Information Theory versus Complexity Theory: Another Test Case

Ivan Damgård* Oded Goldreich† Avi Wigderson‡

September 1995

Abstract

A few cases are known where the computational analogue of some basic information theoretical results is much harder to prove or even not known to hold. A notable example is Yao's XOR Lemma. Actually, even Direct Sum Conjectures can be viewed as analogues of trivial probabilistic facts. Here we present yet another example of this phenumena. Unlike the examples mentioned above, here we dont know whether even a weak version of the computational analogue holds.

Organization

We start with an Information Theoretic formulation of a protocol problem and next we present a protocol (taken from [3, 2]) which solves it. The open problem we suggest refers to the computational properties of this protocol (or protocol problem).

*Dept. of Computer Science, Aarhus Univesity, Denmark.
†Dept. of Applied Math. and Computer Science, Weizmann Institute of Science, Rehovot, Israel.
‡Institute for Computer Science, Hebrew University, Jerusalem, Israel.

1 The Random Selection Problem

The motivation for the following problem comes from [3] (cf., [2]), but can be ignored. We seek a randomized two-party protocol for selecting strings. The two parties to the protocol are called the challenger and the responder. These names are supposed to reflect the asymmetric requirements (presented below) as well as the usage of the protocol. Loosely speaking, we require that

- *efficiency requirements*
  - the protocol uses constant number of rounds;
  - the challenger strategy (determined by the protocol) is probabilistic polynomial-time and reveals all coins it tosses (i.e., it uses "public coins");
- *statistical properties*[1]
  - if the challenger follows the protocol then, no matter which strategy is used by the responder, the output of the protocol is almost uniformly distributed;
  - if the responder follows the protocol then, no string may appear with probability much greater than its probability under the uniform distribution.

We postpone the formal specification of the statistical properties to the analysis of the protocol presented below.

2 The Random Selection Protocol

Actually, we present two version of the protocol.

Construction 1 (Random Selection Protocol – two versions): *Let n and $m < n$ be integers*[2]*, and $H_{n,m}$ be a family of functions, each mapping the set of n-bit long strings onto*[3] *the set of m-bit long strings.*

C1: *the challenger uniformly selects $h \in H_{n,m}$ and sends it to the responder;*

[1] In the following we use two unrelated statistical requirement. The first requirement refers to the statistical ("variation") distance between two distribution, whereas the second refers to a "domination" condition. See statements of the corresponding propositions in Section 3.

[2] In particular, we will use $m \stackrel{\text{def}}{=} n - 4\log_2(n/\varepsilon)$, where ε is an error-bound parameter.

[3] We stress that each function in $H_{n,m}$ rages over all $\{0,1\}^m$. Thus, the challenger may always respond in step C2 even if the responder deviates from the protocol or Version (2) is used.

R1:
- (Version 1): *the responder uniformly selects* $x \in \{0,1\}^n$*, computes* $\alpha = h(x)$ *and sends* α *to the challenger;*
- (Version 2): *the responder uniformly selects* $\alpha \in \{0,1\}^m$ *and sends it to the challenger;*

C2: *the challenger uniformly selects a preimage of* α *under* h *and outputs it.*

We remark that if Version (1) is used and both parties follow the protocol then the output is uniformly distributed in $\{0,1\}^n$. However, the interesting case is when one of the parties deviates from the protocol. In this case, the protocol can be guaranteed to produce "good" output, provided that "good" families of hash functions are being used as $H_{n,m}$. These functions must have relatively succient representation as well as strong random properties. Furthermore, given a function h, it should be easy to evaluate h on a given image and to generate a random preimage (of a given range element) under h. Using the algorithmic properties of $H_{n,m}$ it follows that the instructions specified in the above protocol can be implemented in probabilistic poly(n/ε)-time, which for $\varepsilon = 1/\text{poly}(n)$ means poly(n)-time.

Construction 2 (Preferred family $H^t_{n,m}$): *Let* n*,* $m < n$ *and* $t = \text{poly}(n)$ *be integers. We associate* $\{0,1\}^n$ *with the finite field* $GF(2^n)$ *and consider the set of* $(t-1)$*-degree polynomials over this field. For each such polynomial* f*, we consider the function* h *so that, for every* $x \in \{0,1\}^n$*,* $h(x)$ *is the* m *most significant bits of* $f(x)$*. The family* $H^t_{n,m}$ *consists of all such functions* h*. The canonical description of a function* $h \in H^t_{n,m}$ *is merely the sequence of* t *smallest coefficients of the corresponding polynomial. Finaly, we modify the functions in* $H^t_{n,m}$ *so that for each* $h \in H^t_{n,m}$ *and every* $x' \in \{0,1\}^m$ *it holds* $h(x'0^{n-m}) \stackrel{\text{def}}{=} x'$*.*

In the sequel, we will use the family $H_{n,m} \stackrel{\text{def}}{=} H^n_{n,m}$. We now list the following, easy to verify, properties of the above family.

P1 There is a poly(n)-time algorithm that, on input a function $h \in H^t_{n,m}$ and a string $x \in \{0,1\}^n$, outputs $h(x)$.

P2 The number of preimages of an image y under $h \in H^t_{n,m}$ is bounded above by $2^{n-m} \cdot t$; furthermore, there exists a poly$(2^{n-m}t)$-time algorithm that, on input y and h, outputs the set $h^{-1}(y) \stackrel{\text{def}}{=} \{x : h(x) = y\}$. (The algorithm works by trying all possible extensions of y to an element of $GF(2^t)$; for each such extension it remains to find the roots of a degree $t-1$ polynomial over the field.)

P3 $H^t_{n,m}$ is a family of almost t-wise independent hashing functions in the following sense: for every t distinct images, $x_1, ..., x_t \in (\{0,1\}^n - \{0,1\}^m 0^{n-m})$, for a uniformly chosen $h \in H^t_{n,m}$, the random variables $h(x_1), ..., h(x_t)$ are indepedently and uniformly distributed in $\{0,1\}^m$.

3 The Statistical Behaviour of the Protocol

In the sequel, we will be discussing a computational analogue of the statement proven in the first subsection. Thus, the reader may ignore the rest of this section.

3.1 The output distribution for honest challeger

We now turn to analyze the output distribution of the above protocol, assuming that the challenger plays according to the protocol. In the analysis we allow the responder to deviate arbitrarily from the protocol and thus as far as this analysis goes the two versions in Construction 1 are equivalent. The analysis is done using the "random" properties of the family $H^t_{n,m}$. Recall that the statistical difference between two random variable X and Y is

$$\frac{1}{2} \sum_{\alpha} |\text{Prob}(X = \alpha) - \text{Prob}(Y = \alpha)|$$

We say that X is ε-*away* from Y if the statistical difference between them is ε.

Proposition 1 *Let n be an integer, $\varepsilon \in [0,1]$ and $m \stackrel{\text{def}}{=} n - 4\log_2(n/\varepsilon)$. Suppose that $H_{n,m}$ is a family of almost n-wise independent hashing functions. Then, no matter which strategy is used by the responder, provided that the challenger follows the protocol, the output of the protocol is at most $(2\varepsilon + 2^{-n})$-away from uniform distribution.*

Proof: Recall that an equivalent definition of the statistical difference between two random variables, X and Y, is

$$\max_S \{|\text{Prob}(X \in S) - \text{Prob}(Y \in S)|\}$$

In our case, one random variable is the output of the protocol whereas the other is uniformly distributed. Thus, it suffices to upper bound the difference between the probability that the output hits an arbitrary set S and the density of S (in $\{0,1\}^n$). Furthermore, it suffices to consider sets S of density greater/equal to one half (i.e., $|S| \geq \frac{1}{2} \cdot 2^n$). Let us denote by $\alpha^* : H_{n,m} \mapsto \{0,1\}^m$ an arbitrary strategy employed by the responder. Then, under the conditions of the proposition, the output of the protocol uniformly distributed in the random set $h^{-1}(\alpha^*(h))$, where h is uniformly selected in $H_{n,m}$. Thus, for a set S, the probability that the output is in S equals

$$\mathrm{Exp}_{h\in H_{n,m}}\left(\frac{|h^{-1}(\alpha^*(h))\cap S|}{|h^{-1}(\alpha^*(h))|}\right) \tag{1}$$

For an arbitrarily fixed set S, we can bound the expression in Eq. (1) by considering the event in which a uniformly chosen $h \in H_{n,m}$ satisfies

$$\frac{|h^{-1}(\alpha)\cap S|}{|h^{-1}(\alpha)|} \notin [(1\pm 2\varepsilon)\rho(S)] \quad \text{for all } \alpha \in \{0,1\}^m. \tag{2}$$

where $\rho(S) \stackrel{\rm def}{=} \frac{|S|}{2^n}$. Whenever this event occurs, Eq. (1) is in the interval $[(1-2\varepsilon)\rho(S),(1+2\varepsilon)\rho]$ and so the statistical difference is at most 2ε. Thus, it remains to upper bound the probability that the above event does not hold. We first note that when estimating the cardinality of the sets $h^{-1}(\alpha)$ and $h^{-1}(\alpha)\cap S$ we may ignore the contribution of the preimages in $\{0,1\}^m 0^{n-m}$, since there is at most one such elements (i.e., $\alpha 0^{n-m}$). Fixing an arbitrary α and using the t-moment method, with $t = n$, we get

$$\begin{aligned}
\mathrm{Prob}_{h\in H_{n,m}}\left(|h^{-1}(\alpha)\cap S| \notin [(1\pm\varepsilon)\rho(S)2^{n-m}]\right) &< \left(\frac{t}{\varepsilon\cdot\rho(S)}\cdot 2^{-(n-m)/2}\right)^n \\
&< \left(\frac{n}{n^2}\right)^n \\
&< 2^{-2n}
\end{aligned}$$

Thus, with overwhelmingly high probability, $|h^{-1}(\alpha)\cap S| \in [(1\pm\varepsilon)\rho(S)\cdot 2^{n-m}]$, for all $\alpha \in \{0,1\}^m$. By a similar argument, with overwhelmingly high probability, $|h^{-1}(\alpha)| \in [(1\pm\varepsilon)\cdot 2^{n-m}]$, for all $\alpha \in \{0,1\}^m$. Thus, with overwhelmingly high probability (i.e., at least $1-2^{-n}$), the event in Eq. (2) holds. ■

Corollary 1 *Suppose that $S \subset \{0,1\}^n$ is a set of density ρ and that the challenger follows the protocol. Then, no matter which strategy is used by the responder, the output of the protocol hits S with probability at most $2\varepsilon + 2^{-n} + \rho$.*

3.2 The output distribution for honest responder

We now show that no matter what strategy is used by the challenger, if the responder follows the protocol then the set of possible outputs of the protocol must constitute a non-negligible fraction of the set of n-bit long strings. This claim holds for both versions of Construction 1. Furthermore, we show that no single string may appear with probability which is much more than 2^{-n} (i.e., its probability weight under the uniform distribution).

Proposition 2 *Suppose that $H_{n,m} = H^t_{n,m}$ is a family of hashing functions satisfying property (P2), for some $t = \text{poly}(n)$. Let C^* be an arbitrary challenger strategy. Then, for every $x \in \{0,1\}^n$, the probability that an execution of* Version (1) *of the protocol with challenger strategy C^* ends with output x is at most $(t \cdot 2^{n-m}) \cdot 2^{-n}$.*

Proof: We consider an arbitrary (probabilistic) strategy for the challenger, denoted C^*. Without loss of generality, we may assume that the first message of this strategy is an element of $H_{n,m}$ (messages violating this convention are treated/interpreted as a fixed function $h_0 \in H_{n,m}$). Similarly, we may assume that the second message of the challenger, given partial history (h, α), is an element of $h^{-1}(\alpha)$ (again, messages violating this convention are interpreted as, say, the lexicographically first element of $h^{-1}(\alpha)$). Finally, it suffices to consider deterministic strategies for the challenger; since, given a probabilistic strategy C^*, we can uniformly select a sequence r respresenting the outcome of the coin tosses of C^* and consider the strategy $\mathbf{c}(\cdot) \stackrel{\text{def}}{=} C^*_r(\cdot) \stackrel{\text{def}}{=} C^*(r, \cdot)$.

We now upper bound the probability that an execution of the protocol with challenger strategy $\mathbf{c}$ ends with output x. We denote by $h \stackrel{\text{def}}{=} c(\lambda)$ the first message of strategy $\mathbf{c}$. Now, the protocol may end with output x only if the responder chose the message $\alpha \stackrel{\text{def}}{=} h(x)$. Thus, the probability that the responder choose α is exactly $|\{x' : h(x') = \alpha\}| \cdot 2^{-n}$. By property (P2), for each $h \in H_{n,m}$ and $\alpha \in \{0,1\}^m$, the cardinality of the set $h^{-1}(\alpha)$ is at most $t \cdot 2^{n-m}$. The proposition follows. ■

Proposition 3 *Let C^* be an arbitrary challenger strategy. Then, for every $x \in \{0,1\}^n$, the probability that an execution of* Version (2) *of the protocol with challenger strategy C^* ends with output x is at most 2^{-m}. Furthermore, for every deterministic challenger strategy* $\mathbf{c}$, *exactly 2^m strings may appear as output, each with probability exactly 2^{-m}.*

Proof: Fix a deterministic strategy $\mathbf{c}$ and a string $x \in \{0,1\}^n$. As in the previous proof, we may assume that $h \stackrel{\text{def}}{=} \mathbf{c}(\lambda) \in H_{n,m}$ and $\mathbf{c}(\alpha) \in h^{-1}(\alpha)$. Denoting $h \stackrel{\text{def}}{=} \mathbf{c}(\lambda)$, Version (2) terminates with output x if and only if the responder chooses the message $\alpha \stackrel{\text{def}}{=} h(x)$ and $x = \mathbf{c}(\alpha)$. Since α is selected uniformly in $\{0,1\}^m$, the proposition follows. ■

The original motivation: simultability of the Protocol

The above protocol has the additional usefuly property of being "simulateable" in the sense that one can efficiently generate random transcripts of the protocol having a given outcome. This property, restricted to the case in which the responder follows the instruction specified by the protocol, is important for the application in [3, 2].

As in the proof of the last two propositions, it suffices to consider an arbitrary deterministic challenger strategy, denoted $\mathbf{c}$. Suppose that $H_{n,m} = H^t_{n,m}$ is a family of hashing functions satisfying property (P1), for some $t = \text{poly}(n)$. Then, on input x and access to a function $\mathbf{c}:\{0,1\}^* \mapsto \{0,1\}^*$, we can easily test if $\mathbf{c}(h(x)) = x$, where $h \stackrel{\text{def}}{=} \mathbf{c}(\lambda)$. In case the above condition holds, the triple $(h, h(x), x)$ is the only transcript of the execution of the protocol, with challenger strategy $\mathbf{c}$, which ends with output x. Otherwise, there is no execution of the protocol, with challenger strategy $\mathbf{c}$, which ends with output x. Thus,

Proposition 4 *Consider executions of the Random Selection protocol in which the challenger strategy, denoted* $\mathbf{c}$*, is an arbitrary function and the responder plays according to the protocol. There exists a polynomial-time oracle machine that, on input* $x \in \{0,1\}^n$ *and* $h \in H_{n,m}$ *and oracle access to a function* $\mathbf{c}$*, either generates the unique transcript of a* $\mathbf{c}$*-execution which outputs* x *or indicates that no such execution exists.*

Proposition 1 motivates us to set ε (the parameter governing the approximation of the output in case of honest challenger) as small as possible. On the other hand, Propositions 2 and 3 motivates us to maintain the difference $n - m$ small and in paricular logarithmic (in n). Recalling that $n - m = 4\log_2(n/\varepsilon)$, this suggests setting $\varepsilon = 1/p(n)$ for some fixed positive polynomial p.

4 Open Problem: the Computational Behaviour of the Protocol

We ask whether the following computational analogue of Corollary 1 holds.

> Let $f : \{0,1\}^* \mapsto \{0,1\}^*$ be a length preserving function which is hard to compute on the average. Namely, suppose that for any probabilistic polynomial-time algorithm A (resp., any non-uniform family of polynomail-size circuits $\{C_n\}$) the probability that $A(x) = f(x)$, when $x \in \{0,1\}^n$ is uniformly chosen, is negligible. Does this mean that there exists no efficient strategy for the responder allowing it to play the protocol so that it can always evaluate f on the outcome of the protocol? And if so, is it infeasible to play the protocol so that f can be evaluate on the outcome with probability, say, at least 1/2?

A naive approach which fails is to consider all $\text{poly}(n)$-size circuits. Indeed, for each such circuit, the set of inputs for which f can be evaluated is of negligible density and by Corollary 1 there is no way to have the outcome of the protocol hit it with substaintial probability. Still, there are too many circuits and in particular for every possible choice of $h \in H_{n,m}$ there is a $\text{poly}(n)$-size circuit which evaluates f correctly on all points in $h^{-1}(1^m)$. Thus, one cannot settle the above question by a mere counting argument along these lines.

One motivation for the above question is that it is related to showing that the transformation presented in [3] is applicable to argument systems and proof of knowledge. However, for this purpose one would need even stronger results. One generalization (which suffices only for 1-round zero-knowledge protocols) is to replace f by a relation R. Namely, suppose that given a random x it is hard to find y so that $(x, y) \in R$. Does it follow that it is infeasible to play the protocol so that on outcome x it is also hard to find such a y?

For our application, and also in general, one may relax the open problem and phrase it with respect to the (Random Selection) protocol problem as presented in Section 1 (rather than with respect to the particular protocol presented in Section 2). We stress that the conditions of Section 1 requite a constant-round public-coin protocol and so the Interactive Hashing protocol of [4] and Coin Tossing into the well [1] are ruled-out.

References

[1] M. Blum, *Coin Flipping by Phone*, IEEE Spring COMPCOM, pp. 133–137, February 1982. See also SIGACT News, Vol. 15, No. 1, 1983.

[2] I. Damgård, O. Goldreich, T. Okamoto, and A. Wigderson: *Honest Verifier vs Dishonest Verifier in Public Coin Zero-Knowledge Proofs*, in the proceedings of Crypto95.

[3] I. Damgård, O. Goldreich, and A. Wigderson: *Hashing Functions can Simplify Zero-Knowledge Protocol Design (too)*, BRICS Technical Rerport RS-94-39, Nov. 1994.

[4] M. Naor, R. Ostrovsky, R. Venkatesan and M. Yung: *Zero-Knowledge Arguments for NP can be Based on General Complexity Assumptions*, Proc. of Crypto 92.