T-79.511 Special Course on Cryptology / Zero Knowledge: Statistical Difference is Complete for Honest-Verifier Statistical Zero Knowledge*

Petteri Kaski

January 22, 2002

This manuscript is a continuation of an earlier manuscript on statistical zero knowledge written for the seminar on zero knowledge at HUT.

1 Introduction

The discovery of complete problems for a complexity class typically not only results in greater understanding of the class and the problems that lie in it, but also often provides new insight and a surprising alternative characterization of the class. The complexity classes defined via interactive proof systems [4] provide two good examples of such alternative characterizations.

The first surprise came when Shamir [9] proved[1] that **IP**, the class of languages that admit interactive proof systems, has complete problems, which established that the class coincides with **PSPACE**, the class of languages decidable using a polynomially space-bounded Turing machine. This result not only established that interactive proofs can prove much more than classical proofs (assuming $\mathbf{NP} \subsetneq \mathbf{PSPACE}$), but also gave a static characterization of **IP** without any reference to interaction or proof protocols. Furthermore, given widely believed complexity-theoretic assumptions, Shamir's completeness result extends to *computational zero-knowledge* proof systems [2, 5, 7].

The second surprise came when Sahai and Vadhan [10], building on top of results of Okamoto [8], proved that the class of languages that admit *(honest-verifier) statistical zero-knowledge* proof systems has a complete problem that makes no reference to either zero-knowledge or interactive protocols. Namely, the problems that admit (HV)SZK proof systems can be reduced to STATISTICAL DIFFERENCE, the problem of testing whether the statistical difference between two distributions encoded by Boolean circuits is either large or small. Later, a similar **HVSZK**-complete problem, ENTROPY DIFFERENCE, was discovered [6].

Salil Vadhan's PhD thesis [11] is probably the best reference to consult for the relevance of STATISTICAL DIFFERENCE and ENTROPY DIFFERENCE to the study of statistical zero-knowledge proofs. (Indeed, this manuscript is mostly based on Vadhan's thesis.) For our purposes it suffices to know for motivation that the complete problems provide relatively easy access to nontrivial results about statistical zero knowledge and its relationship to other complexity classes. Examples of such results include Okamoto's result [8] that **HVSZK** is closed under complement and the results of Fortnow [3] and Aiello and Håstad [1] that $\mathbf{HVSZK} \subseteq \mathbf{AM} \cap \mathbf{coAM}$.

The present manuscript is devoted to sketching the **HVSZK**-completeness proof of STATISTICAL DIFFERENCE. Specifically, we present a polynomial-time computable reduction from an arbitrary problem that admits a public-coin HVSZK proof system to (the complement of) STATISTICAL DIFFERENCE.

The other necessary ingredient for the HVSZK completeness proof, a HVSZK proof system for STATISTICAL DIFFERENCE, was presented in an earlier manuscript written for the HUT seminar.

We chose to restrict the treatment to public-coin

*Revised version; thanks to Johan [Wallén] for comments.

[1] For an enlightening background story on how this result was conceived, see G. Kolata. "In a Frenzy, Math Enters Age of Electronic Mail." *The New York Times*, Tuesday, June 26, 1990, page C1.

protocols because the required simulator analysis is more straightforward in the public-coin case. In particular, only elementary properties of statistical difference are required without the need for more advanced concepts (such as entropy and relative entropy required for a discussion of ENTROPY DIFFERENCE). Furthermore, the restriction to public coins does not restrict generality: a public-coin HVSZK protocol can always be transformed to a private-coin HVSZK protocol [6, 8, 11].

2 Informal ideas

At first it may seem controversial that an arbitrary problem instance decidable by an HVSZK proof system can be transformed to deciding the magnitude of statistical difference between two distributions. Moreover, this transformation must be computable in polynomial time. After all, the prover in an interactive protocol has no resource limitations, apart from its replies being polynomially bounded in the problem instance size. So, in the least the transformation must somehow avoid the "hard aspects" of a problem that would normally require interaction with the prover, and transform them into a question involving statistical difference.

The key observation was made by Fortnow [3] (and subsequently refined by Aiello and Håstad [1]), who realized that YES and NO instances of a problem can be distinguished by observing the probabilistic polynomial time *simulator*.

The informal ideas behind the simulator analysis are as follows. Recall that the task of the simulator in a ZK proof system is to demonstrate that the verifier "learns nothing" from an interaction because it cannot distinguish between interactions with the real prover and the output of the simulator. For statistical zero-knowledge this means that the simulator output distribution and the distribution induced by interacting with the real prover have a negligible statistical difference. Specifically, this condition is required from YES instances of the problem. Recall further that the properties of the proof system dictate that the prover must succeed in convincing the verifier to accept most of the time on YES instances. On the other hand, the prover's success probability must be low no matter what strategy is used on NO instances.

So, the statistical closeness requirement for YES instances enforces the simulator output distribution to satisfy:

1. The probability that the simulator outputs an accepting transcript must match the acceptance probability of the real protocol.

2. The *virtual* verifier in a transcript output by the simulator must behave with high probability exactly as the real verifier would behave.

If either of the conditions is violated, then intuitively the simulator output is no longer statistically close the distribution induced by real interactions.

To distinguish between YES and NO instances, we must now demonstrate that the behaviour of the simulator differs noticeably from the above for NO instances. At first this seems futile, since *nothing* is explicitly required from the simulator's output distribution for NO instances.

The key observation is that 1 and 2 cannot both occur at once because this would violate the *soundness* of the interactive proof system. More specifically, suppose that both 1 and 2 hold for a NO instance. Condition 2 says that the virtual verifier behaves as the real verifier would behave if given as input the replies produced by the virtual prover in the simulated protocol run. But this means that we can use the simulator's output distribution (conditioned on messages already exchanged) in determining meaningful prover replies—Condition 1 then ensures that this prover will succeed with high probability in convincing the verifier to accept.

So, at least one of the following must hold for NO instances:

3. The simulator outputs an accepting transcript with low probability.

4. The behaviour of the virtual verifier differs noticeably from the behaviour of the real verifier.

We have now informally illustrated a difference between the simulator output distributions for YES and NO instances. All that remains now is to give a rigorous foundation to the informal analysis conducted. In other words, we shall derive a polynomial-time transformation that exploits the difference in simulator behaviour, and produces from a problem instance two distributions (encoded by Boolean circuits) whose statistical difference is either large (a NO instance of the original problem) or small (a YES instance of the original problem).

3 Terminology and tools

Before presenting the reduction in all detail, we must first review the terminology and the technical results required. We will assume most of the definitions from the earlier survey. Nevertheless, we recall the basic assumptions and notation for the reader's convenience.

All random variables take values in the set of finite binary strings, $\{0,1\}^*$, unless explicitly noted otherwise. If X, Y are random variables, then $\Delta(X,Y)$ denotes their statistical difference.

We require two technical results on statistical difference to simplify the analysis in the next section. Suppose X_1 and Y_1 are random variables taking values in an at most countable universe $\mathcal{U} = \{u_1, u_2, \dots\}$. Similarly, let X_2 and Y_2 take values in $\mathcal{V} = \{v_1, v_2, \dots\}$. Define the random variables $X_1 \otimes X_2$ and $Y_1 \otimes Y_2$ over $\mathcal{U} \times \mathcal{V}$ by setting $\Pr(X_1 \otimes X_2 = (u_i, v_j)) \stackrel{\text{def}}{=} \Pr(X_1 = u_i)\Pr(X_2 = v_j)$ and $\Pr(Y_1 \otimes Y_2 = (u_i, v_j)) \stackrel{\text{def}}{=} \Pr(Y_1 = u_i)\Pr(Y_2 = v_j)$ for all i, j.

Lemma 3.1

$$\Delta(X_1 \otimes X_2, Y_1 \otimes Y_2) \le \Delta(X_1, Y_1) + \Delta(X_2, Y_2).$$

Proof. For convenience, put $x_i^{(1)} \stackrel{\text{def}}{=} \Pr(X_1 = u_i)$, $x_j^{(2)} \stackrel{\text{def}}{=} \Pr(X_2 = v_j)$, and define $y_i^{(1)}$, $y_j^{(2)}$ similarly for Y_1, Y_2. By definition of statistical difference, we have

$$\begin{aligned}
&2\Delta(X_1 \otimes X_2, Y_1 \otimes Y_2) = \\
&\quad = \sum_{i,j} |x_i^{(1)} x_j^{(2)} - y_i^{(1)} y_j^{(2)}| \\
&\quad = \sum_{i,j} |x_i^{(1)} x_j^{(2)} - y_i^{(1)} x_j^{(2)} + y_i^{(1)} x_j^{(2)} - y_i^{(1)} y_j^{(2)}| \\
&\quad \le \sum_{i,j} |x_i^{(1)} x_j^{(2)} - y_i^{(1)} x_j^{(2)}| + |y_i^{(1)} x_j^{(2)} - y_i^{(1)} y_j^{(2)}| \\
&\quad = \sum_{i,j} |x_i^{(1)} - y_i^{(1)}| x_j^{(2)} + y_i^{(1)} |x_j^{(2)} - y_j^{(2)}| \\
&\quad = 2\Delta(X_1, Y_1) + 2\Delta(X_2, Y_2).
\end{aligned}$$

■

In other words, the statistical difference of two distributions obtained by repeated independent sampling is upper bounded by the sum of statistical differences of the individual distributions.

The Direct Product Lemma is a special case of the above result in which also a lower bound can be established using the Chernoff bound.

Lemma 3.2 (Direct Product Lemma) *Let X, Y be distributions such that $\Delta(X,Y) = \delta$. Then, for all $k \in \mathbb{N}$,*

$$1 - 2e^{-k\delta^2/2} \le \Delta(\otimes^k X, \otimes^k Y) \le k\delta.$$

(We omit the proof; see e.g. [11, Lemma 3.1.15].)

The second technical result that we require is that applying the same randomized procedure F to two distributions X, Y cannot increase their statistical difference. More specifically, suppose that X and Y take values in $\mathcal{U}$ and that $\mathcal{F}$ is an at most countable set of functions from $\mathcal{U}$ to $\mathcal{V}$. Let F be a random variable that takes values $\mathcal{F}$. Define the random variable $F(X)$ over $\mathcal{V}$ by setting

$$\begin{aligned}
\Pr(F(X) = z) &\stackrel{\text{def}}{=} \sum_x \Pr(F(x) = z)\Pr(X = x) \\
&= \sum_x \sum_{\substack{f \in \mathcal{F} \\ f(x) = z}} \Pr(F = f)\Pr(X = x)
\end{aligned}$$

for all $z \in \mathcal{V}$.

Lemma 3.3 $\Delta(F(X), F(Y)) \le \Delta(X, Y)$.

Proof.

$$\begin{aligned}
&2\Delta(F(X), F(Y)) = \\
&= \sum_z |\Pr(F(X) = z) - \Pr(F(Y) = z)| \\
&= \sum_z |\sum_x \Pr(F(x) = z)(\Pr(X = x) - \Pr(Y = x))| \\
&= \sum_x \sum_z \Pr(F(x) = z)|\Pr(X = x) - \Pr(Y = x)| \\
&\le 2\Delta(X, Y),
\end{aligned}$$

where the last inequality follows from $\sum_z \Pr(F(x) = z) = 1$. ■

A Boolean circuit C with n inputs and m outputs induces a probability distribution (which is by abuse of notation also denoted by C) on $\{0,1\}^m$, where

$$\Pr(C = y) \stackrel{\text{def}}{=} 2^{-n}|\{x \in \{0,1\}^n \ : \ C(x) = y\}|$$

for all $y \in \{0,1\}^m$.

The promise problem STATISTICAL DIFFERENCE is to decide for two given Boolean circuits X, Y whether the statistical difference $\Delta(X, Y)$ is either

- at least $2/3$, which is a YES instance;
- at most $1/3$, which is a NO instance;
- (between $1/3$ and $2/3$, in which case any answer is acceptable since the promise is violated).

Our conventions and notatation for interactive proof systems are as follows. Let $\Pi = (\Pi_{\text{yes}}, \Pi_{\text{no}})$ be a promise problem, and suppose (P, V) is a honest-verifier public-coin statistical zero-knowledge proof system for Π. Let S be the corresponding (strictly) polynomial time simulator with negligible simulator deviation μ tied to the instance length $|x|$. In other words, on input $x \in \{0,1\}^*$ the distribution induced by the verifier interacting with the prover $\langle P, V\rangle(x)$ and the output distribution of the simulator $S(x)$ have statistical difference at most $\mu(|x|)$. The failure probability of the simulator is assumed to be at most $1/2$. The completeness and soundness errors are assumed to be at most $1/3$.

The verifier's (partial) view $\gamma = (m_1, m_2, \ldots, m_t; r)$ of an interaction consists of all the messages m_i (so far) exchanged between the prover and the verifier, and the random coins r used by the verifier. By convention, odd-numbered messages are sent by the prover to the verifier, and even-numbered messages constitute the verifier's messages to the prover. Since we assume that (P, V) is public-coin, the messages sent by the verifier to the prover constitute the random coins used by the prover. More specifically, $r = r_1 r_2 \cdots r_t$, where $m_{2i} = r_i$ and $|m_{2i}| = \ell_i(x; m_1, \ldots, m_{2i-1})$ for all $1 \leq i \leq t/2$. Since the verifier is polynomial time bounded, the functions ℓ_i are polynomial time computable. For convenience we assume that the protocol always terminates by the verifier sending its decision YES / NO to the prover. Furthermore, since the verifier's messages to the prover completely determine the random coins accessed by the verifier during an interaction, we omit them from explicit consideration.

We assume that $v(|x|)$ denotes the polynomial bound on the messages sent by the verifier to the prover. The verifier's final message is not included in the bound, that is, $t \leq 2v(|x|) + 2$.

Subscripted random variables $\langle P, V\rangle(x)_i$ and $S(x)_i$ denote the random variables obtained from $\langle P, V\rangle(x)_i$ and $S(x)_i$ by truncating an outcome transcript to the first i messages exchanged.

Finally, whenever a problem instance x can be understood from the context, we shall drop it from the notation for convenience.

4 The reduction

In this section we finally present the reduction from an arbitrary problem instance $x \in \{0,1\}^*$ to two Boolean circuits X, Y that encode the desired distributions. (The reduction is in fact to the complement of STATISTICAL DIFFERENCE, that is, YES instances of Π are transformed to NO instances of STATISTICAL DIFFERENCE and vice versa.)

The intuition in the construction is that distribution X (and its soon-to-be-defined components X_j) represent the situation in which both properties 1 and 2 are made to hold by force. The distribution Y and its components Y_j is then constructed from the actual simulator so that if either property 3 or 4 holds, then this is observable as statistical difference between X and Y.

The difference between properties 1 and 3 is easily transformed into statistical difference by running the simulator multiple times on input x and estimating the probability of transcripts that accept. However, there is a (technical) catch.

The catch is that the simulator is strictly polynomial time, and thus has a certain failure probability. Yet the output distribution of such a simulator is conditioned on nonfailure, so the failing of the simulator complicates the analysis. Luckily, we can always run the simulator multiple times to guarantee at least one nonfailing run with very high probability. In particular, by running the simulator N times and outputting the first successful run, we obtain a simulator with exactly the same output distribution and failure probability at most 2^{-N}. So, given that the subsequent analysis leaves some room for simulator failure, we may safely ignore the fact that the simulator may fail, since its failure probability can always be made so small that it does not affect the analysis in any significant way. (In other words, we will *not* discuss the case when the simulator fails and trust that the meticulous reader is able to convince him- or herself that the analysis still goes through without significant changes even if the simulator is allowed to fail with very small probability.)

The first component distributions (Boolean circuits) X_0 and Y_0 are now defined as follows:

- X_0 always outputs 1 (and hence represents the situation in which the simulator nearly always outputs accepting transcripts).
- Y_0 encodes the computation that runs the simulator on input x for $216 \ln 12v(|x|)$ independent executions, and outputs 1 if the verifier accepts in the majority of the transcripts obtained; 0 is output otherwise.

The intuition is that these component distributions ensure that the statistical difference between X and Y is large if property 3 holds.

It remains to distinguish between properties 2 and 4. The restriction to public coins pays off here, since it is very straightforward to say what constitutes "identical behaviour" for the virtual verifier in the public coin case. Namely, if the (simulated) message history is $\gamma_{2i-1} = (m_1, \ldots, m_{2i-1})$, then the virtual verifier behaves as the real public-coin verifier if its subsequent messages are distributed almost uniformly on $\{0,1\}^{\ell_i(x;\gamma_{2i-1})}$. The distributions X_i and Y_i, which are defined using Boolean circuits that encode the following computations, exploit this observation.

X_i: Run S on input x to obtain a transcript γ, and denote by γ_{2i-1} the first $2i-1$ messages exchanged. Compute $\ell_i = \ell_i(x;\gamma_{2i-1})$. Select $r \in \{0,1\}^{\ell_i}$ uniformly at random. Output (γ_{2i-1}, r).

Y_i: Run S on input x to obtain a transcript γ, and denote by γ_{2i} the first $2i$ messages exchanged. Output γ_{2i}.

The Y_i represent the actual behaviour of the virtual verifier, whereas the X_i represent what the virtual verifier should do if it is behaving identically to the real verifier. Intuitively, any difference in behaviour results in statistical difference that can be amplified to noticeable levels.

In what follows we give a more rigorous analysis of the component distributions and build the final distributions X and Y from the components.

Central in the analysis is a *virtual prover* extracted from the simulator. The virtual prover P_S behaves as follows on common input $x \in \{0,1\}^*$. Let γ consist of the $2i$ messages exchanged so far in the protocol.

- If $S(x)$ outputs transcripts that begin with γ with probability 0, then the reply of P_S is arbitrary (but polynomially bounded).
- Otherwise, P_S replies with $\beta \in \{0,1\}^*$ with probability p_β that is determined by the simulator's output distribution conditioned on γ:

$$p_\beta \stackrel{\text{def}}{=} \Pr(S(x)_{2i+1} = (\gamma, \beta) | S(x)_{2i} = \gamma)$$

For convenience we drop the common input $x \in \{0,1\}^*$ from the notation in what follows.

The following lemma is required to upper bound the induced statistical difference on YES instances of the problem.

Lemma 4.1 *For every $i = 1, \ldots, v$,*

$$\Delta(X_i, Y_i) \leq 2\Delta(S, \langle P, V \rangle).$$

Proof. Since statistical difference respects the triangle inequality, we have

$$\Delta(X_i, Y_i) \leq \Delta(X_i, \langle P, V \rangle_{2i}) + \Delta(\langle P, V \rangle_{2i}, Y_i).$$

Since we have $Y_i = S_{2i}$ by definition, we must have

$$\Delta(Y_i, \langle P, V \rangle_{2i}) = \Delta(S_{2i}, \langle P, V \rangle_{2i}) \leq \Delta(S, \langle P, V \rangle),$$

where the last inequality follows from the fact that truncated protocol runs can only decrease the statistical difference since possibly distinct outcomes are merged into one by the truncation process. On the other hand, X_i is obtained from S_{2i-1} by applying the randomized procedure in which ℓ_i random bits are appended to the outcome. By definition of V, $\langle P, V \rangle_{2i}$ is obtained from $\langle P, V \rangle_{2i-1}$ by exactly the same procedure. Thus, by Lemma 3.3 we have

$$\begin{aligned}\Delta(\langle P, V \rangle_{2i}, X_i) &\leq \Delta(\langle P, V \rangle_{2i-1}, S_{2i-1}) \\ &\leq \Delta(\langle P, V \rangle, S),\end{aligned}$$

which completes the proof. ∎

The subsequent two lemmata lower bound the statistical difference on NO instances.

Lemma 4.2

$$\sum_{i=1}^{v} \Delta(X_i, Y_i) \geq \Delta(\langle P_S, V\rangle, S).$$

Proof. The proof is by induction on $j = 1, \ldots, v$. The induction hypothesis is

$$\Delta(\langle P_S, V\rangle_{2j}, S_{2j}) \leq \sum_{i=1}^{j} \Delta(X_i, Y_i).$$

The base case $j = 1$ follows from the definitions (that is, equality holds). Suppose that the IH holds for j and consider the case $j+1$. Now, $\langle P_S, V\rangle_{2j+1}$ is obtained from $\langle P_S, V\rangle_{2j}$ by an application of the same random procedure that is used to produce S_{2j+1} from S_{2j}. Thus,

$$\Delta(\langle P_S, V\rangle_{2j+1}, S_{2j+1}) \leq \Delta(\langle P_S, V\rangle_{2j}, S_{2j})$$

by Lemma 3.3. Since $X_{j+1} = S_{2j+2}$ by definition, we have

$$\begin{aligned} &\Delta(\langle P_S, V\rangle_{2j+2}, S_{2j+2}) = \\ &\quad = \Delta(\langle P_S, V\rangle_{2j+2}, X_{j+1}) \\ &\quad \leq \Delta(\langle P_S, V\rangle_{2j+2}, Y_{j+1}) + \Delta(Y_{j+1}, X_{j+1}). \end{aligned}$$

Now $\langle P_S, V\rangle_{2j+2}$ is obtained from $\langle P_S, V\rangle_{2j+1}$ by an application of the same random procedure that is used to produce Y_{j+1} from S_{2j+1}. Thus,

$$\begin{aligned} \Delta(\langle P_S, V\rangle_{2j+2}, Y_{j+1}) &\leq \Delta(\langle P_S, V\rangle_{2j+1}, S_{2j+1}) \\ &\leq \Delta(\langle P_S, V\rangle_{2j}, S_{2j}) \\ &\leq \sum_{i=1}^{j} \Delta(X_i, Y_i), \end{aligned}$$

where the second inequality appears earlier in the proof, and the last inequality applies the induction hypothesis. The inductive step is now completed by combining the two inequalities.

The claim is now obtained by observing that the final prover-to-verifier message m_{2v+1} and the subsequent verifier decision m_{2v+2} are now determined by applying the same[2] randomized procedure to both distributions. ■

[2] Strictly speaking, it is required here that the final decision that the simulated verifier makes is computed using the real verifier. However, it is obvious that this assumption can be made for any simulator S since it cannot increase the simulator deviation for YES instances.

Lemma 4.3 *Let p denote the probability that $S(x)$ outputs an accepting transcript, and let q be the least upper bound, taken over all provers P^*, for the probability that V accepts on x when interacting with P^*. Then,*

$$\Delta(\langle P_S, V\rangle, S) \geq p - q.$$

Proof. The inequality is an immediate consequence of definition of statistical difference. Namely, the set of transcripts in which the verifier accepts occurs with probability p in the simulator and with probability at most q in $\langle P_S, V\rangle$. ■

Using the derived lemmata, we shall next analyze the statistical difference between distributions

$$X' \stackrel{\text{def}}{=} X_0 \otimes X_1 \otimes \cdots \otimes X_v$$

and

$$Y' \stackrel{\text{def}}{=} Y_0 \otimes Y_1 \otimes \cdots \otimes Y_v$$

from which the final distributions X and Y are constructed (simply by independent sampling). (Recall that $X_0 \otimes X_1$ denotes the distribution induced by taking independent samples of X_0 and X_1. This distribution is realized in terms of Boolean circuits by simply gluing the circuits that represent X_0 and X_1 side by side to form a circuit with distinct inputs and outputs for both X_0 and X_1.) In the following analysis we assume that $\mu \leq 1/[4 \cdot (12v)^3]$. (Since μ is a negligible function, this assumption can evidently be made.)

First we establish that the statistical difference $\Delta(X', Y')$ is reasonably small for YES instances.

Lemma 4.4 *If $x \in \Pi_{\text{yes}}$, then*

$$\Delta(X', Y') \leq \frac{1}{12 \cdot (12v)^2}.$$

Proof. By Lemma 3.1 the statistical difference between X' and Y' is at most $\sum_{i=0}^{v} \Delta(X_i, Y_i)$. Since x is a YES instance, Lemma 4.1 gives

$$\Delta(X_i, Y_i) \leq 2\Delta(\langle P, V\rangle, S) \leq 2\mu \leq \frac{1}{2 \cdot (12v)^3}$$

for $i = 1, \ldots, v$. To bound $\Delta(X_0, Y_0)$, recall that the completeness error is at most $1/3$, so S must output accepting transcripts with probability at least $2/3 - \mu \geq 7/12$ for the simulator deviation to be within the required bound. So, from the Chernoff bound we obtain that Y_0 outputs 1 with probability at least

$$1 - \exp(-2 \cdot (216 \ln 12v) \cdot (1/12)^2) \geq 1 - \frac{1}{(12v)^3}.$$

Thus, the statistical difference between X_0 and Y_0 is at most $1/(12v)^3$, and the total statistical difference between X' and Y' is at most

$$v \cdot \frac{1}{2 \cdot (12v)^3} + \frac{1}{(12v)^3} \leq \frac{1}{12 \cdot (12v)^2}.$$

■

The statistical difference for NO instances is much larger in comparison.

Lemma 4.5 *If $x \in \Pi_{\mathrm{no}}$, then*

$$\Delta(X', Y') \geq \frac{1}{12v}.$$

Proof. It suffices to show that the statistical difference between some X_i and Y_i is at least $1/(12v)$, because the statistical difference between X' and Y' must by construction be at least $\Delta(X_i, Y_i)$. We split the analysis into two cases. First, suppose the simulator outputs accepting transcripts with probability at most $5/12$. Then, by the Chernoff bound, Y_0 outputs 1 with probability at most

$$\exp(-2 \cdot (216 \ln 12v) \cdot (1/12)^2) < 1/2,$$

so $\Delta(X_0, Y_0) > 1/2 \geq 1/(12v)$. On the other hand, suppose that the simulator outputs accepting transcripts with probability at least $5/12$. By Lemma 4.3 we then have

$$\Delta(\langle P_S, V \rangle, S) \geq 5/12 - 1/3 \geq 1/12.$$

Lemma 4.2 now implies that $\Delta(X_i, Y_i) \geq 1/(12v)$ holds for at least one $i = 1, \ldots, v$. ■

The final distributions X and Y are now constructed by independent sampling from X' and Y'. Namely, we put $s = 4 \cdot (12v)^2$ and let

$$X \stackrel{\mathrm{def}}{=} \otimes^s X', \qquad Y \stackrel{\mathrm{def}}{=} \otimes^s Y'.$$

Lemmata 4.4 and 4.5 together with the Direct Product Lemma now imply

$$\Delta(X, Y) \leq (4 \cdot (12v)^2) \cdot \frac{1}{12 \cdot (12v)^2} \leq \frac{1}{3}$$

for $x \in \Pi_{\mathrm{yes}}$ and

$$\Delta(X, Y) \geq 1 - 2\exp(-\frac{4 \cdot (12v)^2 \cdot (1/12v)^2}{2}) > 2/3$$

for $x \in \Pi_{\mathrm{no}}$. This completes the description and the analysis of the reduction.

References

[1] W. Aiello and J. Håstad. Statistical zero-knowledge languages can be recognized in two rounds. *J. Comput. System. Sci.* **42** (1991) no. 3, 327–345.

[2] M. Ben-Or, O. Goldreich, S. Goldwasser, *et al.* Everything provable is provable in zero-knowledge. Proc. CRYPTO '88, LNCS vol. 403, p. 40–51, Springer-Verlag, 1988.

[3] L. Fortnow. The complexity of perfect zero-knowledge. In *Advances in Computing Research* (S. Micali, ed.) volume 5, p. 327–343. JAI Press, 1989.

[4] S. Goldwasser, S. Micali and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.* **18** (1989) no. 1, 186–208.

[5] O. Goldreich, S. Micali and A. Widgerson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *J. ACM* **38** (1991) no. 1, 691–729.

[6] O. Goldreich and S. Vadhan. Comparing entropies in statistical zero-knowledge with applications to the structure of SZK. Proc. 14th IEEE Conference on Computational Complexity, p. 54–73, IEEE Computer Society Press, Los Alamitos, California, 1999.

[7] R. Impagliazzo and M. Yung. Direct minimum knowledge computations. Proc. CRYPTO '87, LNCS vol. 293, p. 40–51, Springer-Verlag, 1987.

[8] T. Okamoto. On relationships between statistical zero-knowledge proofs. Proc. 28th ACM STOC, p. 649–658, ACM, New York, 1996.

[9] A. Shamir. IP = PSPACE. *J. ACM* **39** (1992) no. 4, 869–877.

[10] A. Sahai and S. P. Vadhan. A complete problem for statistical zero-knowledge. Proc. 38th IEEE FOCS, p. 448–457, IEEE Computer Society Press, Los Alamitos, California, 1997.

[11] S. P. Vadhan. A Study of Statistical Zero-Knowledge Proofs. PhD Thesis, Department of Mathematics, Massachusetts Institute of Technology, 1999.