*(x, z) \rangle) = 1$$

and by the triangle inequality

$$\begin{aligned} \text{SD}(\langle P(x, w_1), V^*(x, z) \rangle; S(x, z)) + \text{SD}(\langle P(x, w_2), V^*(x, z) \rangle; S(x, z)) \\ \geq \text{SD}(\langle P(x, w_1), V^*(x, z) \rangle; \langle P(x, w_2), V^*(x, z) \rangle) = 1. \end{aligned}$$

We then have for some  $w \in \{w_1, w_2\}$ :

$$\text{SD}(\langle P(x, w), V^*(x, z) \rangle; S(x, z)) \geq \frac{1}{2}.$$

### Problem 3: Variants of the ZK definition (8=4+4 points)

- (a) Given two machines  $P, V$ , let  $\text{view}\langle P, V \rangle$  denote the view of  $V$  in an interaction with  $P$ . By view we mean the list of all messages sent and received by  $V$ , together with all internal states of  $V$  during the interaction, together with all random bits used by  $V$  (its random tape), and its inputs and output.

Consider the following definition.

**Definition 1 (Statistical zero-knowledge with respect to the view)** *A pair  $(P, V)$  of interactive machines is called statistically zero-knowledge with respect to the view for a relation  $R$  if for any machine  $V^*$  polynomial-time in its first input, there exists an algorithm  $S$  polynomial-time in its first input, and a negligible function  $\mu$  such that for all  $(x, w) \in R$  and all bitstrings  $z \in \{0, 1\}^*$  (the auxiliary input) we have that*

$$\text{SD}(\text{view}\langle P(x, w), V^*(x, z) \rangle; S(x, z)) \leq \mu(|x|).$$

The only change with respect to Definition 6 in the lecture notes is that we used  $\text{view}\langle P(x, w), V^*(x, z) \rangle$  instead of  $\langle P(x, w), V^*(x, z) \rangle$ . That is, the simulator does not only have to simulate the output of  $V^*$ , but everything that was observed by  $V^*$  in the interaction.<sup>1</sup> Prove that  $(P, V)$  is statistically zero-knowledge with respect to the view if and only if it is statistically zero-knowledge with auxiliary input.

**Solution.** To show: zero-knowledge with respect to view  $\Leftrightarrow$  zero-knowledge with auxiliary input.

“ $\Rightarrow$ ” Given a simulator  $S$  which simulates  $\text{view}\langle P(x, w), V^*(x, z) \rangle$  we can construct a simulator  $S'$  which simulates  $\langle P(x, w), V^*(x, z) \rangle$ .  $S'$  behaves as  $S$ , but instead of  $\text{view}\langle P(x, w), V^*(x, z) \rangle$  it only outputs the output of  $V^*$  (i.e.,  $\langle P(x, w), V^*(x, z) \rangle$ ), which is included in the view.

“ $\Leftarrow$ ” Given  $V^*$  we construct  $V^{**}$ :  $V^{**}$  behaves as  $V^*$ , except at the end of the interaction with  $P$  it outputs the view. So we have

$$\text{view}\langle P(x, w), V^*(x, z) \rangle = \langle P(x, w), V^{**}(x, z) \rangle.$$

By definition of zero-knowledge with auxiliary input there exists a simulator  $S^{**}$  simulating the interaction between  $P$  and  $V^{**}$ :

$$\text{SD}(\langle P(x, w), V^{**}(x, z) \rangle; S^{**}(x, z)) \leq \mu(|x|).$$

It follows, that  $S^{**}$  is the simulator which can simulate the view of the interaction between  $P$  and  $V^*$ :

$$\text{SD}(\text{view}\langle P(x, w), V^*(x, z) \rangle; S^{**}(x, z)) \leq \mu(|x|).$$

(b) Also consider the following definition:

**Definition 2 (Statistical zero-knowledge with deterministic verifiers)** A pair  $(P, V)$  of interactive machines is called statistically zero-knowledge with deterministic verifiers for a relation  $R$  if for any deterministic machine  $V^*$  polynomial-time in its first input, there exists an algorithm  $S$  polynomial-time in its first input, and a negligible function  $\mu$  such that for all  $(x, w) \in R$  and all bitstrings  $z \in \{0, 1\}^*$  (the auxiliary input) we have that

$$\text{SD}(\langle P(x, w), V^*(x, z) \rangle; S(x, z)) \leq \mu(|x|).$$

The only difference to Definition 6 in the lecture notes is that we require  $V^*$  to be deterministic.

Prove that  $(P, V)$  is statistically zero-knowledge with deterministic verifiers if and only if it is statistically zero-knowledge with auxiliary input.

**Hint:** To construct a deterministic verifier from a probabilistic verifier  $V^*$ , include the randomness needed by  $V^*$  in its auxiliary input. You will find Problem 1(c) useful.

---

<sup>1</sup>In the proof that the graph-isomorphism proof is zero-knowledge, the simulator does this anyway because it simulates  $V^*$  directly. But simulators for other proof systems might proceed differently.

**Solution.** To show: zero-knowledge with deterministic verifier  $\Leftrightarrow$  zero-knowledge with auxiliary input.

“ $\Leftarrow$ ” A deterministic verifier is just a special case of a probabilistic verifier.

“ $\Rightarrow$ ” Including the randomness needed by  $V^*$  in the auxiliary input we get:

$$\text{SD}(\langle P(x, w), V_d^*(x, (z, r)) \rangle; S_d(x, (z, r))) \leq \mu(|x|),$$

which holds for all  $r$ . Here  $V_d^*$  denotes a deterministic verifier which gets random bits  $r$  in its auxiliary input  $(z, r)$  and behaves as a probabilistic verifier  $V^*$  making its choices according to  $r$ .  $S_d$  denotes the corresponding simulator. By Problem 1(c) we then have

$$\text{SD}(\langle P(x, w), V_d^*(x, (z, R)) \rangle; S_d(x, (z, R))) \leq \mu(|x|),$$

where  $R$  is a random variable for the random tape of the probabilistic verifier  $V^*$ . Now the deterministic verifier  $V_d^*(x, (z, R))$  with auxiliary input  $(z, R)$  behaves as the probabilistic verifier  $V^*(x, z)$  and hence

$$\text{SD}(\langle P(x, w), V^*(x, z) \rangle; S(x, z)) \leq \mu(|x|).$$

## Solution of Exercise Sheet 4

Out: Fri, Nov 21, 2008

Due: Fri, Nov 28, 2008, before noon

## Problem 1: Constructing commitment schemes

**Definition 1 (Hard-core bit)** Let families of functions  $f_n : \{0, 1\}^{\ell(n)} \rightarrow \{0, 1\}^{\ell(n)}$  and  $b_n : \{0, 1\}^{\ell(n)} \rightarrow \{0, 1\}$  be given. We call  $b_n$  a hard-core bit of  $f_n$  if the following holds: For any nonuniform polynomial-time algorithm  $A$ , the following is negligible in  $n$ :

$$\left| \Pr[b' = b_n(x) : x \xleftarrow{R} \{0, 1\}^{\ell(n)}, b' \leftarrow A(1^n, f_n(x))] - \frac{1}{2} \right|.$$

(Intuitively: A computationally-bounded adversary does not learn anything about  $b_n(x)$  from  $f_n(x)$ .)

**Definition 2 (One-way function)** A family of functions  $f_n : \{0, 1\}^{\ell(n)} \rightarrow \{0, 1\}^{\ell(n)}$  is a one-way function if for any nonuniform polynomial-time algorithm  $A$ , the following probability is negligible in  $n$ :

$$\Pr[f_n(x') = f_n(x) : x \xleftarrow{R} \{0, 1\}^{\ell(n)}, x' \leftarrow A(1^n, f_n(x))]$$

(Intuitively: A computationally-bounded adversary cannot find preimages of  $f_n$ .)

If  $f_n$  is a permutation for all  $n$ , we call the family  $f_n$  a one-way permutation.

**Theorem 1 (Goldreich-Levin)** Let  $f_n$  be a one-way function. For  $x, y \in \{0, 1\}^{\ell(n)}$ , let  $\langle x, y \rangle$  be defined as  $x_1y_1 \oplus \dots \oplus x_ny_n$  (the scalar product over  $\text{GF}(2)$ ). For  $x, r \in \{0, 1\}^{\ell(n)}$ , let  $g_n(x||r) := f_n(x)||r$ . Let  $b_n(x||r) := \langle x, r \rangle$ . Then  $b_n$  is a hard-core bit of  $g_n$ .

Assume that a one-way permutation  $f_n$  exists. Use the functions  $g_n$  and  $b_n$  from Theorem 1 to construct a commitment scheme with message space  $M_n = \{0, 1\}$  (a bit commitment).

Show that your scheme is perfectly binding and computationally hiding.

**Hint:** When committing to some  $m \in \{0, 1\}$ , the value  $b_n(x||r) \oplus m$  might occur as part of your commit message (but of course, it is not the whole commit message).

**Solution.** We construct a commitment scheme for  $m \in \{0, 1\}$  in the following way:

- $\text{Com}(1^n, m) = (c, u)$  with commit message  $c = (b_n(x||r) \oplus m, g_n(x||r))$  and unveil information  $u = (x, r)$  for uniformly random  $x, r \in \{0, 1\}^{\ell(n)}$ .
- $\text{Verify}(1^n, m, (c_1, c_2), (u_1, u_2)) = (g_n(u_1||u_2) = c_2 \wedge \langle u_1, u_2 \rangle \oplus m = c_1)$ .

Here  $g_n(x||r) := f_n(x)||r$ ,  $f_n$  is a one-way permutation and  $b_n(x||r) := \langle x, r \rangle$  a hard-core bit of  $g_n$  (according to Theorem 1).

Properties of this bit commitment scheme:

- Correctness: We have  $M_n = \{0, 1\}$  for all  $n \in \mathbb{N}$ . For  $m \in \{0, 1\}$  we have  $(b_n(x||r) \oplus m, g_n(x||r)), (x, r) = Com(1^n, m)$  and

$$\begin{aligned} & Verify(1^n, m, (b_n(x||r) \oplus m, g_n(x||r)), (x, r)) \\ &= (g_n(x||r) = g_n(x||r) \wedge \langle x, r \rangle \oplus m = b_n(x||r) \oplus m) \\ &= (g_n(x||r) = g_n(x||r) \wedge \langle x, r \rangle \oplus m = \langle x, r \rangle \oplus m) \\ &= 1 \end{aligned}$$

For  $m \notin \{0, 1\}$   $b_n(x, r) \oplus m$  is not defined.

- Perfectly binding:  $x$  and  $r$  are determined by  $g_n(x||r) := f_n(x)||r$ .  $r$  is determined by the second part of  $g_n(x||r)$ .  $x$  is determined by the first half of  $g_n(x||r)$ , since  $f_n$  is a one-way permutation. Hence there is no  $x' \neq x$  such that  $f_n(x) = f_n(x')$ .
- Computationally hiding: If a nonuniform polynomial time algorithm  $A$  could distinguish a commitment to  $m = 0$  ( $b_n(x||r) \oplus 0$ ) from a commitment to  $m = 1$  ( $b_n(x||r) \oplus 1$ ) better than by guessing with  $\frac{1}{2} +$  non-negligible probability, then  $A$  can guess the hard-core bit  $b_n(x||r)$  with  $\frac{1}{2} +$  non-negligible probability.

## Problem 2: Graph-3-colouring

Let  $(P, V)$  denote the proof system of graph-3-colouring described in the lecture (Definition 11 in the lecture notes).

- (a) Show that  $(P, V)$  is a proof system with perfect completeness and soundness bound  $1 - \frac{1}{|E|}$  where  $E$  is the set of the edges of the graph  $G$ .

### Solution.

- Completeness bound 1: Knowing a graph-3-colouring  $\gamma$  for  $G$ , no matter what edge  $(v_1, v_2) \in E$   $V$  will pick,  $P$  will always return a valid colouring for  $v_1$  and  $v_2$ .
- Soundness bound  $1 - \frac{1}{|E|}$ : The commitments to the colours  $(c_v)_v$  determine a colouring  $\tilde{\gamma}$  of the graph. If there is no graph-3-colouring for  $G$ , there must be at least one edge  $(v_1, v_2) \in E$  such that  $\tilde{\gamma}(v_1) = \tilde{\gamma}(v_2)$ . The probability of  $V$  to pick an edge  $(v_1, v_2) \in E$  with an invalid colouring is at least  $\frac{1}{|E|}$ . Since the commitment scheme is perfectly binding  $P^*$  is not able to convince  $V$  that  $v_1$  and  $v_2$  have different colours. Hence  $V$  will not be convinced with a probability of at least  $\frac{1}{|E|}$ . Then the soundness bound is  $1 - \frac{1}{|E|}$ .

- (b) Show that  $(P, V)$  is *not* statistically zero-knowledge.<sup>1</sup>

<sup>1</sup>With or without auxiliary input; I do not care.

**Hint:** First, construct a sequence of graphs  $G_n$  with  $|G_n| \rightarrow \infty$  such that each graph has at least two 3-colourings  $\gamma_n^1, \gamma_n^2$  such that  $\gamma_n^1 \neq \rho \circ \gamma_n^2$  for all permutations  $\rho$  on  $\{R, G, B\}$ . Then compute  $\text{SD}(\langle P(G_n, \gamma_n^1), V^*(G_n) \rangle; \langle P(G_n, \gamma_n^2), V^*(G_n) \rangle)$  where  $V^*$  is the verifier that behaves like the honest verifier  $V$  but outputs the messages received from the prover.

**Solution.** Let  $G_n := (V_n, E_n)$  with  $V_n = (v_1, \dots, v_{n+2})$  and  $(v_i, v_{i+1}) \in E_n$  for all  $1 \leq i \leq n+1$ .  $G_n$  is a graph with  $n+2$  vertices, the first vertex has only a “right” neighbour vertex, the last vertex only a “left” neighbour vertex and all other vertices have a left and a right neighbour vertex.

Each  $G_n$  has at least two 3-colourings  $\gamma_n^1, \gamma_n^2$ , with  $\gamma_n^1 \neq \rho \circ \gamma_n^2$  for all  $\rho \in \text{perm}\{R, G, B\}$  (beginning from the left colour the vertices  $R, G, B, \dots$  and  $B, G, B, \dots$ ).

For  $V^*$  behaving like  $V$  but outputting the messages received from the prover, we have that

$$\text{SD}(\langle P(G_n, \gamma_n^1), V^*(G_n) \rangle; \langle P(G_n, \gamma_n^2), V^*(G_n) \rangle) = 1.$$

This is because the commitment scheme is perfectly binding for any simulator and one can construct a function, that extracts the colouring of the graph  $\gamma_1, \gamma_2$  given the commitments to the colours. Then one can always distinguish the different colourings. For any simulator  $S$ , by the triangle inequality we get

$$\begin{aligned} \text{SD}(\langle P(G_n, \gamma_n^1), V^*(G_n) \rangle; S(G_n)) + \text{SD}(\langle P(G_n, \gamma_n^2), V^*(G_n) \rangle; S(G_n)) \\ \geq \text{SD}(\langle P(G_n, \gamma_n^1), V^*(G_n) \rangle; \langle P(G_n, \gamma_n^2), V^*(G_n) \rangle) = 1. \end{aligned}$$

Then there exists  $\gamma \in \{\gamma_n^1, \gamma_n^2\}$  such that

$$\text{SD}(\langle P(G_n, \gamma), V^*(G_n) \rangle; S(G_n)) \geq \frac{1}{2},$$

which means, that the proof system for graph-3-colouring is not statistically zero-knowledge.

### Problem 3: Proof of Quadratic Residuosity

Consider the following relation  $R$ :

$$((r, N), s) \in R \iff s^2 = r \pmod N \text{ and } r \text{ is invertible modulo } N$$

(Note that deciding whether  $(r, N) \in L_R$ , i.e., whether  $r$  is a quadratic residue modulo  $N$ , is considered hard. But you do not need this fact for solving this problem.)

The following is a proof system  $(P, V)$  for  $R$ :

- Statement:  $(r, N)$ . Witness  $s$ .

- $P$  picks  $g \in \{0, \dots, N-1\}$  such that  $g$  is invertible modulo  $N$  and sends  $h := g^2 \bmod N$  to  $V$ .
- $V$  checks whether  $h$  and  $r$  are invertible modulo  $N$  and then  $V$  picks  $i \in \{1, 2\}$  and sends  $i$  to  $P$ .
- If  $i = 1$ ,  $P$  sends  $c_1 := g$ , otherwise  $P$  sends  $c_2 := gs \bmod n$  to  $V$ .
- If  $i = 1$ ,  $V$  checks whether  $c_1^2 \equiv h \bmod N$ . If  $i = 2$ ,  $V$  checks whether  $c_2^2 \equiv hr \bmod N$ .

(a) Prove that  $(P, V)$  has perfect completeness.

**Solution.**

- $V$  checks whether  $h$  and  $r$  are invertible modulo  $N$ : since  $P$  picked  $g$  to be invertible modulo  $N$ ,  $h := g^2$  is also invertible modulo  $N$ ,  $r$  is invertible  $N$  by assumption.
- $V$  picks  $i \in \{1, 2\}$  and sends  $i$  to  $P$ :
  - case  $i = 1$ :  $P$  sends  $c_1 := g$ ,  $V$  then checks that indeed  $c_1^2 \equiv g^2 \equiv h \bmod N$  and outputs 1
  - case  $i = 2$ :  $P$  sends  $c_2 := gs$ ,  $V$  then checks that indeed  $c_2^2 \equiv (gs)^2 \equiv g^2 s^2 \equiv hr \bmod N$  and outputs 1

(b) Prove that  $(P, V)$  has soundness bound  $\frac{1}{2}$ .

**Solution.**

- We can assume that  $r$  and  $h$  are invertible modulo  $N$ , since this is checked by  $V$  and if it is not the case  $V$  will abort.
- Assume that  $(r, N) \notin L_R$  and that are  $c_1, c_2$ , such that  $V$  would accept. From the proof system we see that

$$r = hrh^{-1} = c_2^2(c_1^2)^{-1} = (c_2c_1^{-1})^2.$$

This contradicts the fact that  $(r, N) \notin L_R$ .

Hence for some  $i \in \{1, 2\}$ , there is an  $i$  such that  $V$  would not accept. With probability at least  $\frac{1}{2}$ ,  $V$  picks such an  $i$  and will reject the proof.

(c) Prove that  $(P, V)$  is statistically zero-knowledge with auxiliary input.

**Hint:** Everything is similar to the case of graph isomorphisms.

**Solution.** Similar to the case of graph isomorphism, we first construct a simulator  $S_1(r, N)$ :

- $S_1$  picks  $i \xleftarrow{R} \{1, 2\}$  and  $\tilde{g} \xleftarrow{R} \{0, \dots, N-1\}$ , such that  $\tilde{g}$  is invertible modulo  $N$ . If  $i = 1$  then  $S_1$  sets  $h := \tilde{g}^2$ , otherwise  $h := \tilde{g}^2 r^{-1}$ . Then  $S_1$  sends  $h$  to a simulated  $V^*$ .
- After receiving  $i^*$  from  $V^*$ ,  $S_1$  aborts, if  $i \neq i^*$ , otherwise it sends  $c := \tilde{g}$  to  $V^*$  and outputs, what  $V^*$  outputs.

The simulator  $S(r)$  repeats the simulation  $S_1(r)$  until success, but at most  $n := |x|$  times.

The proof system is zero-knowledge if  $\langle P(r, N, s), V^*(r, N) \rangle$  and  $S_1(r, N)|_{\text{success}}$  have the same distribution (where *success* stands for the event that  $S_1$  does not output  $\perp$ ), because of the similarities to the proof system for graph isomorphism. We consider the following games, where  $\mathbb{Z}_N^*$  is the multiplicative group of  $\mathbb{Z}_N$  (all elements from  $\mathbb{Z}_N^*$  having a multiplicative inverse):

$$\begin{aligned} i \xleftarrow{R} \{1, 2\}, \quad \tilde{g} \xleftarrow{R} \mathbb{Z}_N^*, \quad \text{if } i = 1 \text{ then } h := \tilde{g}^2 \text{ else } h := \tilde{g}^2 r^{-1}, \\ i^* \leftarrow V^*(h), \quad \text{out} \leftarrow V^*(\tilde{g}), \quad \text{if } i = i^* \text{ return out else return } \perp \end{aligned} \quad (1)$$

$$\begin{aligned} i \xleftarrow{R} \{1, 2\}, \quad g \xleftarrow{R} \mathbb{Z}_N^*, \quad \mathbf{\text{if } i = 1 \text{ then } \tilde{g} := g \text{ else } \tilde{g} := gs}, \\ \text{if } i = 1 \text{ then } h := \tilde{g}^2 \text{ else } h := \tilde{g}^2 r^{-1}, \\ i^* \leftarrow V^*(h), \quad \text{out} \leftarrow V^*(\tilde{g}), \quad \text{if } i = i^* \text{ return out else return } \perp \end{aligned} \quad (2)$$

$$\begin{aligned} i \xleftarrow{R} \{1, 2\}, \quad g \xleftarrow{R} \mathbb{Z}_N^*, \quad \text{if } i = 1 \text{ then } \tilde{g} := g \text{ else } \tilde{g} := gs, \quad \mathbf{h := g^2}, \\ i^* \leftarrow V^*(h), \quad \text{out} \leftarrow V^*(\tilde{g}), \quad \text{if } i = i^* \text{ return out else return } \perp \end{aligned} \quad (3)$$

$$\begin{aligned} i \xleftarrow{R} \{1, 2\}, \quad g \xleftarrow{R} \mathbb{Z}_N^*, \quad h := g^2, \quad i^* \leftarrow V^*(h), \\ \mathbf{\text{if } i^* = 1 \text{ then } \tilde{g} := g \text{ else } \tilde{g} := gs}, \\ \text{out} \leftarrow V^*(\tilde{g}), \quad \text{if } i = i^* \text{ return out else return } \perp \end{aligned} \quad (4)$$

$$\begin{aligned} g \xleftarrow{R} \mathbb{Z}_N^*, \quad h := g^2, \quad i^* \leftarrow V^*(h), \quad \text{if } i^* = 1 \text{ then } \tilde{g} := g \text{ else } \tilde{g} := gs, \\ \text{out} \leftarrow V^*(\tilde{g}), \quad \mathbf{i \xleftarrow{R} \{1, 2\}}, \quad \text{if } i = i^* \text{ return out else return } \perp \end{aligned} \quad (5)$$

$$\begin{aligned} g \xleftarrow{R} \mathbb{Z}_N^*, \quad h := g^2, \quad i^* \leftarrow V^*(h), \quad \text{if } i^* = 1 \text{ then } \tilde{g} := g \text{ else } \tilde{g} := gs, \\ \text{out} \leftarrow V^*(\tilde{g}), \quad \mathbf{\text{with probability } \frac{1}{2} \text{ return out else return } \perp} \end{aligned} \quad (6)$$

As for the proof system for graph isomorphism we have that  $S_1(r, N)$  has the same distribution as game (1), game ( $i$ ) has the same distribution as game ( $i + 1$ ) for  $i \in \{1, \dots, 5\}$  and game (6) – under the condition that the output is not  $\perp$  – has the same distribution as  $\langle P(r, N, s), V^*(r, N) \rangle$ .

Hence  $\langle P(r, N, s), V^*(r, N) \rangle$  and  $S_1(r, N)|_{\text{success}}$  have the same distribution. Completely analogous to the calculation in the lecture for the case of graph isomorphism,

it follows that  $\langle P(r, N, s), V^*(r, N) \rangle$  and  $S(r, N)$  have exponentially small statistical distance.

## Solution of Exercise Sheet 5

Out: Wed, Nov 26, 2008

Due: Fri, Dec 12, 2008, before noon

**Problem 1: Computational indistinguishability (21 Points)**

- (a) Prove that statistical indistinguishability implies computational indistinguishability. More precisely, show that if  $\{X_{x,a}\}_{x,a}$  and  $\{Y_{x,a}\}_{x,a}$  are statistically indistinguishable for all  $(x, a) \in A$ , then  $\{X_{x,a}\}_{x,a}$  and  $\{Y_{x,a}\}_{x,a}$  are computationally indistinguishable for all  $(x, a) \in A$ .

**Note:** This would be trivial when defining statistical distance analogous to computational distance (just with unbounded distinguisher  $D$ ). However, you should show that computational indistinguishability is implied by the normal formulation of statistical indistinguishability that uses the statistical distance (Definition 13 in the lecture notes).

**Solution.** Suppose  $\{X_{x,a}\}_{x,a}$  and  $\{Y_{x,a}\}_{x,a}$  are statistically indistinguishable for all  $(x, a) \in A$ , then  $\text{SD}((x, a, X_{x,a}); (x, a, Y_{x,a}))$  is negligible.

Let  $D$  be a distinguisher for  $\{X_{x,a}\}_{x,a}$  and  $\{Y_{x,a}\}_{x,a}$ . Now we have that

$$\begin{aligned} & \left| \Pr[D(x, a, X_{x,a}) = 1] - \Pr[D(x, a, Y_{x,a}) = 1] \right| \\ &= \left| \Pr[\tilde{D}(x, a, X_{x,a}, R) = 1] - \Pr[\tilde{D}(x, a, Y_{x,a}, R) = 1] \right| \\ &\leq \text{SD}(\tilde{D}(x, a, X_{x,a}, R); \tilde{D}(x, a, Y_{x,a}, R)) \\ &\leq \text{SD}((x, a, X_{x,a}, R); (x, a, Y_{x,a}, R)) \\ &= \text{SD}((x, a, X_{x,a}); (x, a, Y_{x,a})) \end{aligned}$$

which is negligible. Here we first add the randomness of  $D$  as additional argument for the deterministic distinguisher  $\tilde{D}$ . Then we use the fact, that the statistical distance is at least as big as the difference  $\tilde{D}$  can compute. Finally we use the properties of the statistical distance (exercise sheet 2, problem 1(b,c,d)).

- (b) Show that computational indistinguishability is an equivalence relation. More precisely, assume that  $\{X_{x,a}\}_{x,a}$  and  $\{Y_{x,a}\}_{x,a}$  are computationally indistinguishable for all  $(x, a) \in A$ , and that  $\{Y_{x,a}\}_{x,a}$  and  $\{Z_{x,a}\}_{x,a}$  are computationally indistinguishable for all  $(x, a) \in A$ . Under these conditions, show that  $\{X_{x,a}\}_{x,a}$  and  $\{Z_{x,a}\}_{x,a}$  are computationally indistinguishable for all  $(x, a) \in A$  (transitivity), that  $\{X_{x,a}\}_{x,a}$  and  $\{X_{x,a}\}_{x,a}$  are computationally indistinguishable for all  $(x, a) \in A$  (reflexivity), and that  $\{Y_{x,a}\}_{x,a}$  and  $\{X_{x,a}\}_{x,a}$  are computationally indistinguishable for all  $(x, a) \in A$  (symmetry).

**Solution.**

- reflexivity:

$$\left| \Pr[D(x, a, X_{x,a}) = 1] - \Pr[D(x, a, X_{x,a}) = 1] \right| = 0$$

and 0 is negligible.

- symmetry:

$$\begin{aligned} & \left| \Pr[D(x, a, X_{x,a}) = 1] - \Pr[D(x, a, Y_{x,a}) = 1] \right| \\ &= \left| \Pr[D(x, a, Y_{x,a}) = 1] - \Pr[D(x, a, X_{x,a}) = 1] \right| \\ &\leq \mu(|x|) \end{aligned}$$

for some negligible  $\mu$ .

- transitivity:

$$\begin{aligned} & \left| \Pr[D(x, a, X_{x,a}) = 1] - \Pr[D(x, a, Z_{x,a}) = 1] \right| \\ &= \left| \Pr[D(x, a, X_{x,a}) = 1] - \Pr[D(x, a, Y_{x,a}) = 1] \right| \\ &\quad + \left| \Pr[D(x, a, Y_{x,a}) = 1] - \Pr[D(x, a, Z_{x,a}) = 1] \right| \\ &\leq \left| \Pr[D(x, a, X_{x,a}) = 1] - \Pr[D(x, a, Y_{x,a}) = 1] \right| + \\ &\quad \left| \Pr[D(x, a, Y_{x,a}) = 1] - \Pr[D(x, a, Z_{x,a}) = 1] \right| \\ &\leq \mu_1(|x|) + \mu_2(|x|) \\ &\leq \mu(|x|) \end{aligned}$$

for some negligible  $\mu_1, \mu_2$  and  $\mu$ . Here he have used the fact, that the sum of two negligible functions is still negligible.

- (c) Fix two families  $\{X_{x,a}\}_{x,a}$  and  $\{Y_{x,a}\}_{x,a}$  that are computationally indistinguishable for all  $(x, a) \in A$ . Assume that  $|X_{x,a}|$  and  $|Y_{x,a}|$  are polynomially bounded in  $|x|$ . Fix an efficiently computable function  $f$  that outputs bitstrings. Show that  $\{f(X_{x,a})\}_{x,a}$  and  $\{f(Y_{x,a})\}_{x,a}$  are computationally indistinguishable for all  $(x, a) \in A$ .

**Solution.** Suppose  $\{X_{x,a}\}_{x,a}$  and  $\{Y_{x,a}\}_{x,a}$  are computationally indistinguishable. Then for all  $D$  polynomial time in its first input there is a negligible function  $\mu$ , such that:

$$\left| \Pr[D(x, a, X_{x,a}) = 1] - \Pr[D(x, a, Y_{x,a}) = 1] \right| \leq \mu(|x|).$$

Let  $D'(\cdot, \cdot, \cdot) = D(\cdot, \cdot, f(\cdot))$ , meaning on input  $(x, a, y)$   $D'$  computes  $f(y)$ , runs  $D(x, a, f(y))$  and outputs, what  $D$  outputs.  $D'$  is polynomial time in its first input, since  $f$  is efficiently computable<sup>1</sup> and  $D$  is polynomial time in its first input. Then we have:

$$\begin{aligned} & \left| \Pr[D(x, a, f(X_{x,a})) = 1] - \Pr[D(x, a, f(Y_{x,a})) = 1] \right| \\ &= \left| \Pr[D'(x, a, X_{x,a}) = 1] - \Pr[D'(x, a, Y_{x,a}) = 1] \right| \\ &\leq \mu(|x|). \end{aligned}$$

- (d) Assume that pseudo-random generators exist (i.e., an efficiently computable family of functions  $G_n : \{0, 1\}^n \rightarrow \{0, 1\}^{n+1}$  exists such that  $\{G_{|x|}(U_{|x|})\}_{x,a}$  and  $\{U_{|x|+1}\}_{x,a}$  are computationally indistinguishable for all  $x, a$ ; here  $U_n$  is the uniform distribution of  $\{0, 1\}^n$ ).

Under this assumption, show that (c) does not hold in general if  $f$  is not efficiently computable.

**Solution.** Choose  $\{X_{x,a}\}_{x,a} = \{G_{|x|}(U_{|x|})\}_{x,a}$  and  $\{Y_{x,a}\}_{x,a} = \{U_{|x|+1}\}_{x,a}$ . Let  $f : \{0, 1\}^{n+1} \rightarrow \{0, 1\}$  be the (inefficient) function that tries to compute preimages of the pseudo-random generator  $G_n$ :

$$f(y) = \begin{cases} 1, & \text{if } y = G_{|x|}(y') \text{ for some } y' \\ 0, & \text{otherwise.} \end{cases}$$

$\{G_{|x|}(U_{|x|})\}_{x,a}$  and  $\{U_{|x|+1}\}_{x,a}$  are computationally indistinguishable according to the definition of pseudo-random generators. But  $\{f(G_{|x|}(U_{|x|}))\}_{x,a}$  and  $\{f(U_{|x|+1})\}_{x,a}$  are not computationally indistinguishable: for  $\{G_{|x|}(U_{|x|})\}_{x,a}$   $f$  will always find a preimage and output 1, while in at least half of the cases  $f$  will output 0 for  $\{U_{|x|+1}\}_{x,a}$  (this is because for  $G_n : \{0, 1\}^n \rightarrow \{0, 1\}^{n+1}$  only  $2^n$  out of  $2^{n+1}$  many values can have a preimage).

- (e) Fix two families  $\{X_{x,a}\}_{x,a}$  and  $\{Y_{x,a}\}_{x,a}$  that are computationally indistinguishable for all  $(x, a) \in A$ . Fix an efficiently computable function  $f : \{0, 1\}^* \rightarrow \mathbb{N}$ . Assume that  $\{X_{x,a}\}_{x,a}$  and  $\{Y_{x,a}\}_{x,a}$  are efficiently sampleable. (This means that there is a probabilistic algorithm  $A_X$  polynomial-time in its first argument such that  $A_X(x, a)$  has the same distribution as  $X_{x,a}$ . And for  $\{Y_{x,a}\}_{x,a}$  analogous.) Assume that  $f(x)$  is polynomially bounded in  $|x|$ . By  $X_{x,a}^f$  denote the distribution resulting from choosing  $f(x)$  values  $x_i$  independently distributed according to the distribution  $X_{x,a}$ . (I.e., choose  $x_1 \leftarrow X_{x,a}, \dots, x_{f(x)} \leftarrow X_{x,a}$  and return  $(x_1, \dots, x_{f(x)})$ .) Then  $\{X_{x,a}^f\}_{x,a}$  and  $\{Y_{x,a}^f\}_{x,a}$  are computationally indistinguishable for all  $(x, a) \in A$ .

---

<sup>1</sup>This also implicitly uses that  $|X_{x,a}|$  and  $|Y_{x,a}|$  are polynomially bounded, otherwise  $f(X_{x,a})$  and  $f(Y_{x,a})$  would be hard to compute even for efficient  $f$ .

**Hint:** Assume a fixed distinguisher that distinguishes  $\{X_{x,a}^f\}_{x,a}$  and  $\{Y_{x,a}^f\}_{x,a}$ . Define a game  $H_{A,x,a}^i$  (depending on some distribution  $A$ ) that does the following: Let  $x_1 \leftarrow X_{x,a}, \dots, x_i \leftarrow X_{x,a}, x_{i+1} \leftarrow A, x_{i+2} \leftarrow Y_{x,a}, \dots, x_{f(x)} \leftarrow Y_{x,a}$ . Let  $H_{A,x,a}^*$  be the game that picks  $i \in \{0, \dots, f(x) - 1\}$  at random and then runs  $H_{A,x,a}^i$ . Then prove a negligible bound on  $\Pr[D(H_{X_{x,a},x,a}^*, x, a) = 1] - \Pr[D(H_{Y_{x,a},x,a}^*, x, a) = 1]$ . Then note that  $\Pr[D(H_{A,x,a}^*, x, a) = 1] = \frac{1}{n} \sum_{i=0}^{n-1} \Pr[D(H_{A,x,a}^i, x, a) = 1]$  and use this to simplify  $\Pr[D(H_{X_{x,a},x,a}^*, x, a) = 1] - \Pr[D(H_{Y_{x,a},x,a}^*, x, a) = 1]$  (so that it is expressed in terms of  $\Pr[D(H_{Y_{x,a},x,a}^0, x, a) = 1]$  and  $\Pr[D(H_{X_{x,a},x,a}^{f(x)-1}, x, a) = 1]$ ).

**Solution.** Let  $n := f(x)$ . Let  $H_{A,x,a}^*$  and  $H_{A,x,a}^i$  be defined as above. Note that  $H_{X_{x,a},x,a}^{n-1} = X_{x,a}^f$ ,  $H_{Y_{x,a},x,a}^0 = Y_{x,a}^f$  and  $H_{X_{x,a},x,a}^i = H_{Y_{x,a},x,a}^{i+1}$ .

Suppose  $\{X_{x,a}\}_{x,a}$  and  $\{Y_{x,a}\}_{x,a}$  are computationally indistinguishable. Suppose furthermore that there exists a distinguisher  $D$  for  $\{X_{x,a}^f\}_{x,a}$  and  $\{Y_{x,a}^f\}_{x,a}$ : there is an algorithm  $D$  polynomial time in its first input and a non-negligible function  $\mu'$  such that

$$\left| \Pr[D(x, a, X_{x,a}^f) = 1] - \Pr[D(x, a, Y_{x,a}^f) = 1] \right| > \mu'(|x|).$$

Consider  $D'$ , which on input  $(x, a, Z)$  picks  $i \in \{0, \dots, f(x) - 1\}$  at random, chooses  $x_1 \leftarrow X_{x,a}, \dots, x_i \leftarrow X_{x,a}, x_{i+1} \leftarrow Z, x_{i+2} \leftarrow Y_{x,a}, \dots, x_{f(x)} \leftarrow Y_{x,a}$ , runs  $D(x, a, (x_1, \dots, x_{f(x)}))$  and outputs, what  $D$  outputs. Then  $\Pr[D'(x, a, X_{x,a}) = 1] = \Pr[D(x, a, H_{X_{x,a},x,a}^*)]$  and analogous for  $Y_{x,a}$ . Furthermore

$$\begin{aligned} & \left| \Pr[D'(x, a, X_{x,a}) = 1] - \Pr[D'(x, a, Y_{x,a}) = 1] \right| \\ &= \left| \Pr[D(x, a, H_{X_{x,a},x,a}^*) = 1] - \Pr[D(x, a, H_{Y_{x,a},x,a}^*) = 1] \right| \\ &= \left| \frac{1}{n} \sum_{i=0}^{n-1} \Pr[D(x, a, H_{X_{x,a},x,a}^i) = 1] - \frac{1}{n} \sum_{i=0}^{n-1} \Pr[D(x, a, H_{Y_{x,a},x,a}^i) = 1] \right| \\ &= \frac{1}{n} \left| \sum_{i=0}^{n-1} (\Pr[D(x, a, H_{X_{x,a},x,a}^i) = 1] - \Pr[D(x, a, H_{Y_{x,a},x,a}^i) = 1]) \right| \\ &= \frac{1}{n} \left| \Pr[D(x, a, H_{X_{x,a},x,a}^{n-1}) = 1] - \Pr[D(x, a, H_{Y_{x,a},x,a}^0) = 1] \right| \\ &= \frac{1}{n} \left| \Pr[D(x, a, X_{x,a}^f) = 1] - \Pr[D(x, a, Y_{x,a}^f) = 1] \right| \\ &> \frac{\mu'(|x|)}{|x|} \end{aligned}$$

Since  $\mu'/n$  is not-negligible, this means that  $D'$  successfully distinguishes  $X_{x,a}$  and  $Y_{x,a}$ . This is a contradiction to the assumption that  $X_{x,a}$  and  $Y_{x,a}$  are computationally indistinguishable. Hence, a distinguisher  $D$  for  $\{X_{x,a}^f\}$  and  $\{Y_{x,a}^f\}$  cannot exist, and hence  $\{X_{x,a}^f\}$  and  $\{Y_{x,a}^f\}$  are computationally indistinguishable.

- (f) Show that (e) does not hold in general, if  $\{X_{x,a}\}_{x,a}$  and  $\{Y_{x,a}\}_{x,a}$  are not efficiently sampleable. More precisely, show that there are families  $\{X_{x,a}\}_{x,a}$  and  $\{Y_{x,a}\}_{x,a}$  with the following properties: They are computationally indistinguishable for all  $(x, a) \in \{(1^n, 0) : n \in \mathbb{N}\}$ . But  $\{X_{x,a}^2\}_{x,a}$  and  $\{Y_{x,a}^2\}_{x,a}$  are not computationally indistinguishable for all  $(x, a) \in \{(1^n, 0) : n \in \mathbb{N}\}$ .

**Hint:** You may use the following fact without proof: There is a sequence of sets  $S_n \subseteq \{0, 1\}^n$  such that  $|S_n| = n$  and such that for any polynomial time algorithm  $A$ ,  $|\Pr[A(1^n, s) = 1 : s \xleftarrow{R} S_n] - \Pr[A(1^n, s) = 1 : s \xleftarrow{R} \{0, 1\}^n]|$  is negligible in  $n$ .

**Solution.** Choose  $X_{x,a} = S_n$  and  $Y_{x,a} = U_n$  for  $(x, a) = (1^n, 0)$ . According to the hint,  $\{X_{x,a}\}_{x,a}$  and  $\{Y_{x,a}\}_{x,a}$  are computationally indistinguishable. Now consider  $D$ : on input  $(1^n, 0, (y_1, y_2))$   $D$  checks whether  $y_1 = y_2$ . If this is the case, it outputs 1, otherwise it outputs 0. Now for  $(x, a) = (1^n, 0)$   $D$  will always output 1 with probability  $\frac{1}{n}$  for  $\{X_{x,a}^2\}_{x,a}$ , but for  $\{Y_{x,a}^2\}_{x,a}$  the probability, that  $y_1 = y_2$  and  $D$  will output 1 is  $\frac{1}{2^n}$ . Hence  $\{X_{x,a}^2\}_{x,a}$  and  $\{Y_{x,a}^2\}_{x,a}$  are not computationally indistinguishable.

- (g) Fix two families  $\{X_{x,a}\}_{x,a}$  and  $\{Y_{x,a}\}_{x,a}$  that are computationally indistinguishable for all  $(x, a) \in A$ . Assume that  $X_{x,a} \in \{0, 1\}$  and  $Y_{x,a} \in \{0, 1\}$  for all  $x, a$ . Show that  $\{X_{x,a}\}_{x,a}$  and  $\{Y_{x,a}\}_{x,a}$  are statistically indistinguishable for all  $(x, a) \in A$ .

**Solution.** Let  $\{X_{x,a}\}_{x,a}$  and  $\{Y_{x,a}\}_{x,a}$  be computationally indistinguishable for all  $(x, a) \in A$  and both,  $X_{x,a} \in \{0, 1\}$  and  $Y_{x,a} \in \{0, 1\}$ . We consider a distinguisher  $D$  which on input  $(x, a, Z)$  outputs  $Z$ . We then have for some negligible  $\mu$

$$\begin{aligned} \mu(|x|) &\geq |\Pr[D(x, a, X_{x,a}) = 1] - \Pr[D(x, a, Y_{x,a}) = 1]| \\ &= |\Pr[X_{x,a} \in \{1\}] - \Pr[Y_{x,a} \in \{1\}]| \\ &= |\Pr[X_{x,a} \in \{0\}] - \Pr[Y_{x,a} \in \{0\}]|, \end{aligned}$$

since

$$|\Pr[X_{x,a} \in T] - \Pr[Y_{x,a} \in T]| = |\Pr[X_{x,a} \in T^C] - \Pr[Y_{x,a} \in T^C]|.$$

For  $T = \{0, 1\}$  and  $T = \emptyset$  we have

$$|\Pr[X_{x,a} \in T] - \Pr[Y_{x,a} \in T]| = 0.$$

So  $\{1\}$  and  $\{0\}$  are sets, that maximise the statical distance. Hence  $\text{SD}(X_{x,a}; Y_{x,a}) \leq \mu(|x|)$  and thus  $X_{x,a}$  and  $Y_{x,a}$  are statistically indistinguishable.

## Problem 2: On the G3C proof system (14 Points)

Shorthand for this problem:  $a?c : d$  equals  $c$  if  $a$  is true and it equals  $d$  if  $a$  is false (like in the programming languages C and Java).

Consider the following game  $\text{Game}_1(G, \gamma, z)$  with  $G = (V, E)$  and  $n := |G|$ :

$$\begin{aligned} (v_1, v_2) &\stackrel{R}{\leftarrow} E, \rho \stackrel{R}{\leftarrow} \text{perm}\{R, G, B\}, \\ &\quad (c_v, u_v) \leftarrow \text{Com}(1^n, (v \notin \{v_1, v_2\})?R : \rho(\gamma(v))) \text{ for all } v \in V, \\ (v_1^*, v_2^*) &\leftarrow V^*(G, z, \{c_v\}_{v \in V}), \text{ out} \leftarrow V^*(\rho(\gamma(v_1)), \rho(\gamma(v_2)), u_{v_1}, u_{v_2}), \\ &\quad \text{return } ((v_1, v_2) = (v_1^*, v_2^*))? \text{out} : \perp. \end{aligned}$$

Here  $V^*$  is some probabilistic polynomial time machine that may keep state between invocations.

Note that this game, although not formulated in the same way, can be easily seen to have the same output distribution as the  $\text{Game}_1$  presented in the lecture (at least if  $\gamma$  is a 3-colouring of  $G$ ).

Consider the following game  $\text{Game}_2(G, \gamma, z)$ :

$$\begin{aligned} (v_1, v_2) &\stackrel{R}{\leftarrow} E, \rho \stackrel{R}{\leftarrow} \text{perm}\{R, G, B\}, (c_v, u_v) \leftarrow \text{Com}(\boldsymbol{\rho}(\boldsymbol{\gamma}(\boldsymbol{v}))) \text{ for all } v \in V, \\ (v_1^*, v_2^*) &\leftarrow V^*(G, z, \{c_v\}_{v \in V}), \text{ out} \leftarrow V^*(\rho(\gamma(v_1)), \rho(\gamma(v_2)), u_{v_1}, u_{v_2}), \\ &\quad \text{return } ((v_1, v_2) = (v_1^*, v_2^*))? \text{out} : \perp. \end{aligned}$$

Differences to  $\text{Game}_1$  are highlighted in bold. Note that this game, although not formulated in the same way, can be easily seen to have the same output distribution as the  $\text{Game}_2$  presented in the lecture.

**Your task:** Prove that  $\{\text{Game}_1(G, \gamma, z)\}_{G, (\gamma, z)}$  and  $\{\text{Game}_2(G, \gamma, z)\}_{G, (\gamma, z)}$  are computationally indistinguishable for all  $(G, \gamma) \in R_{G3C}$ .

**Hint:** For a given distinguisher  $D$  that successfully distinguishes  $\{\text{Game}_1(G, \gamma, z)\}_{G, (\gamma, z)}$  and  $\{\text{Game}_2(G, \gamma, z)\}_{G, (\gamma, z)}$ , do the following: Assume some numbering on the vertices  $V = \{v^{(1)}, \dots, v^{(m)}\}$ . Define a “hybrid game”  $H_i(G, \gamma, z)$  that works like  $\text{Game}_1$  and  $\text{Game}_2$ , except that for the vertices  $v = v^{(1)}, \dots, v^{(i)}$ , it produces the commitments as  $(c_v, u_v) \leftarrow \text{Com}(1^n, (v \notin \{v_1, v_2\})?R : \rho(\gamma(v)))$ , and for the vertices  $v = v^{(i+1)}, \dots, v^{(m)}$ , it produces the commitments as  $(c_v, u_v) \leftarrow \text{Com}(1^n, \rho(\gamma(v)))$ . Show that  $D$  successfully distinguishes  $H_i(G, \gamma, z)$  and  $H_{i+1}(G, \gamma, z)$  for some  $i$ . From this, construct an attack against the computational hiding property of  $\text{Com}$ .

**Solution.** Suppose  $\{\text{Game}_1(G, \gamma, z)\}_{G, (\gamma, z)}$  and  $\{\text{Game}_2(G, \gamma, z)\}_{G, (\gamma, z)}$  are not computationally indistinguishable. Then there exist a non-negligible function  $\mu$  and sequences  $\{G_n\}_n, \{\gamma_n\}_n, \{z_n\}_n$  with  $(G_i, \gamma_i) \in A$  and  $|G_n| = n$ :

$$\left| \Pr[D(G_n, \gamma_n, z_n, \text{Game}_1(G_n, \gamma_n, z_n)) = 1] - \Pr[D(G_n, \gamma_n, z_n, \text{Game}_2(G_n, \gamma_n, z_n)) = 1] \right| \geq \mu(n).$$

In the following, we omit writing the index  $n$ , but it will be implicitly assumed. For example, the preceding equation would be written

$$\left| \Pr[D(G, \gamma, z, \text{Game}_1(G, \gamma, z)) = 1] - \Pr[D(G, \gamma, z, \text{Game}_2(G, \gamma, z)) = 1] \right| \geq \mu(n).$$

Given a graph  $G = (V, E)$ , We assume some numbering on the vertices  $V = \{v^{(1)}, \dots, v^{(m)}\}$  (the lexicographic ordering on the vertex names, for example). We define the hybrid game  $H_i(G, \gamma, z)$  for  $i = 0, \dots, n$  as:

$$\begin{aligned} (v_1, v_2) &\stackrel{R}{\leftarrow} E, \quad \rho \stackrel{R}{\leftarrow} \text{perm}\{R, G, B\}, \\ (c_v, u_v) &\leftarrow \text{Com}(1^n, (v \notin \{v_1, v_2\})?R : \rho(\gamma(v))) \text{ for all } v \in \{v^{(1)}, \dots, v^{(i)}\}, \\ (c_v, u_v) &\leftarrow \text{Com}(1^n, \rho(\gamma(v))) \text{ for all } v \in \{v^{(i+1)}, \dots, v^{(m)}\}, \\ (v_1^*, v_2^*) &\leftarrow V^*(G, z, \{c_v\}_{v \in V}), \quad \text{out} \leftarrow V^*(\rho(\gamma(v_1)), \rho(\gamma(v_2)), u_{v_1}, u_{v_2}), \\ &\text{return } ((v_1, v_2) = (v_1^*, v_2^*))? \text{out} : \perp. \end{aligned}$$

Then he have:

$$\begin{aligned} \mu(|x|) &\leq \left| \Pr[D(G, \gamma, z, \text{Game}_1(G, \gamma, z)) = 1] - \Pr[D(G, \gamma, z, \text{Game}_2(G, \gamma, z)) = 1] \right| \\ &= \left| \Pr[D(G, \gamma, z, H_m(G, \gamma, z)) = 1] - \Pr[D(G, \gamma, z, H_0(G, \gamma, z)) = 1] \right| \\ &= \left| \sum_{i=0}^{m-1} (\Pr[D(G, \gamma, z, H_{i+1}(G, \gamma, z)) = 1] - \Pr[D(G, \gamma, z, H_i(G, \gamma, z)) = 1]) \right| \\ &\leq \sum_{i=0}^{m-1} \left| \Pr[D(G, \gamma, z, H_{i+1}(G, \gamma, z)) = 1] - \Pr[D(G, \gamma, z, H_i(G, \gamma, z)) = 1] \right| \end{aligned}$$

This implies, that there exist a sequence  $\{i_n\}_n$  such that

$$\frac{\mu(|x|)}{m} \leq \left| \Pr[D(G, \gamma, z, H_{i+1}(G, \gamma, z)) = 1] - \Pr[D(G, \gamma, z, H_i(G, \gamma, z)) = 1] \right|$$

(otherwise the sum above would not reach  $\mu(|x|)$ ). The only difference between  $H_i(G, \gamma, z)$  and  $H_{i+1}(G, \gamma, z)$  is the commitment  $(c_{v^{(i+1)}}, u_{v^{(i+1)}})$  to the  $i + 1$ -th vertex  $v^{(i+1)}$ .

From the computational hiding property of  $\text{Com}$ , it follows that  $\left| \Pr[D(G, \gamma, z, H_{i+1}(G, \gamma, z)) = 1] - \Pr[D(G, \gamma, z, H_i(G, \gamma, z)) = 1] \right|$  is negligible. (Note that the unveil information for the commitment  $c_{v^{(i+1)}}$  is only used if  $v^{(i+1)} \in \{v_1, v_2\}$ . And in that case, in both games  $c_{v^{(i+1)}}$  is a commitment to the same value  $\rho(\gamma(v^{(i+1)}))$ ).

Hence  $\mu(|x|)/m$  is negligible. Since  $m$  is polynomially bounded, this is a contradiction to  $\mu$  being non-negligible.

## Solution of Exercise Sheet 6

Out: Wed, Dec 21, 2008

Due: Fri, Jan 9, 2009, before noon

**Problem 1: No auxiliary input – no composition**  
**(5+1+4(+3)=10 points)**

In this exercise we show that there is a proof system that is computationally zero-knowledge *without* auxiliary input, but that does not compose sequentially. (Note however that this proof system will have a computationally unbounded honest prover; no such example is known with polynomial-time honest provers.)

By  $U_n$  we denote the uniform distribution on  $\{0, 1\}^n$ .

**Definition 1 (Pseudorandom sets)** Fix a sequence of sets  $S_n \subseteq \{0, 1\}^n$ . In slight abuse of notation, we also write  $S_n$  for the random variable that returns a uniformly random  $s \in S_n$ . We call  $S_n$  pseudorandom if  $\{S_n\}_{n,a}$  and  $\{U_n\}_{n,a}$  are computationally indistinguishable.

**Definition 2 (Evasive sets)** We call a sequence  $S_n$  of sets evasive if for any (uniform) polynomial probabilistic-time algorithm  $A$  there is a negligible  $\mu$  such that for all  $n$  we have  $\Pr[A(1^n) \in S_n] \leq \mu(n)$ .

**Theorem 1 ([GK90])** There exists an evasive pseudorandom sequence  $S_n$  of sets.

In the following, let an evasive pseudorandom sequence  $S_n$  of sets be given.

Consider the following proof system  $(P, V)$  (adapted from [GK96]):

- Relation:  $R = \{(x, w) : x = 1^n\}$ . (The valid statements are strings containing only 1's, anything is a witness)
- Prover  $P$ 's input:  $x, w$ .
- Verifier  $V$ 's input:  $x$ . Let  $n := |x|$ .
- $V$  samples  $r \leftarrow U_n$  and sends  $r$  to  $P$ .
- $P$  checks whether  $r \in S_n$ . If so,  $P$  sends  $w$  to  $V$ . Otherwise,  $P$  picks a random  $s \in S_n$  and sends  $s$  to  $V$ .
- $V$  outputs 1 if  $x = 1^n$ .

Obviously, this proof system has perfect completeness and perfect soundness. (It is even a proof of knowledge with knowledge error 0.)

(a) Show that  $(P, V)$  is computationally zero-knowledge without auxiliary input for  $R$ .

**Solution.** We have to show that there exists an algorithm  $S$  polynomial-time in its first input such that the following two families of distributions are computationally indistinguishable for all  $x, w \in \{0, 1\}^*$  with  $(x, w) \in R$ :

$$\left\{ \langle P(x, w), V^*(x) \rangle \right\}_{x, w} \quad \text{and} \quad \left\{ S(x) \right\}_{x, w}$$

Consider the following algorithm  $S(x)$  invoking  $V^*(x)$ :

- $S$  receives  $r$  from  $V^*$ .
- $S$  picks  $s \in U_n$  and sends  $s$  to  $V^*$ .
- Finally  $S$  outputs, what  $V^*$  outputs.

Furthermore consider the prover  $P'(x, w)$ , which always picks  $s \in S_n$  and sends  $s$  to  $V$ , no matter whether  $r$  received from  $V$  is in  $S_n$ . For  $P'$  we have

$$\begin{aligned} & \left| \Pr[D(x, w, \langle P(x, w), V^*(x) \rangle) = 1] - \Pr[D(x, w, \langle P'(x, w), V^*(x) \rangle) = 1] \right| \\ & \leq \Pr[r \in S_n] \\ & = \mu(|x|), \end{aligned}$$

where  $\mu$  is a negligible function, since the probability that  $V^*$  chooses an  $r \in S_n$  is negligible ( $S_n$  is an evasive sequence of sets). Note that  $V^*(x) = V^*(1^n)$ . At this point the approach fails if we consider  $R = \{(x, w)\}$  as in part (d). Hence

$$\left\{ \langle P(x, w), V^*(x) \rangle \right\}_{x, w} \quad \text{and} \quad \left\{ \langle P'(x, w), V^*(x) \rangle \right\}_{x, w}$$

are computationally indistinguishable. Furthermore:

$$\begin{aligned} & \left| \Pr[D(x, a, \langle P'(x, w), V^*(x) \rangle) = 1] - \Pr[D(x, a, S(x)) = 1] \right| \\ & = \left| \Pr[D(x, a, out) = 1 : s \stackrel{R}{\leftarrow} S_n, out \leftarrow V^*(s)] \right. \\ & \quad \left. - \Pr[D(x, a, out) = 1 : s \stackrel{R}{\leftarrow} U_n, out \leftarrow V^*(s)] \right| \\ & \leq \mu'(|x|), \end{aligned}$$

where  $\mu'$  is a negligible function, since the probability that  $V^*$  can distinguish  $s \stackrel{R}{\leftarrow} S_n$  from  $s \stackrel{R}{\leftarrow} U_n$  is negligible ( $S_n$  is a sequence of pseudorandom sets). Hence

$$\left\{ \langle P'(x, w), V^*(x) \rangle \right\}_{x, w} \quad \text{and} \quad \left\{ S(x) \right\}_{x, w}$$

are computationally indistinguishable and by transitivity

$$\left\{ \langle P(x, w), V^*(x) \rangle \right\}_{x, w} \quad \text{and} \quad \left\{ S(x) \right\}_{x, w}$$

are computationally indistinguishable and  $(P, V)$  is zero-knowledge without auxiliary input.

- (b) Show that  $(P, V)$  is not computationally zero-knowledge with auxiliary input for  $R$ .

**Solution.** Consider the verifier  $V^*(x, z)$ :

- $V^*$  sends  $z$  to the prover  $P$ .
- $V^*$  receives  $y$  from  $P$ .
- $V^*$  outputs  $y$ .

For a sequence of auxiliary inputs  $z_n \in S_n$ ,  $(x_n, w_n) \in R$ , and  $(x_n, w'_n) \in R$  with  $w_n \neq w'_n$  and  $|x_n| = n$ , we have to show that

$$\left\{ \langle P(x_n, (w_n, z_n)), V^*(x_n, z_n) \rangle \right\}_{x_n, (w_n, z_n)} \quad \text{and} \quad \left\{ S(x_n, z_n) \right\}_{x_n, (w_n, z_n)}$$

as well as

$$\left\{ \langle P(x_n, (w'_n, z_n)), V^*(x_n, z_n) \rangle \right\}_{x_n, (w'_n, z_n)} \quad \text{and} \quad \left\{ S(x_n, z_n) \right\}_{x_n, (w'_n, z_n)}$$

are computationally indistinguishable. But if we fix some  $z_n \in S_n$ ,  $w_n = 0^n$ ,  $w'_n = 1^n$  for each  $n$ ,  $\langle P(x_n, (w_n, z_n)), V^*(x_n, z_n) \rangle = w_n$  and  $\langle P(x_n, (w'_n, z_n)), V^*(x_n, z_n) \rangle = w'_n$  and hence  $S(x_n, z_n)$  would have to have an output indistinguishable both from  $w_n = 0^n$  and  $w'_n = 1^n$  which is a contradiction. Hence there cannot exist a simulator  $S(x_n, z_n)$  that can simulate the real proof for all  $(x_n, w_n) \in R$ .

- (c) Show that  $(P^2, V^2)$  (in the sense of Definition 5 in the lecture notes) is not computationally zero-knowledge without auxiliary input for  $R$ .

**Solution.** Consider the verifier  $V^*(x)$ :

- $V^*$  sends  $r \xleftarrow{R} U_n$  to the prover  $P$ .
- $V^*$  receives  $y_1$  from  $P$ , the first invocation of  $P$  ends.
- In the second invocation of  $P$ ,  $V^*$  sends  $y_1$  to  $P$ .
- $V^*$  receives  $y_2$  from  $P$  and outputs  $y_2$ .

The probability that  $V^*$  sends  $r \in S_n$  to the first invocation of  $P$  is negligible. Hence the first invocation of  $P$  sends  $y_1 \in S_n$  to  $V^*$  with overwhelming probability.  $V^*$  now sends  $y_1$  to the second invocation of  $P$  and receives  $y_2$  from  $P$ . Since  $y_1 \in S_n$ ,  $y_2 = w$  with overwhelming probability. Again for  $(x, w_1) \in R$  and  $(x, w_2) \in R$  there cannot exist  $S(x)$  simulating the interaction between  $P$  and  $V^*$  for  $(x, w_1)$  and  $(x, w_2)$ .

- (d) For three bonus points: Show that (a) would not hold if we had defined  $R$  to contain all  $(x, w)$  and not only those with  $x = 1^n$ .<sup>1</sup>

---

<sup>1</sup>In a sense, this is a check for your proof of (a). If your proof would work in this case, too, then your proof was not rigorous enough.

**Solution.** Consider  $V^*(x)$ :

- $V^*$  sends  $x$  to the prover  $P$ .
- $V^*$  receives  $y$  from  $P$ .
- $V^*$  outputs  $y$ .

Now we have the same situation as in part (b), except now  $x_n$  acts as sequence of auxiliary inputs  $z_n$  in (b). If  $x_n \in S_n$ ,  $(x_n, w_n) \in R$ , and  $(x_n, w'_n) \in R$  the interaction between  $P$  and  $V^*$  cannot be simulated by  $S$ . For  $R = \{(x, w) : x = 1^n\}$  (as in part (a)), an  $x$  does not reveal anything (except the length of  $x$ , which  $V^*$  learns anyway).

## Problem 2: Proofs of knowledge (10 points)

Show that the graph isomorphism proof system (Definition 4 in the lecture notes) is a proof of knowledge with knowledge error  $\kappa(n) = \frac{1}{2}$ .

**Hint:** Once you constructed an extractor  $K$ , you will need that this extractor is indeed a good one, i.e., you will need to lower bound its success probability in extracting a witness from a given prover  $P^*$  for some given statement  $x$ . To do so, you will find the following approach useful: Let  $R_1$  denote the random bits the prover  $P^*$  chooses in its first activation (these include the random bits used to produce the graph  $H$ ). Let  $T_r := \Pr[R_1 = r]$ , and  $S_r^i := \Pr[\langle P^*, V(x) \rangle = 1 | R_1 = r \text{ and } V \text{ sends } i]$ . Then  $\Pr[\langle P^*, V(x) \rangle = 1] = \sum_r T_r \cdot (S_r^1 + S_r^2)/2$ . Similarly, you will be able to express the probability  $\Pr[(x, w) \in R : w \leftarrow K^{P^*}(x)]$  in terms of  $T_r$  and  $S_r^i$ .

(A formula you might have to prove in your solution is  $S_r^1 S_r^2 \geq (S_r^1 + S_r^2)/2 - \frac{1}{2}$  for  $S_r^1, S_r^2 \in [0, 1]$ .)

**Solution.** We construct the extractor  $K$  in the following way:

- $K$  receives  $H$  from  $P^*$ .
- $K$  sends  $i = 1$  to  $P^*$  and receives  $\tilde{\phi}_1$  from  $P^*$ .
- $K$  rewinds  $P^*$  to the point before  $P^*$  receives  $i$ .
- Now  $K$  sends  $i = 2$  to  $P^*$  and receives  $\tilde{\phi}_2$  from  $P^*$ .

When the checks  $\tilde{\phi}_i(G_i) = H$  succeed for  $i = 1$  and  $i = 2$  we get:

$$\tilde{\phi}_1(G_1) = H \quad \text{and} \quad \tilde{\phi}_2(G_2) = H \quad \Rightarrow \quad \tilde{\phi}_2(G_2) = \tilde{\phi}_1(G_1) \quad \Rightarrow \quad G_2 = \underbrace{(\tilde{\phi}_2^{-1} \circ \tilde{\phi}_1)}_{\phi}(G_1)$$

Let  $T_r := \Pr[R_1 = r]$ , and  $S_r^i := \Pr[\langle P^*, V(x) \rangle = 1 | R_1 = r \text{ and } V \text{ sends } i]$  as in the hint. Then

$$\begin{aligned}
& \Pr[\langle P^*, V(x) \rangle = 1] \\
&= \sum_{r,i} \Pr[R_1 = r] \cdot \Pr[V \text{ sends } i] \cdot \Pr[\langle P^*, V(x) \rangle = 1 | R_1 = r \text{ and } V \text{ sends } i] \\
&= \sum_r \Pr[R_1 = r] \cdot \sum_i \frac{1}{2} \cdot \Pr[\langle P^*, V(x) \rangle = 1 | R_1 = r \text{ and } V \text{ sends } i] \\
&= \sum_r T_r \cdot \frac{1}{2} \cdot \sum_i \Pr[\langle P^*, V(x) \rangle = 1 | R_1 = r \text{ and } V \text{ sends } i] \\
&= \sum_r T_r \cdot \frac{1}{2} \cdot (S_r^1 + S_r^2) = \sum_r T_r \cdot \frac{S_r^1 + S_r^2}{2},
\end{aligned}$$

since  $V(x)$  is honest and chooses  $i \in \{1, 2\}$  uniformly at random. We write  $V_i(x)$  for the verifier  $V(x)$ , which sends  $i$  to the prover. Then:

$$\begin{aligned}
& \Pr[(x, w) \in R : w \leftarrow K^{P^*}(x)] \\
&= \sum_r \Pr[R_1 = r] \cdot \Pr[(x, w) \in R : w \leftarrow K^{P^*}(x) | R_1 = r] \\
&= \sum_r \Pr[R_1 = r] \cdot \Pr[\langle P^*, V_1(x) \rangle = 1, \langle P^*, V_2(x) \rangle = 1 | R_1 = r] \\
&= \sum_r \Pr[R_1 = r] \cdot \Pr[\langle P^*, V_1(x) \rangle = 1 | R_1 = r] \cdot \Pr[\langle P^*, V_2(x) \rangle = 1 | R_1 = r] \\
&= \sum_r T_r \cdot S_r^1 \cdot S_r^2
\end{aligned}$$

Note that  $\langle P^*, V_1(x) \rangle = 1$  and  $\langle P^*, V_2(x) \rangle = 1$  are independent given  $R_1 = r$ .

Finally  $(P, V)$  is a proof of knowledge with knowledge error  $\kappa(n) = \frac{1}{2}$ , if

$$\begin{aligned}
& \Pr[(x, w) \in R : w \leftarrow K^{P^*}(x)] \geq \Pr[\langle P^*, V(x) \rangle = 1] - \kappa(n) \\
\Leftrightarrow & \sum_r T_r \cdot S_r^1 \cdot S_r^2 \geq \left( \sum_r T_r \cdot \frac{S_r^1 + S_r^2}{2} \right) - \frac{1}{2} \\
\Leftrightarrow & \sum_r T_r \cdot S_r^1 \cdot S_r^2 \geq \left( \sum_r T_r \cdot \frac{S_r^1 + S_r^2}{2} \right) - \left( \sum_r T_r \cdot \frac{1}{2} \right) \\
\Leftrightarrow & \sum_r T_r \cdot S_r^1 \cdot S_r^2 \geq \left( \sum_r T_r \cdot \left( \frac{S_r^1 + S_r^2}{2} - \frac{1}{2} \right) \right) \\
\Leftarrow & \forall r : S_r^1 \cdot S_r^2 \geq \frac{S_r^1 + S_r^2}{2} - \frac{1}{2}.
\end{aligned}$$

With  $S_r^1, S_r^2 \in [0, 1]$

$$\begin{aligned} & S_r^1 \cdot S_r^2 + (1 - S_r^1) \cdot (1 - S_r^2) \geq 0 \\ \Leftrightarrow & S_r^1 \cdot S_r^2 + 1 - S_r^1 - S_r^2 + S_r^1 \cdot S_r^2 \geq 0 \\ \Leftrightarrow & 2 \cdot S_r^1 \cdot S_r^2 \geq S_r^1 + S_r^2 - 1 \\ \Leftrightarrow & S_r^1 \cdot S_r^2 \geq \frac{S_r^1 + S_r^2}{2} - \frac{1}{2}, \end{aligned}$$

holds, which concludes the proof.

## References

- [GK90] Oded Goldreich and Hugo Krawczyk. Sparse pseudorandom distributions. In *CRYPTO'89*, volume 435 of *LNCS*, pages 113–127. Springer, 1990.
- [GK96] Oded Goldreich and Hugo Krawczyk. On the composition of zero-knowledge proof systems. *SIAM J. Comput.*, 25(1):169–192, 1996. Online available at <http://citeseer.ist.psu.edu/goldreich90composition.html>.

## Solution of Exercise Sheet 7

Out: Wed, Jan 7, 2008

Due: Fri, Jan 16, 2009, before noon

**Problem 1: Miscellaneous (3+3+2=8 Points)**

- (a) Fix a cyclic group  $G$  of prime order  $q$  and a generator  $g$  of  $G$ . Let  $R := \{(x, w) : x = g^w\}$ . Let  $N := \{0, \dots, \text{ord } G - 1\}$ .

Show that for  $(x, w) \in R$ , the following two games have the same output distribution:

$$r \stackrel{R}{\leftarrow} N, \quad s \stackrel{R}{\leftarrow} N, \quad a := g^s x^{-r}, \quad \text{return } (a, r, s) \quad (1)$$

$$b \stackrel{R}{\leftarrow} N, \quad a := g^b, \quad r \stackrel{R}{\leftarrow} N, \quad s := b + rw \pmod{q}, \quad \text{return } (a, r, s) \quad (2)$$

**Note:** This is the missing step in the proof that Schnorr's proof system is HVCZK. The preferred (cleanest, simplest, ...) way to show this is as a short sequence of games.

**Solution.** Let  $q$  be the order of  $G$ . With  $x = g^w$  and  $g$  generator of  $G$  we have

$$\begin{aligned} & b \stackrel{R}{\leftarrow} N, \quad a := g^b, \quad r \stackrel{R}{\leftarrow} N, \quad s := b + rw \pmod{q}, \quad \text{return } (a, r, s) \quad (2) \\ = & r \stackrel{R}{\leftarrow} N, \quad b \stackrel{R}{\leftarrow} N, \quad s := b + rw \pmod{q}, \quad a := g^b, \quad \text{return } (a, r, s) \\ = & r \stackrel{R}{\leftarrow} N, \quad s \stackrel{R}{\leftarrow} N, \quad b := s - rw \pmod{q}, \quad a := g^b, \quad \text{return } (a, r, s) \\ = & r \stackrel{R}{\leftarrow} N, \quad s \stackrel{R}{\leftarrow} N, \quad b := s - rw \pmod{q}, \quad a := g^{s-rw}, \quad \text{return } (a, r, s) \\ = & r \stackrel{R}{\leftarrow} N, \quad s \stackrel{R}{\leftarrow} N, \quad a := g^{s-rw}, \quad \text{return } (a, r, s) \\ = & r \stackrel{R}{\leftarrow} N, \quad s \stackrel{R}{\leftarrow} N, \quad a := g^s x^{-r}, \quad \text{return } (a, r, s) \quad (1) \end{aligned}$$

- (b) Assume that  $(P, V)$  is a  $\Sigma$ -protocol that has special soundness. Let  $N$  denote the set from which the verifier's message  $r$  is chosen. Show that  $(P, V)$  has soundness bound  $s \leq 1/|N|$ .

**Solution.** Consider for  $x \notin L_R$  the interaction between a malicious prover  $P^*$  and the honest verifier  $V$ . After  $P^*$  has sent message  $a$  to the verifier there is at most one message  $r$  such that upon receiving  $s$  from  $P^*$ ,  $V$  will accept. If there would be  $r$  and  $r'$  with  $r \neq r'$  such that  $V$  accepts both possible interactions we would have  $x \in L_R$  by special soundness of  $(P, V)$ . Hence for  $x \notin L_R$ ,  $V$  will send that particular  $r$  and accept at most with probability  $1/|N|$ .

- (c) In the definition of honest-verifier zero-knowledge we require that the simulator successfully simulates the *view* of the verifier (Definition 19 in the lecture notes). On the other hand, in the definition of zero-knowledge (Definition 14 in the lecture notes), we require the simulator to simulate the *output* of the verifier. In exercise sheet 3, problem 3, we have shown that for the definition of zero-knowledge, it does not matter whether we simulate the view or the output of the verifier.

In the case of honest-verifier zero-knowledge, however, this does not hold. In fact, if we require the simulator to simulate the *output* of the verifier, we get a useless definition of honest-verifier ZK. (Call this definition HVCZK'; formally, you get HVCZK' by removing the word *view* from Definition 19 in the lecture notes.)

Explain why this is the case by designing a very simple but completely insecure<sup>1</sup> proof system and show that your proof system is HVCZK'.

**Solution.** Consider the following proof system  $(P, V)$ :

- $P(x, w)$  sends  $w$  to the verifier
- $V(x)$  receives  $w$  and outputs 1

Since the honest verifier will always output 1 the simulator  $S(x)$  can simulate the interaction between  $P$  and  $V$  by simply outputting 1. Hence  $(P, V)$  is HVCZK'.

## Problem 2: Proofs of Knowledge for Pedersen-Commitments (4+1+3+4=12 Points)

Fix a cyclic group  $G$  of prime order and generators  $g$  and  $h$  of  $G$ . Let  $R := \{(x, (w_1, w_2)) : x = g^{w_1} h^{w_2}\}$ . Let  $N := \{0, \dots, \text{ord } G - 1\}$ .

- (a) Fill in the gaps in the following protocol so that it becomes a proof system that has perfect completeness, special soundness, and is special HVCZK.

- Prover's input:  $(x, (w_1, w_2)) \in R$ .
- Verifier's input:  $x$ .
- The prover chooses  $b_1, b_2 \xleftarrow{R} N$  and sends  $a := g^{b_1} h^{b_2}$  to  $V$ .
- The verifier chooses  $r \xleftarrow{R} N$  and sends  $r$  to  $P$ .
- The prover computes  $s_1 := \boxed{\phantom{0}}$  and  $s_2 := \boxed{\phantom{0}}$  and sends  $(s_1, s_2)$  to the verifier.
- The verifier checks whether  $\boxed{\phantom{0}}$ .

---

<sup>1</sup>I.e., the witness should be revealed even to an honest verifier.

**Solution.**

- Prover's input:  $(x, (w_1, w_2)) \in R$ .
- Verifier's input:  $x$ .
- The prover chooses  $b_1, b_2 \xleftarrow{R} N$  and sends  $a := g^{b_1} h^{b_2}$  to  $V$ .
- The verifier chooses  $r \xleftarrow{R} N$  and sends  $r$  to  $P$ .
- The prover computes  $s_1 := \mathbf{b}_1 + \mathbf{r}w_1 \pmod{\mathbf{q}}$  and  $s_2 := \mathbf{b}_2 + \mathbf{r}w_2 \pmod{\mathbf{q}}$  and sends  $(s_1, s_2)$  to the verifier.
- The verifier checks whether  $\mathbf{g}^{s_1} \mathbf{h}^{s_2} = \mathbf{a}x^r$ .

(b) Prove that  $(P, V)$  has perfect completeness.

**Solution.** We compute:

$$\begin{aligned} g^{s_1} h^{s_2} &= g^{b_1 + rw_1} h^{b_2 + rw_2} &= g^{b_1} g^{rw_1} h^{b_2} h^{rw_2} \\ ax^r &= g^{b_1} h^{b_2} g^{rw_1} h^{rw_2} &= g^{b_1} g^{rw_1} h^{b_2} h^{rw_2}. \end{aligned}$$

Hence for  $(x, (w_1, w_2)) \in R$ , the verifier's check  $g^{s_1} h^{s_2} = ax^r$  always succeeds.

(c) Prove that  $(P, V)$  has special soundness.

**Solution.** The protocol is a  $\Sigma$ -protocol. Given  $(a, r, (s_1, s_2))$  and  $(a, r', (s'_1, s'_2))$  with  $r, r' \in N$ ,  $r \neq r'$ , and  $x, a \in G$ , and  $g^{s_1} h^{s_2} = ax^r$ , and  $g^{s'_1} h^{s'_2} = ax^{r'}$  we get:

$$\begin{aligned} \frac{g^{s_1} h^{s_2}}{g^{s'_1} h^{s'_2}} &= \frac{x^r}{x^{r'}} \Rightarrow g^{s_1 - s'_1} h^{s_2 - s'_2} = x^{r - r'} \\ &\Rightarrow g^{(s_1 - s'_1)/(r - r')} h^{(s_2 - s'_2)/(r - r')} = x \\ &\Rightarrow w = \left( \frac{s_1 - s'_1}{r - r'}, \frac{s_2 - s'_2}{r - r'} \right) \text{ is a witness for } x. \end{aligned}$$

In the second step we used the fact that  $r \neq r'$  and that the multiplicative inverse to  $r - r' \neq 0$  exists modulo  $\text{ord } G$ .

(d) Prove that  $(P, V)$  is special HVCZK.

**Solution.** Consider the following simulator  $S(x, r)$ :

- $S$  picks  $s_1, s_2 \xleftarrow{R} N$
- $S$  computes  $a = g^{s_1} h^{s_2} / x^r$
- $S$  outputs  $(a, r, (s_1, s_2))$ .

We have for  $(x, (w_1, w_2)) \in R$

$$\begin{aligned}
& b_1, b_2 \stackrel{R}{\leftarrow} N, \quad a := g^{b_1} h^{b_2}, \quad s_1 := b + rw_1 \bmod q, \quad s_2 := b + rw_2 \bmod q, \\
& \quad \text{return } (a, r, (s_1, s_2)) \\
= & b_1, b_2 \stackrel{R}{\leftarrow} N, \quad s_1 := b + rw_1 \bmod q, \quad s_2 := b + rw_2 \bmod q, \quad a := g^{b_1} h^{b_2}, \\
& \quad \text{return } (a, r, (s_1, s_2)) \\
= & s_1, s_2 \stackrel{R}{\leftarrow} N, \quad b_1 := s - rw_1 \bmod q, \quad b_2 := s - rw_2 \bmod q, \quad a := g^{b_1} h^{b_2}, \\
& \quad \text{return } (a, r, (s_1, s_2)) \\
= & s_1, s_2 \stackrel{R}{\leftarrow} N, \quad b_1 := s - rw_1 \bmod q, \quad b_2 := s - rw_2 \bmod q, \\
& \quad a := g^{s_1 - rw_1} h^{s_2 - rw_2}, \quad \text{return } (a, r, (s_1, s_2)) \\
= & s_1, s_2 \stackrel{R}{\leftarrow} N, \quad a := g^{s_1 - rw_1} h^{s_2 - rw_2}, \quad \text{return } (a, r, (s_1, s_2)) \\
= & s_1, s_2 \stackrel{R}{\leftarrow} N, \quad a := g^{s_1} h^{s_2} x^{-r}, \quad \text{return } (a, r, (s_1, s_2))
\end{aligned}$$

Here the first game is the view of the interaction between  $P(x, w)$  and  $V_r$  and the last game is the output of the simulator  $S(x, r)$ . Here  $V_r$  is the honest verifier that always sends  $r$ .

**Note:** I called this problem “Proofs of Knowledge for Pedersen-Commitments” because  $c := g^m h^u$  is a so-called Pedersen-commitment on  $m$  (with unveil information  $u$ ). Given  $c$ ,  $m$  is information-theoretically hidden, but we can only unveil  $c$  to two different values  $m$  and  $m'$  if we can compute the discrete logarithm of  $h$  (with respect to the basis  $g$ ). Hence the scheme you constructed allows to prove that you know what message  $m$  is inside a Pedersen-commitment without revealing that message.

## Solution of Exercise Sheet 8

Out: Wed, Jan 14, 2008

Due: Fri, Jan 23, 2009, before noon

**Problem 1: An efficient equivocal trapdoor commitment**  
**(11=1+4+3+3 points)**

Let  $G$  be a sequence of cyclic groups of prime order  $q$  with generator  $g$ . (Where  $G, q, g$  depend implicitly on a security parameter and are known. We assume that it is efficiently possible to multiply and invert in the group, and that group elements have a representation whose length is polynomial in  $n$ .) Let  $Q := \{0, \dots, q-1\}$ .

**Definition 1 (Dlog-assumption)** For any nonuniform probabilistic polynomial-time algorithm  $A$ , the following probability is negligible:

$$\Pr[g^x = g^{x'} : x \xleftarrow{R} Q, x' \leftarrow A(1^n, g^x)].$$

We define an equivocal trapdoor commitment scheme  $C$  with message space  $M_n = Q$  as follows (Pedersen commitments):

- $KeyGen(1^n)$  chooses  $td \xleftarrow{R} Q$ ,  $h := g^{td}$  and returns  $(h, td)$ . (I.e.,  $h$  is the CRS.)
- $Com(1^n, h, m)$  chooses  $u \xleftarrow{R} Q$  and returns  $(g^m h^u, u)$ .
- $Verify(1^n, h, m, c, u)$  checks whether  $m \in Q$  and  $c = g^m h^u$ .
- $Equiv(1^n, td, c, \tilde{u}, m) := \boxed{\phantom{0}}$ .

(a) Show that  $C$  has the correctness property.

**Solution.**

- Case  $m \in Q$ :  $Verify = 1$ , since  $m \in Q$  and  $Com$  returns  $c = g^m h^u$  such that the check of  $Verify$  succeeds.
- Case  $m \notin Q$ :  $Verify = 0$ , since  $m \notin Q$ .

(b) Show that  $C$  is computationally binding if the dlog-assumption holds.

**Hint:** Show that from  $(c, m, u, m', u')$  with  $Verify(1^n, h, m, c, u) = 1$  and  $Verify(1^n, h, m', c, u') = 1$  you can compute  $td$ . Replace  $h$  by  $g^{td}$  wherever it occurs.

**Solution.** Similarly to Schnorr's protocol we can compute  $td$  given  $(c, m, u, m', u')$  with  $Verify(1^n, h, m, c, u) = 1$  and  $Verify(1^n, h, m', c, u') = 1$  and  $m \neq m'$ :

$$\begin{aligned}
c &= g^m h^u = g^{m'} h^{u'} \\
\Rightarrow & g^m g^{td \cdot u} = g^{m'} g^{td \cdot u'} \\
\Rightarrow & m + td \cdot u \equiv m' + td \cdot u' \pmod{q} \\
\Rightarrow & td \equiv \frac{m' - m}{u - u'} \pmod{q}
\end{aligned}$$

Note that  $u \neq u'$ , otherwise  $g^m h^u \neq g^{m'} h^{u'}$  for  $m \neq m'$ . If the prover could unveil  $c$  to  $m$  and  $m'$  then we could transform him into a polynomial time algorithm computing  $td$  from  $g^{td}$  and hence the dlog assumption would not hold.

(c) Fill in the . (I.e., specify *Equiv*.)

**Solution.** We want to compute  $u$  such that  $g^0 h^{\tilde{u}} = g^m h^u$ :

$$\begin{aligned}
& g^0 h^{\tilde{u}} = g^m h^u \\
\Rightarrow & g^{td \cdot \tilde{u}} = g^{m + td \cdot u} \\
\Rightarrow & td \cdot \tilde{u} \equiv m + td \cdot u \pmod{q} \\
\Rightarrow & \tilde{u} \equiv m \cdot td^{-1} + u \pmod{q} \\
\Rightarrow & u \equiv \tilde{u} - m \cdot td^{-1} \pmod{q}
\end{aligned}$$

Hence  $Equiv(1^n, td, c, \tilde{u}, m) := \tilde{u} - m \cdot td^{-1} \pmod{q}$ . Note that  $\Pr[td = 0 : td \stackrel{R}{\leftarrow} Q]$  is negligible.

(d) Show that  $C$  (with your variant of *Equiv*) is equivocal.

**Solution.** The following games are computationally indistinguishable:

$$\begin{aligned}
& (c, u) : (h, td) \leftarrow KeyGen(1^n), (c, \tilde{u}) \leftarrow Com(1^n, h, 0), u \leftarrow Equiv(1^n, td, c, \tilde{u}, m) \\
& (c, u) : td \stackrel{R}{\leftarrow} Q, h := g^{td}, \tilde{u} \stackrel{R}{\leftarrow} Q, c := h^{\tilde{u}}, u := \tilde{u} - m \cdot td^{-1} \pmod{q} \\
& (c, u) : td \stackrel{R}{\leftarrow} Q, h := g^{td}, \tilde{u} \stackrel{R}{\leftarrow} Q, u := \tilde{u} - m \cdot td^{-1} \pmod{q}, c := h^{\tilde{u}} \\
& (c, u) : td \stackrel{R}{\leftarrow} Q, h := g^{td}, u \stackrel{R}{\leftarrow} Q, \tilde{u} := u + m \cdot td^{-1} \pmod{q}, c := h^{\tilde{u}} \\
& (c, u) : td \stackrel{R}{\leftarrow} Q, h := g^{td}, u \stackrel{R}{\leftarrow} Q, c := h^{u + m \cdot td^{-1} \pmod{q}} \\
& (c, u) : td \stackrel{R}{\leftarrow} Q, h := g^{td}, u \stackrel{R}{\leftarrow} Q, c := g^m h^u \\
& (c, u) : (h, td) \leftarrow KeyGen(1^n), (c, u) \leftarrow Com(1^n, h, m)
\end{aligned}$$

The first game is a cheating commit to  $m$  (using *Equiv*), while the last game is a normal commit to  $m$ .

## Problem 2: Equivocality vs. hiding (6 points)

Assume that  $C = (\text{KeyGen}, \text{Com}, \text{Verify}, \text{Equiv})$  is an equivocal trapdoor commitment scheme. Show that  $C$  satisfies the following computation hiding property:

**Definition 2 (Computationally hiding)** For all nonuniform probabilistic polynomial-time algorithms  $B^*$  there exists a negligible function  $\mu$  such that for all  $n \in \mathbb{N}$  and all  $m_1, m_2 \in M_n$  we have that

$$\left| \Pr[B^*(1^n, c) = 1 : (crs, td) \leftarrow \text{KeyGen}, (c, u) \leftarrow \text{Com}(1^n, crs, m_1)] - \Pr[B^*(1^n, c) = 1 : (crs, td) \leftarrow \text{KeyGen}, (c, u) \leftarrow \text{Com}(1^n, crs, m_2)] \right| \leq \mu(n).$$

(The only difference with respect to Definition 10 in the lecture notes is that we have included the CRS.)

**Hint:** Compare the following games: A normal commit to  $m_1$ . A cheating commit to  $m_1$  (using *Equiv*). A cheating commit to  $m_2$ . A normal commit to  $m_2$ . Use the fact that the adversary only sees the commitment  $c$  and never sees the unveil information  $u$ .

**Solution.** From equivocality of  $C$  we have that for all  $m \in M_n$ :

$$(c, u) : (crs, td) \leftarrow \text{KeyGen}(1^n), (c, \tilde{u}) \leftarrow \text{Com}(1^n, crs, 0), u \leftarrow \text{Equiv}(1^n, td, c, \tilde{u}, m)$$

and

$$(c, u) : (crs, td) \leftarrow \text{KeyGen}(1^n), (c, u) \leftarrow \text{Com}(1^n, crs, m)$$

are computationally indistinguishable. This implies that the following games are computationally indistinguishable:

$$\begin{aligned} c &: (crs, td) \leftarrow \text{KeyGen}(1^n), (c, u) \leftarrow \text{Com}(1^n, crs, m_1) \\ c &: (crs, td) \leftarrow \text{KeyGen}(1^n), (c, \tilde{u}) \leftarrow \text{Com}(1^n, crs, 0), u \leftarrow \text{Equiv}(1^n, td, c, \tilde{u}, m_1) \\ c &: (crs, td) \leftarrow \text{KeyGen}(1^n), (c, \tilde{u}) \leftarrow \text{Com}(1^n, crs, 0) \\ c &: (crs, td) \leftarrow \text{KeyGen}(1^n), (c, \tilde{u}) \leftarrow \text{Com}(1^n, crs, 0), u \leftarrow \text{Equiv}(1^n, td, c, \tilde{u}, m_2) \\ c &: (crs, td) \leftarrow \text{KeyGen}(1^n), (c, u) \leftarrow \text{Com}(1^n, crs, m_2) \end{aligned}$$

Hence the computational hiding property is fulfilled.

## Problem 3: High-speed protocols (3 points)

Design a CZK (not just special HVCZK!) argument of knowledge in the CRS model for showing that a given Pedersen commitment  $c$  is a commitment to the value  $m$  (without

actually revealing the unveil information). More exactly, show the *knowledge* of an unveil information for  $c$  that unveils  $c$  as  $m$ .<sup>1</sup> Specify the relation  $R$  explicitly.

Your scheme should be efficient and only use three exponentiations on the prover's side and five exponentiations on the verifier's side.

**Hint:** Note that  $u$  unveils  $c$  as  $m$  if and only if  $c/g^m = h^u$ . You only have to plug together known results from the lecture and from exercise sheets to solve this problem.

**Solution.** Let  $G$  be a cyclic group with fixed generators  $g, h$  of prime order  $q$ . For a group element  $a \in G$ , let  $\tilde{a}$  denote the representation of  $a$  as a nonnegative integer. (E.g., if  $G = \mathbb{Z}_n^\times$ , then  $\tilde{a} = a \bmod n$ . For other groups, e.g., elliptic curves, one might have to pick an arbitrary encoding.) Let  $q' := \max_{a \in G} \tilde{a} + 1$ . In most cases,  $q'$  will be close to  $q$ .

The relation for a proof of knowledge of an unveil information  $u$  that unveils  $c$  as  $m$  is

$$R = \{((c, m), w) \mid g^m h^w = c\} = \{((c, m), w) \mid h^w = c g^{-m}\}.$$

Consider now the relation used in Schnorr's protocol:

$$R' = \{(x, w) \mid h^w = x\}.$$

Note that  $(c, w) \in R$  iff  $(c g^{-m}, w) \in R'$ . Hence, it is sufficient to construct a CZK argument of knowledge  $(P', V')$  for  $R'$ . Then  $(P, V)$  is a CZK argument of knowledge for  $R$  if  $P((c, m), w)$  runs  $P'(c g^{-m}, w)$ , and  $V(c, m)$  runs  $V'(c g^{-m})$ .

To apply Damgård's construction, we need an equivocal commitment scheme that has a message space that is large enough to store the value  $a \in G$  sent by  $P'$ . Any such  $a$  is encoded as a number  $\tilde{a} < q'$ . Hence let  $H$  be a cyclic group of prime order  $p \geq q'$ . We will use Pedersen commitments based on the group  $H$ . Let  $P := \{0, \dots, p-1\}$ .

Applying Damgård's construction using Pedersen commitments to Schnorr's protocol we obtain the following protocol  $(P', V')$  in the CRS model:

- *KeyGen*: Choose random generators  $\bar{g}, \bar{h}$  of  $H$ .
- The prover  $P'$  chooses  $b \xleftarrow{R} Q$ ,  $u \xleftarrow{R} P$ , and computes  $a := g^b$  and  $d := \bar{g}^{\tilde{a}} \bar{h}^u$ . He sends  $d$  to the verifier  $V'$ .
- After the verifier receives  $d$ , he picks  $r \xleftarrow{R} Q$  and sends  $r$  to  $P'$ .
- Now the prover computes  $s := b + r w \bmod q$  and sends  $(s, a, u)$  to the verifier.
- The verifier checks whether  $a \in G$  and  $d = \bar{g}^{\tilde{a}} \bar{h}^u$  and  $g^s = a x^r$ .

Since  $(P, V)$  execute  $(P', V')$  with  $x := c g^{-m}$ , we get the following final protocol:

- *KeyGen*: Choose random generators  $\bar{g}, \bar{h}$  of  $H$ .

---

<sup>1</sup>This is another example where a ZK proof would not be enough: Since Pedersen commitments are equivocal, there always *exists* an unveil information for any value  $m$ . It's the knowledge that counts.

- The prover  $P'((c, m), w)$  chooses  $b \xleftarrow{R} Q$ ,  $u \xleftarrow{R} P$ , and computes  $a := g^b$  and  $d := \bar{g}^{\bar{a}} \bar{h}^u$ . He sends  $d$  to the verifier  $V'$ .
- After the verifier  $V'(c, m)$  receives  $d$ , he picks  $r \xleftarrow{R} Q$  and sends  $r$  to  $P'$ .
- Now the prover computes  $s := b + rw \pmod q$  and sends  $(s, a, u)$  to the verifier.
- The verifier checks whether  $a \in G$  and  $d = \bar{g}^{\bar{a}} \bar{h}^u$  and  $g^s = ac^r g^{-rm}$ .

The prover performs three exponentiations  $(g^b, \bar{g}^{\bar{a}}, \bar{h}^u)$ . The verifier performs five exponentiations  $(\bar{g}^{\bar{a}}, \bar{h}^u, g^s, c^r, g^{-rm})$ .

## Solution of Exercise Sheet 9

Out: Fri, Jan 23, 2008

Due: Fri, Jan 30, 2009, before noon

### Problem 1: Combining zero-knowledge proofs: AND (5+6=11 Points)

In the lecture, we saw a construction that takes two special HVCZK  $\Sigma$ -protocols with special soundness  $(P_1, V_1)$  and  $(P_2, V_2)$  (for relations  $R_1$  and  $R_2$ ) and constructs a new  $\Sigma$ -protocol  $(P_{OR}, V_{OR})$  for the relation  $R_1 \vee R_2 = \{((x_1, x_2), (i, w)) : i \in \{1, 2\}, (x_i, w) \in R_i\}$  (again with special soundness and special HVCZK).

In this problem, we are interested in constructing a proof system  $(P_{AND}, V_{AND})$  for the relation  $R_{AND} := R_1 \wedge R_2 = \{((x_1, x_2), (w_1, w_2)) : (x_1, w_1) \in R_1, (x_2, w_2) \in R_2\}$ .

Assume for the following, that  $(P_1, V_1)$  and  $(P_2, V_2)$  both are special HVCZK  $\Sigma$ -protocols with special soundness and perfect completeness.<sup>1</sup>

- (a) The following construction immediately springs to mind: On input  $((x_1, x_2), (w_1, w_2)) \in R_{AND}$ , the prover  $P_{AND}$  constructs  $a_1$  and  $a_2$  by running the provers  $P_1(x_1, w_1)$  and  $P_2(x_2, w_2)$ , respectively. He sends  $(a_1, a_2)$  to the verifier. Then the verifier  $V_{AND}$  picks  $r_1 \xleftarrow{R} N_1$  and  $r_2 \xleftarrow{R} N_2$  (as the verifiers  $V_1(x_1)$  and  $V_2(x_2)$  would have done) and sends  $(r_1, r_2)$  to the prover. Then the prover computes answers  $s_1$  and  $s_2$  by passing  $r_1$  and  $r_2$  to  $P_1$  and  $P_2$ , respectively, and sends  $(s_1, s_2)$  to the verifier. The verifier checks whether  $V_1$  would accept the interaction  $(a_1, r_1, s_1)$  and  $V_2$  would accept the interaction  $(a_2, r_2, s_2)$ . If so, the verifier accepts.

Show that with this construction, the  $\Sigma$ -protocol  $(P_{AND}, V_{AND})$  does *not*, in general, have special soundness.

**Note:** The protocol  $(P_{AND}, V_{AND})$  is, however, a proof of knowledge, even if it does not have special soundness. But since, for example, the OR construction assumes special soundness, we could not use  $(P_{AND}, V_{AND})$  as a subprotocol in that construction.

**Solution.** The definition of special soundness requires that an algorithm  $E$  is able to extract a witness  $w = (w_1, w_2)$  for  $x = (x_1, x_2)$  given two interactions  $(a, r, s)$  and  $(a, r', s')$  with  $r \neq r'$ , which the verifier would accept. For the construction from above these interactions would be  $((a_1, a_2), (r_1, r_2), (s_1, s_2))$  and  $((a_1, a_2), (r'_1, r'_2), (s'_1, s'_2))$  with  $r_1 \neq r'_1$  **or**  $r_2 \neq r'_2$  (since  $(r_1, r_2) \neq (r'_1, r'_2)$  implies  $r_1 \neq r'_1$  or  $r_2 \neq r'_2$ ). However having only  $r_1 \neq r'_1$  we are able to extract a witness  $w_1$  for  $x_1$ , but not a

<sup>1</sup>The latter assumption is not really necessary, but we add it for simplicity.

witness for  $x_2$ , and vice versa. Hence in general the construction from above does not have special soundness.

- (b) Give a construction for  $(P_{AND}, V_{AND})$  such that the resulting protocol is special HVCZK and has special soundness and perfect completeness. Prove that it has these properties.

**Hint:** Let  $V_{AND}$  send only one value.

**Solution.** We construct  $(P_{AND}, V_{AND})$  as in (a) except the verifier sends only one  $r \xleftarrow{R} N$ . After receiving  $r$  the prover computes the answers  $s_1$  and  $s_2$  by passing  $r$  both to  $P_1$  and  $P_2$ . Finally the verifier checks whether  $V_1$  would accept the interaction  $(a_1, r, s_1)$  and  $V_2$  would accept the interaction  $(a_2, r, s_2)$ .

- Perfect completeness follows from the construction and the fact that  $(P_1, V_1)$  and  $(P_2, V_2)$  have perfect completeness.
- Special soundness: Given two interactions  $((a_1, a_2), r, (s_1, s_2))$  and  $((a_1, a_2), r', (s'_1, s'_2))$  with  $r \neq r'$ , which the verifier  $V_{AND}$  would accept. We construct  $E$  as follows:

$$\begin{aligned} E((x_1, x_2), (a_1, a_2), r, (s_1, s_2), r', (s'_1, s'_2)) \\ := (E_1(x_1, a_1, r, s_1, r', s'_1), E_2(x_2, a_2, r, s_2, r', s'_2)). \end{aligned}$$

Since  $(P_1, V_1)$  and  $(P_2, V_2)$  have special soundness  $E_1$  and  $E_2$  exist.  $E_1$  extracts a witness for  $x_1$  from interactions  $(a_1, r, s_1)$  and  $(a_1, r', s'_1)$ . Similarly  $E_2$  extracts a witness for  $x_2$  from interactions  $(a_2, r, s_2)$  and  $(a_2, r', s'_2)$ . The interactions would be accepted by  $V_1$  and  $V_2$ , respectively, since  $V$  would accept  $((a_1, a_2), r, (s_1, s_2))$  and  $((a_1, a_2), r', (s'_1, s'_2))$ . Hence  $E$  extracts a witness for  $(x_1, x_2)$ .

- Special HVCZK: We construct a simulator  $S$  for  $(P_{AND}, V_{AND})$  as follows:
  - $S$  runs the simulator  $S_1$  for  $(P_1, V_1)$  and obtains  $(a_1, r, s_1)$ ,
  - $S$  runs the simulator  $S_2$  for  $(P_2, V_2)$  and obtains  $(a_2, r, s_2)$ ,
  - $S$  outputs  $((a_1, a_2), r, (s_1, s_2))$ .

The following games are computationally indistinguishable:

$$\begin{aligned} (a_1, r, s_1) \leftarrow S_1(x_1, r), \quad (a_2, r, s_2) \leftarrow S_2(x_2, r), \quad \text{return } ((a_1, a_2), r, (s_1, s_2)) \\ a_1 \leftarrow P_1(x_1, w_1), \quad s_1 \leftarrow P_1(r), \quad a_2 \leftarrow P_2(x_2, w_2), \quad s_2 \leftarrow P_2(r), \\ \text{return } ((a_1, a_2), r, (s_1, s_2)) \\ a_1 \leftarrow P_1(x_1, w_1), \quad a_2 \leftarrow P_2(x_2, w_2), \quad s_1 \leftarrow P_1(r), \quad s_2 \leftarrow P_2(r), \\ \text{return } ((a_1, a_2), r, (s_1, s_2)) \end{aligned}$$

Here the first game is the simulator  $S$  and the last game is the view of the interaction between the prover and the verifier. In the first step we used the fact that  $(P_1, V_1)$  and  $(P_2, V_2)$  are special HVCZK. In the second step we reordered terms.

## Problem 2: Combining zero-knowledge proofs: OR (6(+5)=6 Points)

In the lecture, we saw a construction that takes two special HVCZK  $\Sigma$ -protocols with special soundness  $(P_1, V_1)$  and  $(P_2, V_2)$  (for relations  $R_1$  and  $R_2$ ) and constructs a new  $\Sigma$ -protocol  $(P_{OR}, V_{OR})$  for the relation  $R_1 \vee R_2 = \{((x_1, x_2), (i, w)) : i \in \{1, 2\}, (x_i, w) \in R_i\}$  (again with special soundness and special HVCZK).

Prove that in general, if  $(P_1, V_1)$  and  $(P_2, V_2)$  do not have special soundness, but if instead they are special HVCZK proofs of knowledge with knowledge error 0, negligible soundness error, and perfect completeness (i.e., the next best thing), that then  $(P_{OR}, V_{OR})$  is completely insecure (i.e., a malicious prover could convince  $V_{OR}$  of a wrong statement with probability 1).

**Hint:** Construct a protocol  $(P_1, V_1)$  where only the first half of the message  $r$  sent by the verifier is actually used. (The second half being random bits that are sent to  $P$  but ignored.) Similarly for  $(P_2, V_2)$ .

**Solution.** Given a  $\Sigma$ -protocol  $(P, V)$  which is special HVCZK, has knowledge error 0, and perfect completeness we construct  $(P_i, V_i)$  in the following way:

- $P_i(x_i, w_i)$  obtains  $a_i \leftarrow P(x_i, w_i)$  and sends it to  $V_i$ .
- $V_i(x_i)$  chooses  $r_1, r_2 \xleftarrow{R} N$  and sends  $r_1 || r_2$  to  $P_i$  (here  $a || b$  denotes the concatenation of  $a$  and  $b$ ).
- $P_i$  receives  $r = r_1 || r_2$  from  $V_i$ , obtains  $s_i \leftarrow P(r_i)$ , and sends  $s_i$  to  $V_i$ .
- $V_i$  outputs  $V(a_i, r_i, s_i)$ .

$(P_i, V_i)$  is a special HVCZK proof of knowledge with knowledge error 0, has a negligible soundness error, and perfect completeness.

Let  $x_1 \notin L_{R_1}$  and  $x_2 \notin L_{R_2}$ . Now consider the following malicious prover  $P^*$  for the proof system  $(P', V')$  resulting from applying the OR construction to  $(P_1, V_1)$  and  $(P_2, V_2)$ :

- $P^*$  chooses  $\tilde{r}_1, \tilde{r}_2 \xleftarrow{R} N$ .
- $P^*$  obtains  $(a_1, \tilde{r}_1, s_1) \leftarrow S(x_1, \tilde{r}_1)$  and  $(a_2, \tilde{r}_2, s_2) \leftarrow S(x_2, \tilde{r}_2)$ , where  $S$  is the simulator for  $(P, V)$ .
- $P^*$  sends  $(a_1, a_2)$  to  $V'$  and receives  $r = r_1 || r_2$ .
- Finally  $P^*$  computes  $r'_1 := (\tilde{r}_1 || (r_2 - \tilde{r}_2))$ , and  $r'_2 := ((r_1 - \tilde{r}_1) || \tilde{r}_2)$ , and sends  $(r'_1, r'_2, s_1, s_2)$  to the verifier.

We have that

$$\begin{aligned}
r'_1 + r'_2 &= (\tilde{r}_1 \parallel (r_2 - \tilde{r}_2)) + ((r_1 - \tilde{r}_1) \parallel \tilde{r}_2) \\
&= (\tilde{r}_1 + r_1 - \tilde{r}_1) \parallel (r_2 - \tilde{r}_2 + \tilde{r}_2) \\
&= r_1 \parallel r_2 = r
\end{aligned}$$

and  $V'(a_i, r'_i, s_i) = (V_1(a_1, \tilde{r}_1, s_1) \wedge V_2(a_2, \tilde{r}_2, s_2)) = 1$ , since  $(a_i, \tilde{r}_i, s_i)$  with  $i \in \{1, 2\}$  has been generated by the simulator  $S$  with overwhelming probability. Hence the verifier  $V'$  accepts with overwhelming probability.

Hence the constructed proof system  $(P_{OR}, V_{OR})$  for  $R_1 \vee_{M_1, M_2} R_2$  is insecure, since  $P^*$  has convinced  $V$  of a false statement.

**Bonus points:** Try to make a counterexample that shows that the construction from Problem 1(a) is not suited for use in the OR-construction. More exactly, let  $(P, V)$  a special HVCZK with special soundness and perfect completeness for relation  $R$ . Let  $(P_{AND}, V_{AND})$  be the protocol for the relation  $R \wedge R$  (using the construction from Problem 1(a)), and let  $(P_{OR}, V_{OR})$  be the protocol for the relation  $(R \wedge R) \vee (R \wedge R)$  resulting from applying the OR construction from the lecture to  $(P_{AND}, V_{AND})$ . Show that  $(P_{OR}, V_{OR})$  is insecure (i.e., a malicious prover could convince  $V_{OR}$  of a wrong statement with probability 1).

**Solution.** The proof system  $(P_{OR}, V_{OR})$  for  $(R \wedge R) \vee (R \wedge R) = \{(x_1, x_2, x_3, x_4), (i, w_1, w_2) \mid i \in \{1, 2\}, (x_{2i-1}, w_1) \in R, (x_{2i}, w_2) \in R\}$  looks as follows:

- Prover  $P_{OR}$ 's input:  $((x_1, x_2, x_3, x_4), (i, w_1, w_2))$ .
- Verifier  $V_{OR}$ 's input:  $(x_1, x_2, x_3, x_4)$ .
- Assume  $i = 1$ , in the case  $i = 2$  the prover's code is analogous and the verifier's code is identical.
- The prover computes  $a_1 \leftarrow P_1(x_1, w_1), a_2 \leftarrow P_2(x_2, w_2), r_3 \xleftarrow{R} N, r_4 \xleftarrow{R} N, (a_3, r_3, s_3) \leftarrow S_3(x_3, r_3), (a_4, r_4, s_4) \leftarrow S_4(x_4, r_4)$  where  $S_3$  and  $S_4$  are the special HVCZK simulators for  $(P_3, V_3)$  and  $(P_4, V_4)$ .
- The prover sends  $(a_1, a_2, a_3, a_4)$  to the verifier.
- The verifier picks  $r'_1 \xleftarrow{R} N, r'_2 \xleftarrow{R} N$  and sends  $(r'_1, r'_2)$  to the prover.
- The prover computes  $r_1 := r'_1 - r_3, r_2 := r'_2 - r_4, s_1 \leftarrow P_1(r_1), s_2 \leftarrow P_2(r_2)$ , and sends  $(r_1, r_2, r_3, r_4, s_1, s_2, s_3, s_4)$  to the verifier.
- The verifier checks that  $r_1, r_2, r_3, r_4 \in N, r'_1 = r_1 + r_3, r'_2 = r_2 + r_4$  and that for  $i \in \{1, 2, 3, 4\}$ ,  $V_i$  would accept  $(a_i, r_i, s_i)$ .

The proof system  $(P_{OR}, V_{OR})$  is insecure as the following example shows:

- Assume the malicious prover  $P^*$  has a witness  $w_1$  for  $x_1$  and  $w_2$  for  $x_4$ , but  $x_2, x_3 \notin L_R$ .
- $P^*$  computes  $a_1 \leftarrow P_1(x_1, w_1), a_4 \leftarrow P_4(x_4, w_2), r_2 \xleftarrow{R} N, r_3 \xleftarrow{R} N, (a_2, r_2, s_2) \leftarrow S_2(x_2, r_2), (a_3, r_3, s_3) \leftarrow S_3(x_3, r_3)$ .
- $P^*$  sends  $(a_1, a_2, a_3, a_4)$  to the verifier.
- After receiving  $(r'_1, r'_2)$  from the verifier, the prover computes  $r_3 := r'_1 - r_1, r_2 := r'_2 - r_4, s_2 \leftarrow P_2(r_2), s_3 \leftarrow P_3(r_3)$ , and sends  $(r_1, r_2, r_3, r_4, s_1, s_2, s_3, s_4)$  to the verifier.

The verifier behaves as above, but although  $(x_1, x_2, x_3, x_4) \notin L_{(R \wedge R) \vee (R \wedge R)}$  all checks of the verifier succeed and  $V$  accepts.

### Problem 3: A few thoughts on simulators (3(+5)=3 Points)

In most cases, we are interested in zero-knowledge protocols in which the prover and the verifier are very efficient (because in the final application, the user will have to wait longer if the prover/verifier are less efficient). Since the simulator is only a technical tool for describing what it means that the witness does not leak, the running time of the simulator is usually not considered to be that important (as long as it is polynomial-time).

- (a) In some cases, however, one would be interested in efficient simulators. Why?

**Note:** This is not a deep question.

**Solution.** Construction 2 for combining two zero-knowledge proofs with OR makes usage of the simulator in the construction of the prover. For the result of the construction being efficient, the simulator used in it should be efficient.

- (b) For bonus points: There is another reason why one would want simulators not to be too slow (e.g., running time  $\Omega(n^{100})$  where  $n$  is the running time of the adversary), even if one never intends to actually run it. Namely, if the simulator runs very long, the bit lengths of the primitives used in your protocols (e.g., the size of the group we work in, the length of keys, etc.) needs to be larger to get suitable security guarantees. Explain why this is the case.

**Hint:** It might help to think about the following example: Assume that computing the discrete logarithm in a group  $G$  with generator  $g$  takes at least  $\text{ord } G$  steps.<sup>2</sup> Assume a zero-knowledge proof of knowledge  $(P, V)$  for  $R = \{(x, w) : g^w = x\}$ . Alice's secret is some  $s \in \{0, \dots, \text{ord } G - 1\}$ , and the public information is  $p := g^s$ .

---

<sup>2</sup>This is actually not true, because the so-called baby-step giant-step algorithm find the discrete logarithm in  $O(\sqrt{\text{ord } G})$  steps. But for the sake of illustration, it is simpler to assume this bound.

To identify herself, Alice proves using  $P$  that she knows some  $s$  with  $p = g^s$ . Bob, who is only given  $g$  and  $p$  has negligible probability of succeeding in impersonating Alice (i.e., in convincing the verifier  $V$  of statement  $p$ ). But for actually using this scheme, we need to know what the size of  $G$  should be. For example, given an adversary Bob that runs at most  $2^{80}$  steps, how large does  $\text{ord } G$  have to be such that the probability of Bob impersonating Alice is small?

**Note: I think this is a difficult problem.**

**Solution.** For simplicity, assume that Bob, running in  $2^{80}$  steps, succeeds in guessing  $s$  after interacting with Alice (where Bob plays the role of the prover).<sup>3</sup> Then there is a simulator  $S$  for Bob that also guesses  $s$  (given input  $p$ ), i.e., the simulator breaks the discrete logarithm problem in  $G$ . The simulator runs in  $2^{8000}$  steps. Since we assume that breaking the discrete logarithm takes at least  $\text{ord } G$  steps, we only get a contradiction if  $\text{ord } G \gg 2^{8000}$ , i.e., for key lengths greater 8000 bit. On the other hand, if the simulator's running time would have been  $O(n)$ , a key length of about 80 bit would have been sufficient. Since the key length directly impacts the efficiency of the scheme, it follows that an inefficient simulator can lead to less efficient schemes if one takes concrete security bounds into account.

---

<sup>3</sup>For more general impersonation attacks the argumentation would be similar.

## Solution of Exercise Sheet 10

Out: Wed, Jan 28, 2008

Due: Fri, Feb 6, 2009, before noon

**Problem 1: Witness indistinguishability (2+4+4=10 Points)**

Zero-knowledge is a very strong requirement, because we need to be able to actually produce a simulation of the proof in polynomial time (this is in particular difficult in some cases of protocol composition). The following definition is less demanding, but still useful in many applications. It only requires that one cannot distinguish, which of several possible witnesses is used.

**Definition 1 (Non-interactive witness-indistinguishable proof)** A triple  $(P, V, KeyGen)$  of polynomial-time algorithms is called a non-interactive computationally witness-indistinguishable (NICWI) proof in the CRS-model for a relation  $R$  if the following holds:

- Completeness and soundness: As in Definition 25 in the lecture notes.
- Computational witness-indistinguishability: The distributions

$$\left\{ \begin{array}{l} (crs, \pi) : \\ crs \leftarrow KeyGen(1^{|x|}), \\ \pi \leftarrow P(x, crs, w_1) \end{array} \right\}_{x, (w_1, w_2, z)} \quad \text{and} \quad \left\{ \begin{array}{l} (crs, \pi) : \\ crs \leftarrow KeyGen(1^{|x|}), \\ \pi \leftarrow P(x, crs, w_2) \end{array} \right\}_{x, (w_1, w_2, z)}$$

are computationally indistinguishable for all  $(x, w_1, w_2)$  with  $(x, w_1) \in R$  and  $(x, w_2) \in R$  and  $z \in \{0, 1\}^*$ .

- (a) Let  $f$  be a one-way permutation. Let  $R := \{(x, w) : x = f(w)\}$ . The  $P(x, crs, w)$  outputs the proof  $\pi := w$ . Then verifier  $V(x, crs, \pi)$  checks whether  $x = f(\pi)$ .  $KeyGen$  always returns the empty string. Show that  $(P, V, KeyGen)$  is a NICWI proof.

**Note:** This shows that the NICWI property is not very useful if the witness is determined by the statement. However, if there are several witnesses, NICWI can be a useful property.

**Solution.** Since  $f$  is a one-way permutation for each  $x$  there exists exactly one  $w$  with  $x = f(w)$ . We have that

$$\left\{ \begin{array}{l} (\varepsilon, w_1) : \\ \varepsilon \leftarrow KeyGen(1^{|x|}), \\ w_1 \leftarrow P(x, crs, w_1) \end{array} \right\}_{x, (w_1, w_2, z)} \quad \text{and} \quad \left\{ \begin{array}{l} (\varepsilon, w_2) : \\ \varepsilon \leftarrow KeyGen(1^{|x|}), \\ w_2 \leftarrow P(x, crs, w_2) \end{array} \right\}_{x, (w_1, w_2, z)}$$

are computationally indistinguishable, since  $(x, w_1) \in R$  and  $(x, w_2) \in R$  implies  $w_1 = w_2$ . Furthermore  $(P, V, KeyGen)$  has perfect completeness and perfect soundness. Hence  $(P, V, KeyGen)$  is a NICWI proof.

- (b) Assume that  $(P, V, KeyGen)$  is a NICZK proof in the CRS-model. Show that it is a NICWI proof in the CRS model.

**Note:** This even holds if the simulator of  $(P, V, KeyGen)$  is not efficient.

**Solution.** Since  $(P, V, KeyGen)$  is a NICZK proof there exists a simulator  $S$  such that

$$\left\{ \begin{array}{l} (crs, \pi) : \\ crs \leftarrow KeyGen(1^{|x|}), \\ \pi \leftarrow P(x, crs, w) \end{array} \right\}_{x, (w, z)} \approx_c \left\{ S(x) \right\}_{x, (w, z)}$$

This implies that

$$\left\{ \begin{array}{l} (crs, \pi) : \\ crs \leftarrow KeyGen(1^{|x|}), \\ \pi \leftarrow P(x, crs, w_1) \end{array} \right\}_{x, (w_1, w_2, z)} \approx_c \left\{ S(x) \right\}_{x, (w_1, w_2, z)}$$

and

$$\left\{ S(x) \right\}_{x, (w_1, w_2, z)} \approx_c \left\{ \begin{array}{l} (crs, \pi) : \\ crs \leftarrow KeyGen(1^{|x|}), \\ \pi \leftarrow P(x, crs, w_2) \end{array} \right\}_{x, (w_1, w_2, z)}$$

Hence a NICWI proof in the CRS model is computationally witness-indistinguishable. Furthermore completeness and soundness for the NICWI proof follows from completeness and soundness of the NICZK proof. Hence if  $(P, V, KeyGen)$  is a NICZK proof, it is also a NICWI proof.

- (c) Assume that  $(P, V, KeyGen)$  is a NICWI proof. Show that

$$\left\{ (crs, \pi_1, \dots, \pi_s) \right\}_{(x_1, \dots, x_s), (w_1, w'_1, \dots, w_s, w'_s, z)} \quad \text{and} \quad \left\{ (crs, \pi'_1, \dots, \pi'_s) \right\}_{(x_1, \dots, x_s), (w_1, w'_1, \dots, w_s, w'_s, z)}$$

are computationally indistinguishable for all  $(x_1, \dots, x_s), (w_1, w'_1, \dots, w_s, w'_s)$  with  $(x_i, w_i), (x_i, w'_i) \in R$  for all  $i$  and  $|x_1| = \dots = |x_s|$  and all  $z \in \{0, 1\}^*$ . (And  $s$  being polynomially bounded in  $|x_1|$ .)

Here  $crs \leftarrow KeyGen(1^{|x_1|})$  and  $\pi_i \leftarrow P(x_i, crs, w_i)$  and  $\pi'_i \leftarrow P(x_i, crs, w'_i)$ .

**Note:** All proofs use the *same* CRS. Hence this property is analogous to the multi-theorem NIZK property in Problem 2.

**Hint:** Use a hybrid argument.

**Solution.** Let  $crs, \pi_i, \pi'_i$  defined as above. We write short  $\underline{\pi}$  for  $\pi_1, \dots, \pi_s$ , and  $\underline{w}, \underline{w}'$  analogously. Assume there exists a distinguisher  $D$  that distinguishes  $(crs, \pi_1, \dots, \pi_s)$  and  $(crs, \pi'_1, \dots, \pi'_s)$  with non-negligible probability:

$$\nu := \left| \Pr[D(crs, \underline{\pi}, \underline{w}, \underline{w}', z) = 1] - \Pr[D(crs, \underline{\pi}, \underline{w}, \underline{w}', z) = 1] \right|.$$

We define the hybrid game  $H_i := (crs, \pi'_1, \dots, \pi'_i, \pi_{i+1}, \dots, \pi_s, \underline{w}, \underline{w}', z)$ . Hence  $H_0 = (crs, \underline{\pi}, \underline{w}, \underline{w}', z)$  and  $H_s = (crs, \underline{\pi}', \underline{w}, \underline{w}', z)$ .

We have

$$\begin{aligned} \nu &= \left| \Pr[D(H_0) = 1] - \Pr[D(H_s) = 1] \right| \\ &= \left| \sum_{i=0}^{s-1} (\Pr[D(H_i) = 1] - \Pr[D(H_{i+1}) = 1]) \right| \\ &\leq \sum_{i=0}^{s-1} \left| \Pr[D(H_i) = 1] - \Pr[D(H_{i+1}) = 1] \right|. \end{aligned}$$

Thus there is an  $i$  such that  $|\Pr[D(H_i) = 1] - \Pr[D(H_{i+1}) = 1]| \geq \frac{\nu}{s}$ . By  $\underline{w}_{\neq i}$  we denote  $w_1, \dots, w_{i-1}, w_{i+1}, \dots, w_s$ . Let  $z' := (\underline{w}_{\neq i}, \underline{w}'_{\neq i}, i, z)$ . Let  $D'$  be constructed such that upon input  $(crs, \pi_i, w_i, z')$  it runs  $D(H_i)$  ( $D'$  needs to produce the proofs  $\underline{\pi}_{\neq i}$  and  $\underline{\pi}'_{\neq i}$  on its own by running the prover on  $\underline{w}_{\neq i}, \underline{w}'_{\neq i}$ ). Then  $|\Pr[D'(crs, \pi_i, w_i, z') = 1] - \Pr[D'(crs, \pi_i, w_i, z') = 1]| = |\Pr[D(H_i) = 1] - \Pr[D(H_{i+1}) = 1]| \geq \frac{\nu}{s}$ . Since  $\nu/s$  is non-negligible, this is a contradiction to the assumption that  $(P, V, KeyGen)$  is a NICWI proof.

## Problem 2: Multi-theorem NIZK proofs (2+1+2+5=10 Points)

The definition of NICZK as presented in the lecture (Definition 25 in the lecture notes) only guarantees that the simulator can simulate a single proof. In particular, if the prover produces two proofs with respect to the *same* CRS, there is no guarantee that the simulator will be able to produce a simulation of both proofs simultaneously. Hence the following definition:

**Definition 2 (Multi-theorem NICZK)** A triple  $(P, V, KeyGen)$  of polynomial-time algorithms is called a multi-theorem non-interactive computational zero-knowledge proof in the CRS-model for a relation  $R$  if the following holds:

- Completeness and soundness: As in Definition 25 in the lecture notes.
- Multi-theorem computational zero-knowledge: There is a polynomial-time algorithm  $S$  such that

$$\left\{ \begin{array}{l} (crs, \pi_1, \dots, \pi_s) : \\ crs \leftarrow KeyGen(1^{|x_1|}), \\ \pi_i \leftarrow P(x_i, crs, w_i) \text{ for } i = 1, \dots, s \end{array} \right\}_{(x_1, \dots, x_s), (w_1, \dots, w_s, z)}$$

and

$$\left\{ S(x_1, \dots, x_s) \right\}_{(x_1, \dots, x_s), (w_1, \dots, w_s, z)}$$

are computationally indistinguishable for all  $(x_1, \dots, x_s, w_1, \dots, w_s)$  with  $(x_i, w_i) \in R$  for all  $i = 1, \dots, s$  and  $|x_i| = |x_j|$  for all  $i, j = 1, \dots, s$ . (As long as  $s$  is polynomially bounded in  $|x_1|$ .)

The following construction allows to transform NICZK proofs into multi-theorem NICZK proofs:

**Construction 1 (FLS-technique<sup>1</sup>)** Let  $R$  be an NP-relation. Let  $G : \{0, 1\}^n \rightarrow \{0, 1\}^{2n}$  be a pseudo-random generator. Let  $R' := \{((x, c), w) : (x, w) \in R \text{ or } c = G(w)\}$ . Assume that  $(P', V', KeyGen')$  is a NICZK proof for  $R'$  in the CRS-model. We construct a proof system  $(P, V, KeyGen)$  for  $R$  as follows:

- $KeyGen(1^{|x|})$  computes  $crs' \leftarrow KeyGen'(1^n)$ ,  $c \xleftarrow{R} \{0, 1\}^{2n}$ ,  $crs := (crs', c)$ . It returns  $crs$ .
  - $P(x, crs, w)$  sets  $(crs', c) := crs$  and computes  $\pi \leftarrow P'((x, c), crs', w)$ . It returns  $\pi$ .
  - $V(x, crs, \pi)$  sets  $(crs', c) := crs$  and checks whether  $V'((x, c), crs', \pi)$  accepts.
- (a) Show that NICZK proof for HC presented in the lecture<sup>2</sup> is *not* multi-theorem NICZK unless  $NP \subseteq BPP$ .

**Hint:** Let  $x_1$  be a graph with  $n$  vertices. Let  $x_2$  be a graph with  $n$  vertices that consists only of a single cycle. Show that from proofs for  $x_1 \in L_R$  and  $x_2 \in L_R$  with respect to the same CRS, you can extract a HC of  $x_1$ .

**Solution.** If we have proofs for  $x_1 \in L_R$  and  $x_2 \in L_R$  with respect to the same CRS we know that in both proofs the graphs  $x_1$  and  $x_2$  have the same Hamiltonian cycle  $\tilde{H}$  the prover is bound to by the CRS. In the second proof  $P$  sends  $\pi_2, I_2$  with  $(i, j) \notin I_2 \Leftrightarrow (i, j) \in \pi_2(x_2)$ . Since  $x_2$  consists only of a single cycle, all edges missing in  $I_2$  belong to a Hamiltonian cycle  $\pi_2(\tilde{H})$  of  $\pi_2(x_2)$ . Hence  $\{\pi_2^{-1}(i, j) \mid (i, j) \notin I_2\}$  is the set of edges of a Hamiltonian cycle of  $x_1$ .

- (b) Show the completeness of  $(P, V, KeyGen)$  from Construction 1.

**Solution.**  $(P, V, KeyGen)$  has overwhelming completeness, since  $(P', V', KeyGen')$  has overwhelming completeness and  $V(x, crs, \pi) = V'((x, c), crs', \pi)$ , and  $(x, w) \in R$  implies  $((x, c), w) \in R'$

- (c) Show the soundness of  $(P, V, KeyGen)$  from Construction 1.

<sup>1</sup>The technique of proving  $X \vee Y$  where  $X$  is the statement we actually want to prove and  $Y$  is something that cannot be proven is often called the FLS-technique after Fiat, Lapidot, Shamir. It is a very powerful technique with many different applications.

<sup>2</sup>The one with inefficient prover resulting from applying Construction 3 in the lecture notes to the hidden-bit NICZK proof for HC.

**Solution.** Assume  $x \notin L_R$ . Since the range of  $G$  has size  $2^{2n}$ , and the domain of  $G$  only size  $2^n$ , with overwhelming probability  $c$  is not in the domain of  $G$ . Hence with overwhelming probability,  $(x, c) \notin L_{R'}$ . From the soundness of  $(P', V')$  it then follows that the prover can only convince the verifier  $V'(x, c) = V'(x)$  with negligible probability.

(d) Show that  $(P, V, KeyGen)$  from Construction 1 is multi-theorem zero-knowledge.

**Hint:** The simulator  $S(x_1, \dots, x_n)$  produces the CRS as follows:  $crs' \leftarrow KeyGen'(1^{|x|})$ ,  $r \xleftarrow{R} \{0, 1\}^n$ ,  $c := G(r)$ ,  $crs := (crs', c)$ . To produce a proof  $\pi_i$  for  $x_i$ , the simulator runs  $\pi_i \leftarrow P'((x_i, c), crs, r)$ . (That is, the simulator does not actually prove that  $x_i \in L_R$ , but instead that a certain trapdoor  $r$  exists in the CRS.) Use the results from Problem 1 in your proof.

**Solution.** We need to show that the simulator  $S$  sketched in the hint is a good simulator for  $(P, V)$ . Unfolding the definitions of  $S$  and of  $P$ , this means that we have to show that the following are computationally indistinguishable:

$$\left\{ \begin{array}{l} (crs, \pi_1, \dots, \pi_s) : \\ crs' \leftarrow KeyGen'(1^{|x_1|}), c \xleftarrow{R} \{0, 1\}^{2n}, crs := (crs', c), \\ \pi_i \leftarrow P'((x_i, c), crs', w_i) \text{ for } i = 1, \dots, s \end{array} \right\}_{(x_1, \dots, x_s), (w_1, \dots, w_s, z)} \quad \text{and} \quad (1)$$

$$\left\{ \begin{array}{l} (crs, \pi_1, \dots, \pi_s) : \\ crs' \leftarrow KeyGen'(1^{|x_1|}), r \xleftarrow{R} \{0, 1\}^n, \\ c := G(r), crs := (crs', c), \\ \pi_i \leftarrow P'((x_i, c), crs', r) \text{ for } i = 1, \dots, s \end{array} \right\}_{(x_1, \dots, x_s), (w_1, \dots, w_s, z)}$$

Note that by Problem 1 (b),  $(P', V')$  is NICWI. Hence by Problem 1 (c), and since both  $w_i$  and  $r$  are valid witnesses for  $(x_i, c)$ , the indistinguishability (1) follows.

## Solution of Exercise Sheet 11

Out: Wed, Feb 4, 2008

Due: Fri, Feb 13, 2009, before noon

**Problem 1: Dolev-Dwork-Naor (10 Points)**

Consider the construction of  $(K', E', D')$  presented in the lecture (see also Construction 4 in the lecture notes). Assume that we change it as follows: In the encryption  $E'$ , we produce  $\sigma$  as  $\sigma \leftarrow \text{Sig}(\text{sigk}, (c_1, \dots, c_n))$  instead of  $\sigma \leftarrow \text{Sig}(\text{sigk}, (c_1, \dots, c_n, \pi))$ . (That is, we do not sign the NICZK proof.) And the check in the decryption is modified accordingly, i.e., we check whether  $\text{Verify}(vk, \sigma, (c_1, \dots, c_\ell)) = 1$  instead of checking  $\text{Verify}(vk, \sigma, (c_1, \dots, c_\ell, \pi)) = 1$ .

Show that this modified construction is not secure in general.

**Hint:** Let  $(K, E, D)$  be an IND-CPA secure encryption scheme. Let  $(KS, \text{Sig}, \text{Verify})$  be a strongly unforgeable one-time signature scheme. Let  $(\text{KeyGen}', P', V')$  be an unbounded adaptive NICZK argument with perfect completeness. Construct  $(\text{KeyGen}, P, V)$  from  $(\text{KeyGen}', P', V')$  by appending a bit of useless information to the proof. Construct an adversary that breaks the CCA2-game for  $(K', E', D')$ .

**Solution.** Let  $\text{KeyGen} := \text{KeyGen}'$ .  $P(x, crs, w)$  runs  $\pi' \leftarrow P'(crs, x, w)$  and returns  $\pi := \pi' \| 0$ .  $V(x, crs, \pi)$  strips the last bit off  $\pi$  (resulting in  $\pi'$ ) and runs  $V'(x, crs, \pi')$ .

Note that if  $\pi \| 0$  is accepted by the verifier  $V$ , then  $\pi \| 1$  is, too.

We construct an adversary  $A$  for the IND-CCA2 game for  $(K', E', D')$ . First,  $A$  sets  $m_0 := 0$ ,  $m_1 := 1$  and asks for a decryption of  $(m_0, m_1)$ . He gets  $c^* := E'(pk, m_b)$  where  $b \in \{0, 1\}$  is random. The adversary parses  $c^*$  as  $(c_1^*, \dots, c_n^*, \pi^*, \sigma^*, vk^*)$ . Since  $\pi^*$  has been constructed by  $P$ , it is of the form  $\pi^* = \pi' \| 0$ . Let  $\tilde{\pi} := \pi' \| 1$ . Let  $c := (c_1^*, \dots, c_n^*, \tilde{\pi}, \sigma^*, vk^*)$ . Then  $c \neq c^*$ . Thus  $A$  can ask for a decryption of  $c$  and gets  $D'(sk, c)$ . Since  $\sigma^*$  is a signature over  $c_1^*, \dots, c_n^*$ , but not over  $\pi^*$ , the signature is still valid. Furthermore,  $\tilde{\pi}$  is accepted by  $V$ . Hence the decryption algorithm accepts and decrypts  $c$  and returns  $m_b = b$ . Thus the adversary guesses  $b$  with probability 1 and breaks the IND-CCA2 game.