Theory Comput Syst
DOI 10.1007/s00224-015-9661-1

Distinguishing Two Probability Ensembles with One Sample from each Ensemble

Luís Antunes[1] · Harry Buhrman[2] · Armando Matos[3,4] · André Souto[5,6] · Andreia Teixeira[3,7]

Abstract We introduced a new method for distinguishing two probability ensembles called *one from each* method, in which the distinguisher receives as input two samples, one from each ensemble. We compare this new method with *multi-sample from the same* method already exiting in the literature and prove that there are ensembles distinguishable by the new method, but indistinguishable by the *multi-sample from the same* method. To evaluate the power of the proposed method we also show that if non-uniform distinguishers (probabilistic circuits) are used, the *one from each*

A very preliminary version of this paper appeared in Proceedings of the 7nd International Conference on Computability in Europe, Sofia, Bulgaria, June 2011, pages 155-164.

✉ Luís Antunes
lfa@dcc.fc.up.pt

Harry Buhrman
harry.buhrman@cwi.nl

Armando Matos
acm@dcc.fc.up.pt

André Souto
asouto@math.ist.utl.pt

Andreia Teixeira
andreiasofia@ncc.up.pt

[1] CRACS, INESC-TEC, Faculty of Sciences, University of Porto, Rua Campo Alegre,1021/1055, 4169 - 007 Porto, Portugal

[2] Centrum Wiskunde & Informatica, Nederlands and University of Amsterdam, Amsterdam, Netherlands

[3] DCC - Faculdade de Ciências da Universidade do Porto, Rua Campo Alegre, 1021/1055, 4169 - 007 Porto, Portugal

Published online: 14 October 2015

method is not more powerful than the *classical* one, in the sense that does not distinguish more probability ensembles. Moreover we obtain that there are classes of ensembles, such that

– any two members of the class are *easily distinguishable* (a definition introduced in this paper) using one sample from each ensemble;
– there are pairs of ensembles in the same class that are indistinguishable by *multi-sample from the same* method.

Keywords Indistinguishability · Multi-sample distinguishers · Communication complexity · Single-message protocols

1 Introduction

The computational indistinguishability of two probabilistic ensembles is a fundamental concept in Cryptography and in Physics. It is usually assumed that the most general method for efficiently distinguishing two probabilistic ensembles P and Q is an efficient (probabilistic poly-time) algorithm A, which "measures" a property of the ensembles, outputting either 0 or 1. If the value of $|E(A(P)) - E(A(Q))|$ is not negligible,[1] we say that the algorithm distinguishes the two ensembles; in the previous expression, $E(\cdot)$ denotes the expected value of a random variable, while $A(P)$ and $A(Q)$ are random variables corresponding to the output of A when the input is distributed according to P or Q, respectively. In this paper, due to the extensive use in the literature, this method for distinguishing ensembles will be called the *classical* method.

However, in many situations, this is not the most general method for (efficiently) distinguishing two ensembles. For instance, in [4] the authors consider distinguishing algorithms that receive as input two independent samples of the same ensemble (either both from P or both from Q) and prove that there are pairs of ensembles distinguishable by this method (which in this paper we call the *two from the same*

[1] A non negative function f is said to be *negligible if for each polynomial p, there is n_0 such that for all $n \geq n_0$, $f(n) \leq 1/p(n)$.*

[4] Laboratório de Inteligência Artificial e Ciência de Computadores, Rua Campo Alegre, 1021/1055, 4169 - 007 Porto, Portugal

[5] DM - Instituto Superior Técnico da Universidade de Lisboa, Av. Rovisco Pais 1049 - 001, Lisboa, Portugal

[6] SQIG at Instituto de Telecomunicações, Av. Rovisco Pais 1049 - 001, Lisboa, Portugal

[7] CINTESIS - Center for Health Technology and Services Research, Rua Dr. Plácido da Costa, s/n Edifício Nascente, Piso 2, 4200-450 Porto, Portugal

method[2]), which are indistinguishable by the *classical* method. In [4] distinguishers with two or more samples have been studied and it was shown that, for every integer $k \geq 1$, a distinguisher that has as input $k + 1$ samples (of the same ensemble) is more powerful, i.e. distinguishes more pairs of ensembles, than a distinguisher that has as input just k samples.

Although traditionally in Cryptography the *classical* method is the used method for distinguishing it is clear that concerning security, one should study and consider all efficient methods of distinguishing distributions ("worst case analysis"), since some eavesdropper can eventually access the two distributions and gain advantage of that information. For that reason, it is important to study the concept of *two from the same* distinguishability introduced in [4] and to explore new forms to efficiently distinguish probability distributions. In this paper, we introduce a new method that, in some cases is better in the sense that is able to distinguish more distributions than the *two from the same* method.

It should be remarked that there are situations in which the *multi-sample from the same* ensamble method is not more powerful than the *classical* method, see [4, 5]. That happens, for instance, when both ensembles are poly-time computable and the distinguishers are probabilistic poly-time algorithms; it also happens when the distinguishers are (non-uniform) probabilistic poly-time circuits. Cryptographers usually consider distributions P and Q which are efficiently samplable in poly-time and distinguishers that are non-uniform. In these cases, our proposed method is not more powerful. From a security point of view, considering only this way of generating probability distributions is not the best option, in the sense that many distributions are generated by unknown methods (based on natural phenomena) and some of these distributions only make sense for certain individual input lengths. Except in Section 3, we assume that the distinguishers are probabilistic poly-time algorithms.

In this paper, we further generalize the concept of distinguisher, by defining a new form of distinguishing ensembles. The generalization is the following: the distinguishing algorithm has two input samples, one from each ensemble.

In Section 3, we show that, if (non-uniform) probabilistic poly-time circuits are used as distinguishers, this new method, like the *multi-sample* method (Definition 2), is not more powerful than the *classical* one.

In Section 4, (uniform) probabilistic poly-time algorithms are used as distinguishers. We show that, in particular, there are probabilistic ensembles distinguishable by the *one from each* method that are indistinguishable by any algorithm corresponding to the hierarchy defined in [5] (algorithms that receive as input several samples of the same ensemble); see Theorem 3 and Corollary 1.

[2]We opt for the nomenclature *two from the same* and *multi-sample from the same* as a short hand to properly indicate that the algorithm used as distinguisher receives two (or more respectively) samples of the same ensemble.

A class $\mathcal{C}$ of probabilistic ensembles is *easily distinguishable* if there is a constant $a > 1/2$ and a probabilistic poly-time algorithm A such that, for any ensembles P and Q in the class $\mathcal{C}$, we have $E(A(P, P)) \leq 1 - a$ and $E(A(P, Q)) \geq a$. Using a modification of Ambainis' single message protocol of Communication Complexity (see [1, 2, 6, 9, 11]) we show in, Section 5, that *easily distinguishable* classes do in fact exist and moreover that they contain pairs of ensembles that are indistinguishable by any algorithm corresponding to the hierarchy defined in [5].

1.1 Notation and Background

The notation used in this paper is standard. For background on Probability Theory and Complexity, we refer the reader to standard textbooks, such as [3] and [10].

All polynomials $p : \mathbb{N}^+ \to \mathbb{R}^+$ considered in this work are positive and have degree at least 1, so that $\lim_{n\to\infty} p(n) = +\infty$. The function $f : \mathbb{N}^+ \to \mathbb{R}^+$ is *negligible* in n if, for every polynomial $p(n)$, we have $f(n) \leq 1/p(n)$ for sufficiently large n. The logical implication and its negation will be denoted by "$\Rightarrow$" and "$\not\Rightarrow$", respectively. The cardinality of a set and the absolute value of an expression are both denoted by "$|\cdot|$".

The probability of an event x is denoted by $\mathrm{pr}(x)$. The expected value of a random variable X is denoted by $E(X)$. By $x \sim P$ we mean that the event x is drawn according to the probabilities associated with the ensemble P. If x is uniformly drawn from the set or tuple X, we write $x \in_U X$. The notation $\binom{a}{b}$ means the number of combinations of a choosing b.

If A is an algorithm with two inputs, an expression like $A(P, Q)$ means that A receives as input two *independent* samples, one from the ensemble P and the other from the ensemble Q; similarly, the inputs of $A(P, P)$ are two *independent* samples from P. Given an algorithm, we denote by R the ensemble that corresponds to the internal uniform random source; by $r, r' \ldots$ we denote independent samples of R. Whenever no confusion may arise we will drop the algorithm argument R, writing for instance $A(P, Q)$ instead of $A(P, Q, R)$. For simplicity, an expression like $E_{\substack{x\sim P\\ r\sim R}} E(A(x, y, r))$, or equivalently $E_{x\sim P,\, r\sim R} E(A(x, y, r))$, which contains the free variable y, will be denoted by $E(A(P, y, R))$ where it is assumed that the expected value of the random variable $A(P, y, R)$ is taken over the first and third arguments, distributed according to P and R, respectively.

Whenever we talk about "algorithms" we mean "probabilistic poly-time algorithms".

Whenever it is important to express that an ensemble P has the parameter n, we write P_n instead of P. For each $n \in \mathbb{N}$, the *domain* of the ensemble P_n is the set $\{0, 1\}^n$ and its *support* is $\sup(P_n) = \{x : |x| = n \wedge P_n(x) \neq 0\}$. An ensemble P_n is *equiprobable* for a value of n if, for each $x \in_U \sup(P_n)$, the probability $P_n(x)$ is the same; P is *equiprobable* if P_n is equiprobable for every n. For each of the indistinguishability definitions used in this paper, the word "distinguishable" is always used as the logical negation of "indistinguishable".

2 Definitions of Indistinguishability

We present the following definitions of indistinguishability: *classical* (computational indistinguishability), *two from the same* indistinguishability, and *one from each* indistinguishability.

The first two concepts had already been studied in the literature (see for instance [4]) and the *two from the same* indistinguishability [4, 5]. As far as the authors of this paper know, the *one from each* indistinguishability is presented here for the first time.

In each of the following definitions of indistinguishability there are two parts: the algorithm having as input one or more samples of the ensembles and the distinguishing criterion, usually based on a difference between expected values of the algorithms.

In expressions like $E_{x\sim P,\, x'\sim Q}(A(x, x'))$ it is assumed that x and x' are of the same size. Moreover, we will sometimes denote by $E(A(P, Q))$ and $E(A(P, P))$ the expressions $E_{x\sim P,\, x'\sim Q}(A(x, x'))$ and $E_{x\sim P,\, x'\sim P}(A(x, x'))$, respectively, when the size is understood from the context.

Definition 1 (*Classical* (computational) indistinguishability) The ensembles $\{P_n\}$ and $\{Q_n\}$ are *classically* indistinguishable or *indistinguishable with one sample* if, for any probabilistic poly-time algorithm A with output 0 or 1, the function $|E(A(P_n)) - E(A(Q_n))|$ is negligible in n.

Definition 2 (*Two from the same* indistinguishability) The ensembles $\{P_n\}$ and $\{Q_n\}$ are *two from the same* indistinguishable or *indistinguishable with 2 samples from the same* if, for any probabilistic poly-time algorithm A with output 0 or 1, the function $|E(A(P_n, P_n)) - E(A(Q_n, Q_n))|$ is negligible in n.

For any $k \in \mathbb{N}$ the definition of *multi-sample from the same* indistinguishability (or *indistinguishability with k samples from the same*) is similar.

Definition 3 (*One from each* indistinguishability) The ensembles P_n and Q_n are *one from each* indistinguishable if, for any probabilistic poly-time algorithm $A(x, y)$ with output 0 or 1, the function $|E(A(P_n, Q_n)) - E(A(P_n, P_n))| + |E(A(P_n, P_n)) - E(A(Q_n, Q_n))|$ is negligible in n.[3]

In the previous definition, if an algorithm A distinguishes P from Q it means that either it received two samples (one from each distribution) and distinguishes them from any other two samples from P (the first term) or that P and Q are two from the same distinguishable distributions (second term). This definition, can be

[3]Notice that since the distinguisher has access to two samples, one could consider a more complex sum involving all possible combinations of samples from P and Q as arguments of A, but this simple form is enough to capture the desired properties, like strengthening the classical definition of indistinguishability and symmetry.

seen as a generalization to the hypothetical scenario of a protocol using random objects (distribution P), where an adversary could interfere with a stream of those random processes by injecting non-uniform ones (distribution Q) and a method that the party(ies) of the protocol could use to detect that interference.

First we show that this definition is at least as powerful as the *classical* one.

Let Δ be the criterion corresponding to ensembles $\{P_n\}$ and $\{Q_n\}$ and regarding the *classical* definition,

$$\Delta = |E(A(P_n)) - E(A(Q_n))| \tag{1}$$

and Δ' the criterion for the same ensembles corresponding to the *one from each* indistinguishability definition,

$$\Delta' = |E(A(P, Q)) - E(A(P, P))| + |E(A(P, P)) - E(A(Q, Q))|. \tag{2}$$

The expressions Δ and Δ' are called the *ensemble distance* (associated with the corresponding definition) between P_n and Q_n.

One desirable property in the concept of indistinguishability is symmetry. The notion introduced in this paper is also symmetric in the sense that if we have two ensembles P and Q, such that P is *one from each* distinguishable from Q then Q is also *one from each* distinguishable from P although the fact that the distance might be slightly different. We list some facts which are either easy to prove, already known or proved in the rest of the paper. An important result of this paper is item 4.

1. If the ensembles P and Q are *classically* distinguishable they are also *distinguishable with k samples* for every $k \geq 1$, see [5].
2. If the ensembles P and Q are *classically* distinguishable or are *distinguishable with 2 samples from the same* ensemble, they are also *one from each* distinguishable.
3. For any $k \geq 1$ there are ensembles P and Q that are *indistinguishable with k samples from the same* ensemble but *distinguishable with* $k+1$ *samples from the same* ensemble, see [4] and [5].
4. There are ensembles P and Q that are *indistinguishable with k samples from the same* ensemble for every $k \geq 1$, but are *one from each* distinguishable, see Theorem 4.

3 When *One From Each* Method is no Better than the *Classical* One

The model of computation influences the power of the distinguishers and have direct impact in Cryptography (see for example [8]). We prove that, in the non-uniform setting, access to one sample from each ensemble does not increase the power of the distinguisher. For simplicity, in this section, we use only the first term of Δ' (Definition 2), namely $|E(A(P, P)) - E(A(P, Q))|$, the term that is different in the two definitions. The consideration of the other term is straightforward.

Theorem 1 *If A is a probabilistic poly-size circuit, $\{P_n\}$ and $\{Q_n\}$ are ensembles such that P_n and Q_n are defined in $\{0,1\}^n$, and $p(n)$ is a polynomial, then*

there is a probabilistic poly-size circuit B such that, for every sufficiently large n, $|E(A(P_n, P_n)) - E(A(P_n, Q_n))| \leq |E(B(P_n)) - E(B(Q_n))| + 1/p(n)$, where the averages are taken over the inputs and over the random coin-flips used by the circuit.

Proof Consider the difference $\Delta' = |E(A(P_n, P_n)) - E(A(P_n, Q_n))|$. In more detail,

$$\begin{aligned}\Delta' &= |E_{x\in P_n, y\in Q_n}(A(x, y)) - E_{w\in P_n, z\in P_n}(A(w, z))| \\ &= |E_{x\in P_n}\left(E_{y\in Q_n}(A(x, y)) - E_{z\in P_n}(A(x, z))\right)| \\ &= E_{x\in P_n}|E_{y\in Q_n}(B(y)) - E_{z\in P_n}(B(z))| \\ &= E_{x\in P_n}|f(x)|\end{aligned}$$

where $f(x) = E_{y\in Q_n}(B(y)) - E_{z\in P_n}(B(z))$, A and B are circuits and Δ' and $f(x)$ are parametrized for inputs of size n.

In order to prove the statement of the theorem, it is necessary to replace the circuit $A(\cdot, \cdot)$ by a circuit $B(\cdot)$ that distinguishes the ensembles that can be distinguished by A. If A distinguishes P_n from Q_n, Δ' is non negligible, so that there are infinitely many values of n and a polynomial $p(n)$, such that $E_{x\in P_n}|f(x)| \geq \frac{1}{p(n)}$. This is the expectation over the 2^n values of x of size n and hence, for at least one x we must have

$$f(x) = \left|E_{y\in Q}(A(x, y)) - E_{z\in P}(A(x, z))\right| \geq \frac{1}{p(n)}.$$

Otherwise, Δ' would be negligible. Thus, B is a circuit that, for these values of n, "contains" inside the value of x, and outputs $A(x, u)$, where u is the only input of $B(u)$. We may write $B(u) = A(x, u)$. Then,

$$|E(B(P_n)) - E(B(Q_n))| = |E(A(x, P_n)) - E(A(x, Q_n))|$$

is non negligible, so that the ensembles P_n and Q_n are distinguishable by a circuit with a single sample as input. □

If P_n and Q_n are poly-time samplable probability ensembles, one sample from each ensemble also does not increase the power of the distinguisher, when we consider probabilistic poly-time algorithms as distinguishers.

Theorem 2 *If A is a probabilistic poly-time algorithm, $\{P_n\}_n$ and $\{Q_n\}_n$ are samplable ensembles defined in $\{0, 1\}^n$, and $p(n)$ is a polynomial, then there is a probabilistic poly-time algorithm B such that, for every sufficiently large n, $|E(A(P_n, P_n)) - E(A(P_n, Q_n))| \leq |E(B(P_n)) - E(B(Q_n))| + 1/p(n)$, where the averages are taken over the inputs and over the random coin-flips used by the algorithm.*

Proof Suppose that A and B are probabilistic poly-time algorithms with 2 and 1 inputs respectively and that P_n and Q_n are poly-time samplable. We define $B(u)$ as follows. The algorithm receives u (distributed according to P_n or Q_n) as input.

1. Using the poly-time samplable distribution P_n, generate the (independent) samples x_1, x_2, x_3.

2. Run A to obtain the values $\alpha = A(u, x_1)$ and $\beta = A(x_2, x_3)$.
3. Output $\alpha - \beta$.

Note that

- If u is distributed according to P_n, we have $E(B(u)) = E(A(P_n, P_n)) - E(A(P_n, P_n)) = 0$ (the expectation values of α and β are equal).
- If u is distributed according to Q_n, we have

$$\begin{aligned}\Delta' &= \left|E_{u\in Q_n}(E_{x\in P_n}(B(u))) - E_{u\in P_n}(E_{x\in P_n}(B(u)))\right| \\ &= \left|E_{u\in Q_n}(E_{x\in P_n}(B(u)))\right|\end{aligned}$$

  which is assumed to be non negligible. Observe that taking the expectation over $x \in P_n$ corresponds to an average over the internal random source of the algorithm. Not representing this internal source, one would write

$$\Delta' = \left|E_{u\in Q_n}(B(u)) - E_{u\in P_n}(B(u))\right| = \left|E(B(Q_n)) - E(B(P_n))\right|.$$

Obviously, the single input randomized algorithm B distinguishes P_n from Q_n. □

4 Algorithms as Distinguishers

In this section we establish some simple relationships between Definitions 1 (*classical*), 2 (*two of the same*), and 3 (*one from each*). Notice that if the ensembles $\{P_n\}$ and $\{Q_n\}$ are distinguishable according to Definition 1 (*classical*), they are also distinguishable according to Definition 2 (*two of the same*), and also according to Definition 3 (*one from each*). If the ensembles P and Q are distinguishable according to Definition 2 (*two of the same*), they are also distinguishable according to Definition 3 (*one from each*).

One can summarize these relationships using the following implications between the distinguishability definitions:

$$\textit{classical} \Rightarrow \textit{two from the same} \Rightarrow \textit{one from each}.$$

It is known that *two from the same* $\not\Rightarrow$ *classical* (see [4] for more details), and the fact that *one from each* $\not\Rightarrow$ *two from the same* follows from Theorems 3 and 4; in fact, as we shall see (Corollary 1), we have that for every integer k, *one from each* $\not\Rightarrow$ k *samples from the same*.

Theorem 3 *If $\{P_n\}$ and $\{Q_n\}$ are (different) uniformly distributed ensembles with support size bounded by a polynomial, they are distinguishable by the one from each method.*

Proof For simplicity, assume that the support sizes of P_n and Q_n are equal for each n; denote them by $s(n)$. Let a be the number of elements common to the supports of P_n and Q_n. Consider the algorithm

$$A(x, y) : \text{if } x = y \text{ return 1, else return 0.}$$

We have $E(A(P_n, P_n)) = E(A(Q_n, Q_n)) = 1/s(n)$ (here the value of a is irrelevant). Let us compute $E(A(P_n, Q_n))$. Consider the event (x, y) with x and y distributed according to P_n and Q_n respectively. The probability that $x = y$ is $a/s^2(n)$;
then $|E(A(P_n, Q_n))| = \text{pr}(x = y) = a/s^2(n)$; as $a \leq s(n) - 1$, we have

$$|E(A(P_n, Q_n)) - E(A(P_n, P_n))| = \left| \frac{a}{s^2(n)} - \frac{1}{s(n)} \right| = \frac{s(n) - a}{s^2(n)} \geq \frac{1}{s^2(n)}$$

so that, as $s(n)$ is bounded by a polynomial, algorithm A distinguishes the ensembles P_n and Q_n by the *one from each* method. □

We now show that there are probability ensembles P_n and Q_n that can not be distinguished by any probabilistic poly-time algorithm that receives as input two samples of the same ensemble.

Theorem 4 *For every integer $k \geq 1$ and for every non-decreasing positive function $s : \mathbb{N}^+ \to \mathbb{R}^+$ which is not bounded by a constant but bounded by a polynomial, there are uniform (in the support) probability ensembles $\{P_n\}$ and $\{Q_n\}$ with support size $s(n)$, that are indistinguishable with k samples from the same ensemble.*

Proof For sake of simplicity we present the proof for the case of $k = 2$.

Given an (probabilistic poly-time) algorithm $A(P_n, P_n)$, we consider its average behavior over all the pairs (x_i, x_j), $1 \leq i, j \leq n$ with $x_i, x_j \in \text{sup}(P_n)$. Let us consider the function g_A

$$P_n \xrightarrow{g_A} \quad E(A(P_n, P_n)) \in [0, 1]$$

where the arguments of the algorithm A are two independent samples distributed according to P_n. Notice that there are $\binom{2^n}{s(n)}$ probability distributions over $\{0, 1\}^n$ with support size $s(n)$.

Let $A_1, A_2, \ldots, A_{t(n)}$ be the algorithms considered up to input length n. Once $s(n)$ is not bounded by a constant, we can choose $t(n) = s(n)$. Fix the number of subintervals of $[0, 1]$ considered in each stage, that is for each of the first $s(n)$ algorithms, as

$$f(n) = \left(\frac{1}{2} \times \binom{2^n}{s(n)} \right)^{1/s(n)}.$$

For sufficiently large n we have (recall that $s(n)$ is bounded by a polynomial)

$$\binom{2^n}{s(n)} \geq \left(\frac{2^n}{s(n)} \right)^{s(n)}.$$

Then

$$f(n) = \left(\frac{1}{2} \times \binom{2^n}{s(n)} \right)^{1/s(n)} \geq \left(\frac{1}{2} \times \left(\frac{2^n}{s(n)} \right)^{s(n)} \right)^{1/s(n)} = \frac{2^{n-1/s(n)}}{s(n)} \geq \frac{2^{n-1}}{s(n)}.$$

Thus, as $s(n)$ is bounded by a polynomial, $1/f(n)$ is negligible.

It remains to show that $\binom{2^n}{s(n)} / f^{s(n)}(n) \geq 2$, so that, after $s(n)$ stages, one for each algorithm, we have at least two distinct ensembles, P and Q; in fact we have

$$\binom{2^n}{s(n)} \Big/ f^{s(n)}(n) = \binom{2^n}{s(n)} \Big/ \left(\frac{1}{2} \times \binom{2^n}{s(n)}\right) = 2.$$

In summary, considering an enumeration of the poly-time (halting) Turing machines A_1, A_2,..., where there are $s(n)$ machines considered up to n, we have the following result. For each $m \in \mathbb{N}$ there are ensembles P and Q, defined above, that are indistinguishable with two samples by the algorithms A_1,..., A_m. In fact: (i) every algorithm, say with index m, is introduced at some n_0, (ii) by construction, A_m can not distinguish P from Q, and (iii) the same is true for all $n \geq n_0$. This implies that P and Q are indistinguishable by any probabilistic poly-time algorithm which receives two samples from the same ensemble as input.

The general case. i.e., with $k \geq 2$ samples of the same ensemble is straightforward from the previous argument considering the function h_A

$$P_n \xrightarrow{h_A} \quad E(A(\underbrace{P_n, \cdots, P_n}_{k})) \in [0, 1].$$

□

The proof is insensitive to the number of input samples and also to the class of time and space resources allowed in the distinguishing algorithms, and somewhat insensitive to the (positive) probabilities in the support. Also, the condition that the support size must have a polynomial upper bound, may be relaxed. Some of these generalizations are included in the following result.

Corollary 1 *For every $k \geq 1$ and for every positive function $s : \mathbb{N}^+ \to \mathbb{R}^+$ which is bounded by a polynomial but not by a constant, there are uniform (in the support) probability ensembles $P = \{P_n\}$ and $Q = \{Q_n\}$ with support size $s(n)$, that are indistinguishable with k samples of the same ensemble but distinguishable by the one from each method.*

This result can be further generalized to non-uniform probabilities in the support: given any polynomial p, the result remains valid for probabilities with lower bound $1/p$ (in the support).

5 Easily Distinguishable Classes of Ensembles

We now envisage the existence of classes containing *one from each* distinguishable ensembles that are *indistinguishable with k samples of the same*. To this end we introduce the classes of *easily distinguishable* ensembles.

Definition 4 A class $\mathcal{C}$ of probabilistic ensembles is *easily distinguishable* if there is a probabilistic poly-time algorithm A and a constant $a > 1/2$ such that, for

any ensembles $\{P_n\}$ and $\{Q_n\}$ in the class, we have $E(A(P_n, P_n)) \leq 1 - a$ and $E(A(P_n, Q_n)) \geq a$. In particular, notice that $|E(A(P_n, P_n)) - E(A(P_n, Q_n))|$ is always greater than a constant.

Clearly, if two ensembles are *easily distinguishable*, then they are also *one from each* distinguishable.

In order to prove the existence of *easily distinguishable* classes which are *indistinguishable with k samples from the same*, we will borrow ideas from a variant of Communication Complexity protocols, the SMP (simultaneous message passing) model (see [2, 9]) and in particular the protocol for string equality described in [1], which will be modified in order to define *easily distinguishable* classes of ensembles. The messages sent by Alice or Bob will correspond to events of the ensemble. For sake of completeness of the paper and a better understanding of the argument, we included details in Appendix A of the aforementioned protocol.

We modify the protocol in [1], so that the messages sent by Alice or Bob can be seen as events corresponding to certain probabilistic ensembles. Let p be any real number in the open interval $(1/2, 3/5)$, for instance $p = 6/11$. The choice of this interval and the suggested p are explained in the Appendix A.

1. From $x \in \{0, 1\}^n$, Alice builds the $6m \times 6m$ matrix M where m is the smallest integer satisfying $(6m)^2 \geq 3n$, by using a $[(6m)^2, n, (6m)^2/6]$-code. The codeword for x is laid out in a $(6m) \times (6m)$ square forming the matrix M. See [1] for more details. From $y \in \{0, 1\}^n$, Bob builds, in a similar manner, the $6m \times 6m$ matrix N.
2. Alice selects a random row r and a column c of M and sends the "event" $\langle i, j, r, c\rangle$ to the Referee, where i is the index of row r and j is the index of column c, and r contains all the entries of the i^{th}-row and c all the entries of the j^{th} column.
3. Bob selects a random row r' and a column c' of N and sends the "event" $\langle i', j', r', c'\rangle$ to the Referee, where i' is the index of row r' and j' is the index of column c'.
4. Referee:
   (a) if $r_{j'} \neq c'_i$ or $c_{i'} \neq r'_j$, output NO
   (b) if $r_{j'} = c'_i$ and $c_{i'} = r'_j$, output NO with probability p and YES with probability $1 - p$.

Notice that we are considering an adaptation of Ambainis' protocol to easily distinguish two distributions defined by x and y and defined next.

5.1 The Ensemble Corresponding to a String x

Given any string x with length n we define the probability distribution $P_n(x)$ consisting of the sequences $\langle i, j, r, c\rangle$ that are the possible messages sent by Alice in the previous protocol.

Similarly to the protocol described in [1], it can be shown that, independently of the strings x and y, the answer of the protocol is correct with probability at least p.

5.2 Throwing the Dice

Given a probability distribution $P(x)$, an event corresponds to a selection (i, j) by Alice. Bob's situation is similar, but he "uses" the probability distribution $P(y)$ represented by matrix N.

Notice that the Referee, seen as a distinguisher, receives *one sample from each ensemble*, namely $\langle i, j, r, c\rangle$, an event from $P_n(x)$ and $\langle i', j', r', c'\rangle$, an event from $P_n(y)$.

Moreover, the Algorithm C described above can be used as a distinguisher since it is a probabilistic algorithm that *easily distinguish* the distributions presented if $x \neq y$.

Now to finish the proof we need to show that, among these *easily distinguishable* ensembles there are some that are not *multi-sample from the same* distinguishable.

5.3 Applying the Pigeonhole Principle

Our goal is to use the pigeonhole principle, as in the proof of Theorem 4, in order to prove the existence of x and y (both with length n) such that $P(x)$ and $P(y)$ are indistinguishable by any *multi-sample* (from the same ensemble) distinguisher.

The number of bits in each message is $\Theta(\sqrt{n})$, so that it has an upper bound $c\sqrt{n}$ for some constant c. In fact, we can take as c any number greater than $2\sqrt{3}$, because in this variant of the protocol, the communication length is doubled (relatively to the protocol in [1] presented in the Appendix). From the fact that the support of the considered distributions has size $(6m)^2$, its description is given by $2\log(6m) = 2\log(\sqrt{3n})$ bits, which is unbounded in n. Recall that from Ambainis' protocol m and n are related by $n = (6m)^2$. In terms of Theorem 4, we can thus make the identification

$$\begin{pmatrix} 2^n \\ s(n) \end{pmatrix} \leftrightarrow \begin{pmatrix} 2^{c\sqrt{n}} \\ 2\log(\sqrt{3n}) \end{pmatrix} = \begin{pmatrix} 2^p \\ s(p) \end{pmatrix}$$

where $p = c\sqrt{n}$ is a new variable, and $s(p)$ is unbounded in p. A negligible interval in $[0, 1]$ in terms of p, like 2^{-p} is also negligible in terms of n, $2^{-p} = 2^{-c\sqrt{n}}$ and the function $s(n)$ is certainly unbounded in p and n. In summary, Theorem 2 applies to this class of ensembles and we get the following result.

Theorem 5 *There are infinitely many classes $\mathcal{C}_1$, $\mathcal{C}_2$,... of easily distinguishable ensembles such that, for sufficiently large n and for every k, the class $\mathcal{C}_n$ contains pairs of probability ensembles that are indistinguishable with k samples from the same ensemble.*

As a consequence, there are many non-trivial ensembles (in fact 2^n ensembles with elements of the form $\langle i, j, r, c\rangle$ and size $12m + 2\log(6m)$) that are *easily distinguishable* using the adaptation of Ambainis' protocol but are indistinguishable by *multi-sample*.

Acknowledgments The authors thank to the anonymous reviewers for helpful comments. For the financial support, Andre Souto thanks EU FEDER and FCT project PEst-OE/EEI/LA0008/2011 and the grants of SQIG-Instituto de Telecomunicações and FCT grant SFRH/ BPD/ 76231/ 2011.

Andreia Teixeira thanks SFRH/ BPD/ 86383/ 2012. Harry Buhrman is supported by the grant NWO Zwaartekracht proposal Networks and the EU grant SIQS. Luis Antunes thanks ERDF European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT Fundao para a Cincia e a Tecnologia (Portuguese Foundation for Science and Technology) within project UID/EEA/50014/2013 and also Smartgrids - NORTE-07-0124-FEDER-000056.

Appendix A: 3-computer model

In this part of the paper we provide insight and details for the understanding of Section 5.

First, consider the following "equality" problem: A has a string x, B has a string y and we want to know if $x = y$. If deterministic algorithms are considered, it is known that for each algorithm there are "bad" cases where the whole string needs to be sent from A to B. In the worst case, the number of bits used by any algorithm for string equality is $\Omega(n)$. If probabilistic algorithms with an arbitrary small probability of error are allowed, the situation changes. It becomes possible to compute whether strings are equal just with one message of length $O(\log n)$ from one party to another (see [11]). This result is optimal in the sense that $c \log n$ is also a lower bound, for some c.

In [11], Yao introduced another model called "3-computer model". In this model, there are three parties A, B and C, where one string is given to the first party A and another string is given to the second party B. These two parties can send messages to C. They can not exchange information between themselves and C can not send any messages to the other two. A has variables $x_1, x_2, \cdots, x_n$ and B has variables $y_1, y_2, \cdots, y_n$. A analyzes its variables, sends a message to C, B analyzes its variables and sends a message to C, too. Then C compares the two messages received from A and B and announces the result of the computation. In this new model, it appears to be more difficult to compute if two strings are equal. In [1], the author presents a protocol for "equality function", where $O(\sqrt{n})$ bits are sent.

Definition 5 (Hamming distance) If x, y are strings of Σ^n, $x = (x_1, x_2, \cdots, x_n)$ and $y = (y_1, y_2, \cdots, y_n)$, then the *Hamming distance* between x and y is the number of i such that $x_i \neq y_i$. It is denoted by $d(x, y)$.

Definition 6 (Code) $M \subset \Sigma^n$ is called a $[n, k, \varepsilon]$- *code* if it contains 2^k elements and $d(x, y) \geq \varepsilon$ for every two distinct $x, y \in M$.

Lemma 1 (Theorem 17.30 in [7]) *If* $0 < \delta < \frac{1}{2}$, *then for each n there is an* $[n, k, \varepsilon]$-*code such that* $\varepsilon/n \geq \delta$ *and* $k/n \geq 1 - H_2(\varepsilon/n)$, *where* $H_2(x) = -x \log(x) - (1 - x) \log(1 - x)$.

In [1], the author uses a particular case of this lemma.

Lemma 2 *For each m there is a* $[3m, m, m/2]$-*code.*

Proof In Lemma 1 replace n by $3m$ and δ by $\frac{1}{6}$. □

Theorem 6 (Theorem 1 in [1]) *It is possible to compute the equality function in the 3-processor model so that both A and B transmit* $\sqrt{3n} + o(\sqrt{n})$ *bits to C and the probability of the correct answer is at least* $6/11$.

Proof Let n be a fixed size of the string and m be the smallest integer satisfying the inequality $(6m)^2 \geq 3n$. There is a $[(6m)^2, (6m)^2/3, (6m)^2/6]$-code, from Lemma 2 above. In order to obtain a $[(6m)^2, n, (6m)^2/6]$-code, 2^n elements of the $[(6m)^2, (6m)^2/3, (6m)^2/6]$-code are chosen. Then, we establish a one-to-one correspondence between the elements of this new code and words $x \in \Sigma^n$. Next, we detail the description of the algorithm.

For the party A: Find the codeword $s = (s_1, \cdots, s_{(6m)^2})$ corresponding to input data $x = (x_1, \cdots, x_n)$ and consider a $6m \times 6m$ table with the numbers $s_1, \cdots, s_{(6m)^2}$ in the squares of the table. Choose a random row i, where i is uniformly distributed over $\{1, \cdots, 6m\}$, and communicate $(i, a_1, a_2, \cdots, a_{6m})$ to C, where $(a_1, a_2, \cdots, a_{6m})$ is the content of row i.

For the party B: Find the codeword $s = (s_1, \cdots, s_{(6m)^2})$ corresponding to the input data $y = (y_1, \cdots, y_n)$ and consider a table as in the case for A. Choose a random column j, where j is uniformly distributed over $\{1, \cdots, 6m\}$, and communicate $(j, b_1, b_2, \cdots, b_{6m})$ to C , where $(b_1, b_2, \cdots, b_{6m})$ is the content of column j.
For the party C: C compares a_j and b_i. If $a_j \neq b_i$, C communicates that $g = 0$ $(x \neq y)$. If $a_j = b_i$, C communicates that $g = 1$ $(x = y)$ with probability $\frac{6}{11}$ and that $g = 0$ with probability $\frac{5}{11}$.

The number of bits communicated from A (or B) to C is $6m + \lceil \log(6m) \rceil = \sqrt{3n} + o(\sqrt{3n})$.

In order to prove that the algorithm really computes g with the probability of a correct answer being at least $\frac{6}{11}$, notice that a_j corresponds to the entry (i, j) of the table constructed by A and that b_i corresponds to the entry (i, j) of the table constructed by B. If $g = 1$ $(x = y)$, then the tables of A and B are equal. Hence $a_j = b_i$. Thus, with probability $\frac{6}{11}$, C gives the answer $g = 1$. If $g = 0$ $(x \neq y)$, then A and B construct two different tables. As the squares of these tables have the codewords from a $[(6m)^2, n, (6m)^2/6]$-code, these tables are different in at least $\frac{(6m)^2}{6}$ squares (one-sixth of all squares).

A and B choose the value for the pair (i, j) with equal probability. Thus, each square becomes the square contents of which C receives from both A and B with equal probability. With probability $p_0 \geq \frac{1}{6}$ the square in which the numbers in two tables are different is chosen, *i.e.*, with probability p_0, C receives two different values and with probability $1 - p_0$ two equal values. So, C announces the correct answer $g = 0$ with probability $p_0 + \frac{5}{11}(1 - p_0) = \frac{5}{11} + \frac{6}{11} \times \frac{1}{6} = \frac{6}{11}$. □

Notice that the protocol presented in the proof above outputs 1 correctly with probability p and outputs 0 correctly with probability $1 - \frac{5}{6}p$. In particular, to have this two probabilities greater than $1/2$, the value of p should belong to $(1/2, 3/5)$. Furthermore, $p = 6/11$ turns the aforementioned probabilities equals.

Then, the probability of error can be made arbitrarily small, repeating this algorithm many times and considering the majority of the outcomes as the final result by C. The number of communicated bits is $O(\sqrt{n})$.

References

1. Ambainis, A.: Communication complexity in a 3-computer model. Algorithmica **16**(3), 298–301 (1996)
2. Babai, L., Kimmel, P.: Randomized simultaneous messages. In: Proceedings 12Th IEEE Symposium on Computational Complexity, pp. 239–246. IEEE (1996)
3. Feller, W.: An Introduction to Probability Theory and Its Applications, vol. 1. Wiley (1968). http://www.amazon.ca/exec/obidos/redirect?tag=citeulike04-20&path=ASIN/0471257087
4. Goldreich, O., Meyer, B.: Computational indistinguishability: Algorithms vs. circuits. Theor. Comput. Sci. **191**, 215–218 (1998). doi:10.1016/S0304-3975(97)00162-X. http://portal.acm.org/citation.cfm?id=278633.278647
5. Goldreich, O., Sudan, M.: Computational indistinguishability: a sample hierarchy. J. Comput. System Sci. **59**(2), 253–269 (1999). doi:10.1006/jcss.1999.1652
6. Kushilevitz, E., Nisan, N.: Communication Complexity. Cambridge University Press (1997)
7. Macwilliams, F., Sloane N.: The Theory of Error-Correcting Codes (1978)
8. Maurer, U., Tessaro, S.: A hardcore lemma for computational indistinguishability: Security amplification for arbitrarily weak prgs with optimal stretch. In: Micciancio, D. (ed.) Theory of Cryptography, Lecture Notes in Computer Science, vol. 5978, pp. 237–254. Springer, Berlin Heidelberg (2010). doi:10.1007/978-3-642-11799-2-15
9. Newman, I., Szegedy, M.: Public vs. private coin flips in one round communication games (extended abstract). In: Proceedings 28Th ACM Symposium on the Theory of Computing, pp. 561–570. ACM Press (1996)
10. Sipser, M. Introduction to the Theory of Computation, 1st edn. International Thomson Publishing (1996)
11. Yao, A.: Some complexity questions related to distributive computing(preliminary report). In: Proceedings of the 11th annual ACM symposium on Theory of computing, STOC '79, pp. 209–213. ACM, New York (1979). doi:10.1145/800135.804414