Succinct Proofs for NP and Spooky Interactions

Cynthia Dwork * Michael Langberg† Moni Naor‡ Kobbi Nissim§ Omer Reingold¶

December 21, 2004

Abstract

This paper investigates the possibility that any NP statement can be proven by an *argument system* in two rounds of communication (i.e. one message from the verifier to the prover and one message back) while sending only relatively few bits, without relying on the existence of a random oracle. More specifically, we focus on a natural approach (suggested in [1]) for designing such an argument system by a combination of two tools: (i) The PCP (probabilistically checkable proofs) Theorem that states that for every language $L \in$ NP there exist polynomial size witnesses that may be verified, with constant error probability, by probing only a constant number of locations of the proof, and (ii) Computational PIR (private information retrieval) schemes. The idea is simple: to verify an NP statement, the verifier simulates a PCP verifier where every query is performed via a (computational) PIR. Although this protocol is very natural, attempts to prove its soundness have failed. We exhibit inherent difficulties in such attempts (even when applied to extensions of this protocol). Our results give some indications of the direction one must take in order to construct efficient proof systems in this way.

*Microsoft Research - Silicon Valley, Mountain View, CA USA. E-mail: dwork@microsoft.com. Some of this work done while at Compaq Systems Research Center.

†Dept. of Computer Science and Applied Math, Weizmann Institute of Science, Rehovot, Israel. E-mail: mikel@wisdom.weizmann.ac.il

‡Dept. of Computer Science and Applied Math, Weizmann Institute of Science, Rehovot, Israel. E-mail: naor@wisdom.weizmann.ac.il. Research Supported by an EU Grant. Some of this work done while visiting Stanford Univ. and IBM Almaden.

§DIMACS, Rutgers University, Piscataway, NJ 08854, USA. E-mail: kobbi@dimacs.rutgers.edu. Some of this work done while at the Weizmann Institute of Science.

¶AT&T Labs - Research. Building 103, 180 Park Avenue Florham Park, NJ, 07932, USA. E-mail: omer@research.att.com

1 Introduction

NP is the class of languages with polynomial length proofs. I.e. for every instance x in the language there is a proof (also called a witness), of size polynomial in the length of x. Viewed as a proof system, there is a prover that can send a message whose length is polynomial in $|x|$ and a polynomial time verifier that can check the proof. If $x \in L$ then the verifier accepts the proof (completeness) and for any $x \notin L$, there is no proof that makes the verifier accept (soundness). The PCP Theorem [2, 3] states that for every language $L \in$ NP there exist witnesses of polynomial length that may be verified, with constant error probability, by probing only a constant number of locations of the proof. However if we want to obtain a proof system where the total communication from the prover to the verifier and back is low (polylogarithmic or sublinear), then we are faced with the impossibility results of Goldreich and Håstad [13] and Goldreich, Vadhan and Wigderson [15] who showed that only languages which are relatively easy may have such proof systems. Therefore we must settle for *computational* soundness, or an *argument system* where the prover is assumed to be computationally bounded and for no $x \notin L$ can it succeed in making the verifier accept with non-negligible probability. Such argument systems for all languages in NP, based on the PCP Theorem and computational assumptions, were constructed by Kilian [17, 18] and Micali [21]. The constructions of Kilian are based on a standard computational primitive, namely a collision-intractable hash function. The resulting protocol is a communication-efficient 3-round (prover-verifier-prover) protocol. Micali [21] constructs a one-round argument system (called CS – computationally sound – proofs). The construction assumes the existence of a random oracle.

The problem studied in this paper is whether there exist two-round (or single-round) communication efficient argument systems for NP in the *standard* model. This is central to understanding the tradeoff between the number of rounds and the number of communication bits in the context of non-trivial argument systems. Moreover, two-round communication-efficient argument systems, if they exist, would have many appealing applications. Consider for example the setting of remote procedure calls: Alice asks Bob to compute the outcome of some piece of code on a specific input. This can be done in two rounds (Alice's request and Bob's reply – the outcome of the computation). Two-round communication-efficient argument systems for NP may enable Bob to also prove the correctness of his answer with a relatively low overhead in communication and without changing the communication pattern of the original protocol. The significance of the two-round requirement is that Bob can be stateless and does not have to remember what he did in previous rounds. The complexity of generating such proofs using current techniques may be too high to be used as described; however the hope is that showing that such a process is possible in principle will inspire more practical methods for specific instances.

Combining PCPs and computational PIR We focus on a specific approach, previously suggested by Aiello et al. [1], for designing efficient argument systems in two rounds. This approach is based on the composition of PCP systems with computational PIR (private information retrieval) schemes. A computational PIR scheme enables a user to read a specific location of a database without revealing any information about this location.

The starting point of [1] (as in [17, 18, 21]) is the PCP Theorem. Since the PCP verifier only looks at a few locations, we may consider the following (seemingly ridiculous) two-round protocol:

Verifier: Simulate a PCP verifier and ask the prover for the probed locations.
Prover: Reply with the corresponding bits of the PCP proof.
Verifier: Accept iff the PCP verifier accepts.

This protocol is obviously not sound: allowing the prover to choose its PCP proof after seeing the verifier's queries makes it easy for him to cheat. But what if he does not get to see the actual locations of interest but rather some "encrypted" version of them? In such a case, the prover *seems* to have no way of adjusting his proof to the verifier's queries. This is exactly the intuition behind the basic two-round arguments we study: The verifier simulates the PCP verifier but instead of sending his queries in the clear, *he performs them via parallel executions* (independent on his part) *of a PIR scheme*. Using the PIR scheme of [7], the protocol will consist of two rounds and have small communication complexity.

Our results Despite the intuition described above, our first result shows that the basic two-round protocol is not always sound: There exists a concrete PCP which, if used in the basic two-round construction, can be fooled, i.e. every statement can be proved correct. The flaw of the basic protocol is that nothing forces the prover to answer different queries consistently, i.e. according to the same "database". (The authors of [1] have withdrawn their work as a result of this observation and our results that follow.)

However, this problem seems easy to fix following a standard way of transforming PCP to sound MIP (multiprover interactive proofs) [11, 6]. The verifier may check the consistency of the provers answers: In addition to the PCP queries, the verifier can hide (in random locations) repeated queries to the same location with the sole purpose of verifying that the prover answers consistently.

Our main results show fundamental difficulties in proving the soundness of the revised protocol as well (and even more general versions of this protocol). The essence of the problem is that seemingly independent executions of PIR schemes may have what we call **"spooky interactions"** (see precise Definition 5.5). Even though the queries are formed independently and the prover does not have a real understanding of them, there may be hidden interactions (i.e. unexpected correlations between queries and outputs). This problem holds irrespectively of the specific PCP system in use.

The possibility of spooky interactions seems to be a fundamental issue. We do not know whether there exist PIR schemes for which parallel executions yield spooky interactions or whether the properties of PIR schemes (namely *privacy* - that the database learns nothing about the locations being probed; and *correctness* - that database information can be read by the user assuming both parties are honest) rule out such interactions. This is an intriguing open problem. In our work, we identify for any efficient PCP system $\mathcal{P}$, a possible form of spooky interaction which contradicts the soundness of the basic protocol with $\mathcal{P}$. We conclude, that such interactions must be ruled out in any attempt to prove the soundness of the basic protocol.

In addition we study the existence of spooky interactions in settings other than the basic two round protocol in which parallel executions of a computational PIR are used. In Appendix B, we introduce a variant of PIR that performs several queries at once – Multi-PIR. We show that Multi-PIR does allow spooky interactions. Therefore, the basic protocol with Multi-PIR is not necessarily sound. Furthermore, in Appendix C we show an information theoretic PIR where spooky interactions occur.

The results of this paper seem to identify the directions one should follow in order to construct two-round arguments using the approach of [1]. It seems apparent that any success in this line of research requires better understanding of PIR schemes. We discuss the resulting questions in Section 7.

Organization: In Section 2 we review some standard definitions used throughout the paper and define the parallel PIR protocol. In Section 3 we review the basic two round protocol. Sections 4, and 5 explain the difficulties in proving the soundness of such protocols: Section 4 shows a counterexample based on the lack of consistency checks. In Section 5 we discuss the possibility of fixing the basic two round protocol using consistency checks. For that end we look in more depth into the parallel PIR protocol. The results of Section 5 are generalized in Section 6. In Appendix D we discuss two issues: how to reduce the soundness error via parallel repetition and how to achieve witness indistinguishability. Finally, we discuss our results and state the main open problems.

2 Tools and definitions

In this section we review the tools used in the construction of the proposed argument system for NP statements, namely, the PCP Theorem and computational PIR schemes. We then define communication efficient argument systems.

2.1 Probabilistically checkable proofs

We now define a probabilistically checkable proof (PCP) system in terms of its verifier $\mathcal{V}$. We are assuming that $\mathcal{V}$ has access to an oracle tape π where on query i the response is $\pi(i)$, i.e. the bit at location i of π.

Definition 2.1 (Probabilistically checkable proofs (PCP)) *A language L is in $PCP_{\delta,\varepsilon}(r,q)$ if there exists a probabilistic polynomial time verifier $\mathcal{V}$ with access to an oracle tape π which uses at most r random bits and queries at most q oracle bits such that:*

Completeness: *For any $x \in L$, there exists a proof tape π s.t.* $\Pr[\mathcal{V}(x,\pi)=1] \geq \delta$.

Soundness: *For any $x \notin L$ and for all tapes π^*,* $\Pr[\mathcal{V}(x,\pi^*)=1] \leq \varepsilon$.

The probabilities are over the random choices of the verifier $\mathcal{V}$.

After a long line of work including [3, 2], a new characterization of NP (known as the PCP Theorem) was achieved, namely it was shown that NP $= PCP_{1,1/2}(O(\lg|x|), O(1))$. This implies that $|\pi| = \text{poly}(|x|)$. In the constructions that are analyzed in this work we use PCP systems with low query complexity (up to polylogarithmic in the instance length $|x|$) so that a simulation of such systems in our context results in arguments of low communication complexity.

2.2 Private information retrieval

The parties of a private information retrieval scheme are a chooser and a sender. The sender holds a database $\mathcal{DB}$ of length n and the chooser holds an index $i \in [n]$. Intuitively, the protocol enables the chooser to read location i of $\mathcal{DB}$ without the sender learning anything about i. It is convenient for us to define PIR privacy against a non-uniform adversary[1]. The PIR scheme is defined by three algorithms: Q and R are executed by the chooser and D by the sender. Algorithm Q is the query generator that maps the index i into a query q using the chooser's secret random string s (s is to remain secret if the chooser does not wish to reveal i). Algorithm D is the one executed by the (honest) prover in response to query q and as a function of the database $\mathcal{DB}$. Finally, algorithm R allows the chooser to reconstruct the value at location i as a function of the response it received from the sender and the secret random string s.

Definition 2.2 (Private information retrieval (PIR)) *Let D, Q, R be probabilistic polynomial time algorithms. Let k be a security parameter and n the length of the database. (D, Q, R) is a (computational) PIR scheme if*

Correctness: *For any n-bit string $\mathcal{DB}$ and any location $i \in [n]$,*

$$\Pr\left[(q,s) \leftarrow Q(n,i,1^k); a \leftarrow D(\mathcal{DB},q,1^k) : R(n,i,(q,s),a,1^k) = \mathcal{DB}[i]\right] \geq 1 - \textit{neg}(k)$$

[1]This implies a cleaner statement of Lemma 5.3. Considering a stronger definition of PIR only makes our results more meaningful.

Privacy: *For any family of polynomial-size (probabilistic) circuits* $\{A_k\}_{k\in\mathbb{N}}$ *and any* $i, j \in [n]$,

$$\left|\Pr\left[(q,s) \leftarrow Q(n,i,1^k) : A_k(q,1^k) = 1\right] - \Pr[(q,s) \leftarrow Q(n,j,1^k) : A_k(q,1^k) = 1]\right| \le neg(k)$$

The probabilities are taken over all choices made by all machines involved.

There are two constructions of PIR schemes that are most relevant to our work. The construction by Kushilevitz and Ostrovsky [19] was the first to eliminate the need of multiple databases and was based on a computational assumption (the Quadratic Residue Assumption). The communication complexity of their construction is of the form $k \cdot n^{\epsilon}$ where k is the security parameter and n is the database size. A more efficient construction (in terms of communication complexity) was suggested by Cachin, Micali and Stadler [7]. It is based on a new computational hardness assumption – The ϕ-hiding assumption – and has communication complexity $O(\kappa)$ where κ is the length of composites for which the ϕ-hiding assumption holds. Note that there is also a lot of work related to PIR schemes with more than one sender, but they are less related to our setting.

As described in the Introduction, the two-round arguments analyzed in this paper are composed of several instances of a PIR scheme that are executed in parallel, with independent random coins for each instance. Let $\bar{i} = (i_1, i_2, \ldots, i_\ell)$ be a vector of (not necessarily distinct) locations. Given a PIR scheme (D, Q, R) let $PAR_{PIR}(\bar{i}, \mathcal{DB})$ be the two round protocol in which ℓ independent executions of (D, Q, R) are performed in parallel. For $j = 1 \ldots \ell$ the j'th execution queries location i_j on database $\mathcal{DB}$. Protocol PAR_{PIR} is rigorously defined in Appendix A.

2.3 Argument systems

Finally we introduce the object we are seeking – the low communication argument system.

Definition 2.3 (Two-round argument system) *Let* $\mathcal{V}_1, \mathcal{V}_2, \mathcal{P}$ *be probabilistic polynomial time algorithms. Let* k *be a security parameter.* $(\mathcal{V}_1, \mathcal{V}_2, \mathcal{P})$ *is a two-round argument system for a language* L *if*

Completeness: *For any* $x \in L$ *there exists a proof tape* π *s.t.*

$$\Pr[(q,s) \leftarrow \mathcal{V}_1(x,1^k); a \leftarrow \mathcal{P}(x,\pi,q,1^k) : \mathcal{V}_2(s,a,1^k) = 1] \ge 1 - neg(k)$$

Soundness: *For any* $x \notin L$ *for all probabilistic poly-time machines* $\mathcal{P}^*$

$$\Pr[(q,s) \leftarrow \mathcal{V}_1(x,1^k); a \leftarrow \mathcal{P}^*(x,q,1^k) : \mathcal{V}_2(s,a,1^k) = 1] \le neg(k)$$

The probabilities are over the random choices of all machines involved.

The communication complexity of an argument system is $|q|+|a|$. An argument system is *communication-efficient* if its communication complexity is $k \cdot \text{polylog}(|\pi|)$.

3 The basic two-round protocol

We now formalize and analyze the basic two-round arguments obtained by composing PCP systems with PIR schemes.

Definition 3.1 *Given the verifier* $\mathcal{V}$ *of a PCP system* $\mathcal{P}$ *for* $L \in NP$ *with randomness* r *and query complexity* ℓ*, and a PIR scheme* (D, Q, R)*, let the basic two-round protocol be as follows:*

Protocol $BASIC^{\mathcal{P}}(x)$:

Input: *Both parties hold a string x supposedly in L. If $x \in L$, the prover holds as auxiliary input a corresponding PCP proof tape π.*

Step 1: *The verifier simulates the PCP-verifier $\mathcal{V}$. Let $\bar{\imath}$ be the vector of locations accessed by $\mathcal{V}$.*

Step 2: *Both parties invoke protocol $PAR_{PIR}(\pi, \bar{\imath})$ where the verifier plays the chooser and the prover plays the sender. The verifier's outcome is $\bar{b}$.*

Step 3: *The verifier simulates $\mathcal{V}$ on $\bar{b}$ and accepts iff $\mathcal{V}$ accepts.*

The PIR access to locations $\bar{\imath} = (i_1, i_2, \ldots, i_\ell)$ is supposed to simulate access to these locations in the PCP tape π. Attempts to prove the correctness of $BASIC^{\mathcal{P}}$ were based on the privacy requirement of the PIR scheme [1]. It was suspected that the privacy of the PIR scheme forces the (malicious) prover to answer according to a particular proof tape π. We show that this is *not* the case.

4 A counterexample based on inconsistent replies

In this section we show that the construction suggested in [1] is not sound.

Observation 4.1 *There exists an explicit PCP system $\mathcal{P}$ such that for any computational PIR scheme, the corresponding argument system $BASIC^{\mathcal{P}}$ is not sound.*

Proof Our counter example uses a PCP constructed by Petrank [22] based on a reduction from Max-3-SAT to Max-3-Coloring. It is shown in [22] that for any NP language L and input x one can produce a graph G such that:

1. if $x \in L$, then G is 3-colorable.
2. if $x \notin L$, then any 3-coloring χ of G will miscolor at least a γ fraction of the edges of G for some constant $\gamma > 0$.

The implied PCP system $\mathcal{P}$ is as follows. Given any language $L \in$ NP and input x, the verifier constructs the graph G corresponding to x, chooses randomly an edge (i, j) in G, and queries the PCP proof π on locations i, j; each entry is in $\{1, 2, 3\}$. The verifier accepts iff $\pi_i \neq \pi_j$. In particular, if $x \notin L$ the verifier has probability at most $1 - \gamma$ of accepting.

The malicious prover's strategy for $BASIC^{\mathcal{P}}$ is as follows: Pick two different random colors $c_1 \neq c_2 \in \{1, 2, 3\}$. Answer the first query as if all the nodes are colored c_1 and the second one as if all of them are colored c_2. This strategy will convince the verifier with probability 1, independent of whether or not x is in L. Note that the cheating prover's strategy can be implemented regardless of the specific PIR scheme. All that is being used is that the prover is able to simulate each of the parallel PIR protocols in $BASIC^{\mathcal{P}}$ *with a different database*. □

4.1 Consistency checks

The counterexample described above exploits the following flaw of the basic protocol: A cheating prover may use *different databases* in its answers to the different PIR queries of the verifier. A standard approach for solving this problem is by incorporating *consistency checks* into $\mathcal{P}$. For example, the verifier can select its queries according to $\mathcal{P}$ but with probability $1/2$, instead of querying for the original list of locations, it picks at random one of these locations i_j and sends i_j in all instances of the PIR protocol. In such a case, the verifier accepts iff all the PIR answers it received are the same. Is it enough to enhance $BASIC^{\mathcal{P}}$ with consistency checks in order to make it sound? In such a system, a cheating prover can no longer get away with just giving answers from different databases. But is this the only kind of malicious behavior a $BASIC^{\mathcal{P}}$ prover (i.e. a PAR_{PIR} sender) can carry out?

Consider a variant of PIR that performs several queries at once. We denote such a primitive as Multi-PIR. In Appendix B we define the Multi-PIR primitive and show that the corresponding variant of protocol $BASIC$ in which the PAR_{PIR} sender is replaced by an arbitrary Multi-PIR sender is not sound, even when enhanced by consistency checks.

In the remainder of our work we concentrate on the basic protocol with PAR_{PIR} senders. We characterize in Section 5 the possible behaviors of a PAR_{PIR} sender, we define a type of malicious behaviors we call *spooky* and show that spooky behaviors (if exist) challenge the soundness of the basic protocol enhanced with consistency checks.

5 A deeper look into PIR privacy

PIR privacy served the main tool in attempts to prove the soundness of the basic protocol (see e.g. [1]). This is indeed a natural approach – the usage of the PIR correctness requirement for proving soundness seems beyond our current understanding of PIR schemes. What we show in the following is that PIR privacy is not enough for proving the soundness of the basic protocol as well as similar protocols.

In the following, we present an alternative definition of PAR_{PIR} privacy based on the *input output* behavior of PAR_{PIR}. We show that it suffices to prove the soundness of protocol $BASIC$ when we restrict ourselves to PAR_{PIR} protocols with input output behavior that obeys what we call the *projection condition*. We then study two types of behaviors which obey this property: tape distributions (which are easy to implement using any PIR scheme) and spooky behaviors (which are far from any tape distribution). Finally we show (in Section 6) that for *any* efficient PCP system $\mathcal{P}$ (no matter how sophisticated we make its checks) there exists a spooky behavior that challenges the soundness of protocol $BASIC$ with $\mathcal{P}$. This implies that any proof for the soundness of protocol $BASIC$ must rule out such spooky behaviors.

5.1 Behaviors

Let $\bar{i} = (i_1, \ldots, i_\ell)$ be a vector of locations, and $\bar{b} = (b_1, \ldots, b_\ell)$ be a vector of possible answers. We define the input-output behavior of PAR_{PIR} to be a function that takes a set of values $\bar{b}$ and an equinumerous set of indices $\bar{i}$ and determines the probability that invoking PAR_{PIR} on chooser input $\bar{i}$ results in chooser output $\bar{b}$. I.e. the probability that the outcome is $\bar{b}$ conditioned on the fact that the input is $\bar{i}$.

Definition 5.1 (Behavior P) *A behavior P is a function of $\bar{i} \in [n]^\ell$ and $\bar{b} \in \{0,1\}^\ell$, denoted $P[\bar{b}|\bar{i}]$ so that (i) $0 \leq P[\bar{b}|\bar{i}] \leq 1$ For every $\bar{i}, \bar{b}$ and (ii) $\sum_{\sigma \in \{0,1\}^l} P[\sigma|\bar{i}] = 1$ for every $\bar{i}$.*

The chooser of PAR_{PIR} is assumed to honestly follow her prescribed procedure. The sender, however, may follow any arbitrary strategy that is a family of polynomial-size probabilistic circuits $S^* = \{S_k^*\}_{k \in \mathbb{N}}$. It follows that the behavior of PAR_{PIR} is completely determined by the sender strategy S^*.

For every $\bar{i} \in [n]^\ell, \bar{b} \in \{0,1\}^\ell$, let $P^{S^*}[\bar{b}|\bar{i}]$ be the probability that the chooser's output is $\bar{b}$ given that its input is $\bar{i}$. The probabilities are over the coins of Q, R and S^*. Note that P^{S^*} is a behavior since $\sum_{\sigma \in \{0,1\}^l} P^{S^*}[\sigma|\bar{i}] = 1$ for every $\bar{i}$.

Definition 5.2 *Let (D, Q, R) be a PIR scheme. A behavior P* has efficient implementation under (D, Q, R) *if there exists a family of polynomial-size (probabilistic) circuits $\{S_k^*\}_{k \in \mathbb{N}}$ so that $P = P^{S^*}$.*

5.2 The projection condition

To define the projection condition we first define the input-output behavior also for *subsets of the ℓ locations* retrieved by the chooser. Let $\alpha = \{\alpha_1, \ldots, \alpha_m\}$ be a subset of $[\ell]$. For a vector of locations $\bar{i}$ define the α-projection of $\bar{i}$ as $\bar{i}_\alpha = (i_{\alpha_1}, \ldots, i_{\alpha_m})$. Similarly, for a vector of possible answers $\bar{b}$, define the α-projection of $\bar{b}$ as $\bar{b}_\alpha = (b_{\alpha_1}, \ldots, b_{\alpha_m})$. For α and $\sigma \in \{0,1\}^{|\alpha|}$, define $P_\alpha[\sigma|\bar{i}]$ to be $\sum_{\bar{b}_\alpha = \sigma} P[\bar{b}|\bar{i}]$.

$P_\alpha^{S^*}[\sigma|\bar{i}]$ thus equals the probability that the α-projection, $\bar{b}_\alpha$, of the answers, $\bar{b}$, received by the chooser equals to σ, given that the chooser's input is $\bar{i}$. We now define the two flavors of the projection condition.

Definition 5.3 (Projection condition) *Let P be a behavior. If for all $\alpha \subseteq [\ell]$, for all location vectors $\bar{i}, \bar{i}'$ such that $\bar{i}_\alpha = \bar{i}'_\alpha$ (i.e. their α-projection is equal) and for all $\sigma \in \{0,1\}^{|\alpha|}$ it holds that*

- $P_\alpha[\sigma|\bar{i}] = P_\alpha[\sigma|\bar{i'}]$, *then we say that P satisfies the* perfect projection condition
- $\left|P_\alpha[\sigma|\bar{i}] - P_\alpha[\sigma|\bar{i'}]\right| \leq neg(k)$, *then we say that P satisfies the* statistical projection condition

In the following we refer to the statistical projection condition *as the* projection condition.

5.2.1 Tape distributions and spooky interactions

There are some behaviors which obey the projection condition and have efficient implementation under any PIR scheme used. These are those where the sender chooses ℓ binary vectors of length n denoted $\pi_1, \pi_2, \ldots \pi_\ell$ and responds to query $(i_1, i_2, \ldots i_\ell)$ with $(\pi_1[i_1], \pi_2[i_2], \ldots \pi_\ell[i_\ell])$. We now define the resulting distributions:

Definition 5.4 (Tape Distribution) *A behavior that has an efficient implementation under any PIR scheme with a distribution on tapes $\pi_1, \pi_2, \ldots \pi_\ell$ is called a* tape distribution.

For example, the counterexample presented in Section 4 demonstrates a behavior that is a tape distribution. It is conceivable, however, that there exist behaviors that are far from any tape distribution but still obey the projection condition, we call such behaviors *spooky*.

Definition 5.5 (Spooky Behavior) *A behavior P is ε-spooky if it obeys the (statistical) projection condition and is of distance (L_1 norm) at least ε from any tape distribution.*

In general when the security parameter is fixed we say that a behavior is spooky if it is ε-spooky for a non-negligible ε.

5.2.2 Examples of behaviors obeying the projection condition

Example 5.1 (A tape distribution) *Let $\hat{P}$ be the following behavior. For all i_1, i_2 let $\hat{P}[(0,1)|(i_1,i_2)] = \hat{P}[(1,0)|(i_1,i_2)] = 1/2$. It follows that for all i_1, i_2 and for $\alpha = \{1\}$ or $\alpha = \{2\}$ it holds that $\hat{P}[(0,0)|(i_1,i_2)] = \hat{P}[(1,1)|(i_1,i_2)] = 0$ and $\hat{P}_\alpha[0|(i_1,i_2)] = \hat{P}_\alpha[1|(i_1,i_2)] = 1/2$. It is simple to verify that $\hat{P}$ satisfies the projection condition.*

The behavior $\hat{P}$ of Example 5.1 is essentially the same as the cheating prover's behavior in the counter example of Section 4. Indeed, $\hat{P}$ not only satisfies the projection condition but also has efficient PIR implementation *under any PIR scheme*. Define a sender strategy S^* as follows: pick random bit σ. Answer the first query as if all the entries of the database are σ and the second query as if all the entries are the complement of σ. As discussed above, such a strategy can be detected by consistency checks. That is, by allowing (with some probability) multiple queries to the same location. The strategy S^* will give such equal queries different answers, indicating that the sender deviated from the protocol. We now give a behavior $\tilde{P}$ that resists consistency checks and *still satisfies the projection condition*.

Example 5.2 (A spooky behavior) *Let $\tilde{P}$ be the following behavior. For all $i_1 \neq i_2$ let $\tilde{P}[(0,1)|(i_1,i_2)] = \tilde{P}[(1,0)|(i_1,i_2)] = 1/2$, and for all $i_1 = i_2$ let $\tilde{P}[(0,0)|(i_1,i_2)] = \tilde{P}[(1,1)|(i_1,i_2)] = 1/2$ where $i_i, i_2 \in n$. It is simple to verify that $\tilde{P}$ satisfies the projection condition.*

Note that for $n = 2$ the resulting behavior is a tape distributions – take $\pi_1 = \pi_2$ uniformly distributed over $\{01, 10\}$. However, when $n \geq 3$ it is not hard to see that the resulting behavior is spooky.

We do not know whether the behavior $\tilde{P}$ has an efficient implementation under some PIR scheme. On one hand, this seems counterintuitive since it means that the sender is able to correlate its answers to different queries in an unexpected way *even without learning anything about the queries (and in fact also about its answers)*. That is why we call such a behavior "spooky". On the other hand, PIR inherently uses a malleable encryption scheme so that the (honest) sender is able to manipulate the choosers 'encrypted' queries according to his database. (See also Appenxid C.)

5.2.3 The Projection Lemma

We show an equivalence between privacy and the projection condition. We use the following notation. For a vector of locations $\bar{\imath}$ denote by $(\bar{q}, \bar{s}) = (q(\bar{\imath}), s(\bar{\imath})) = Q(n, \bar{\imath}, 1^k) = Q(n, i_1, 1^k) \circ \cdots \circ Q(n, i_l, 1^k)$ the independent applications of Q to all the locations.

Lemma 5.3 *Let (D, Q, R) be a PIR scheme. The following conditions are equivalent:*

1. *PIR Privacy: for any family of polynomial-size (probabilistic) circuits $\{A_k\}_{k \in \mathbb{N}}$, for any $i \neq i' \in [n]$:*

$$\left|\Pr[(q,s) \leftarrow Q(n,i,1^k) : A_k(q,1^k) = 1] - \Pr[(q',s') \leftarrow Q(n,i',1^k) : A_k(q',1^k) = 1]\right| \leq neg(k)$$

2. *PAR_{PIR} privacy: for any family of polynomial-size (probabilistic) circuits $\{B_k\}_{k \in \mathbb{N}}$, for any vectors $\bar{\imath} \neq \bar{\imath}' \in [n]^\ell$:*

$$\left|\Pr[(\bar{q},\bar{s}) \leftarrow Q(n,\bar{\imath},1^k) : B_k(\bar{q},1^k) = 1] - \Pr[(\bar{q}',\bar{s}') \leftarrow Q(n,\bar{\imath}',1^k) : B_k(\bar{q}',1^k) = 1]\right| \leq neg(k)$$

3. *Every behavior P that has an efficient implementation under (D, Q, R) satisfies the projection condition.*

Proof (sketch)

$1 \rightarrow 2$**:** Let $\bar{\imath} = (i_1, \ldots, i_l), \bar{\imath}' = (i'_1, \ldots, i'_l)$ and B a distinguisher of $\bar{q} = q(\bar{\imath})$ and $\bar{q}' = q(\bar{\imath}')$. It follows by a standard hybrid argument that there exists an index j so that B distinguishes $(q(i_1), \ldots, q(i_{j-1}), q(i'_j), \ldots, q(i'_\ell))$ from $(q(i_1), \ldots, q(i_j), q(i'_{j+1}), \ldots, q(i'_\ell))$. Let $i = i_j, i' = i'_j$.

Construct a distinguisher A of $q(i), q(i')$ in contrast to PIR privacy. On input Inp^A, circuit A creates a query Inp^B for B as follows: let $Inp^B_1, \ldots, Inp^B_{j-1} = q(i_1), \ldots, q(i_{j-1})$, let $Inp^B_{j+1}, \ldots, Inp^B_\ell = q(i'_{j+1}), \ldots, q(i'_\ell)$ and $Inp^B_j = Inp^A$. A outputs $B(Inp^B)$.

$2 \to 3$**:** Suppose that the projection condition does not hold. Let S^* be a sender strategy, $\alpha \subset [\ell]$ and $\bar{i} \neq \bar{i}'$ vectors satisfying $\bar{i}_\alpha = \bar{i}'_\alpha$ such that for $\sigma \in \{0,1\}^{|\alpha|}$ it holds that $P_\alpha[\sigma|\bar{i}] - P_\alpha[\sigma|\bar{i}']$ is non-negligible. We construct a circuit B based on S^* that distinguishes $q(\bar{i})$ from $q(\bar{i}')$ (violating (2)). On input Inp^B circuit B creates a query Inp^{S^*} for S^* as follows:

1. Let $(q_\alpha, s_\alpha) \in_R Q(\bar{i}_\alpha)$. Let $Inp^{S^*}_\alpha = q_\alpha$. Thus, B can decrypt the response of S^* on locations α. Note that $Inp^{S^*}_\alpha$ and Inp^B_α are equally distributed.
2. Let $Inp^{S^*}_{[\ell]\setminus\alpha} = Inp^B_{[\ell]\setminus\alpha}$. (Surely $\alpha \neq [\ell]$ and thus Inp^B affects Inp^{S^*}.)

Given the reply of S^*, B decrypts $\bar{b}_\alpha$ and outputs 1 iff it equals σ.

$3 \to 1$**:** Suppose that the PIR Privacy property does not hold. Let $i \neq i'$ and A be a distinguisher of $q(i)$ from $q(i')$. Let $\bar{i} = (i, 1, \ldots, 1)$ be the vector with first location set to i and all other $\ell - 1$ locations set to 1. Similarly, let $\bar{i}' = (i', 1, \ldots, 1)$. Let $\alpha = [\ell] \setminus \{1\}$. We now construct a sender strategy S^* so that the induced behavior P^{S^*} does not obey the projection condition. In particular, the difference $P^{S^*}_\alpha[0^{|\alpha|}|\bar{i}] - P^{S^*}_\alpha[0^{|\alpha|}|\bar{i}']$ is non-negligible.

On input Inp^{S^*} let $Inp^A = Inp^{S^*}_1$ be the first query in Inp^{S^*}.

If $A(Inp^A) = 1$ then S^* uses D to coerce all answers to be 1, otherwise it uses D to coerce all answers to 0.

□

5.3 Consistency checks - revisited

Consider again the enhancement of consistency checks as described in Section 4.1. More specifically, let $\hat{\mathcal{P}}$ be the PCP of Section 4 enhanced with consistency checks. Given the graph G that corresponds to the input x, the verifier of $\hat{\mathcal{P}}$ randomly chooses an edge (i, j) in G and queries the PCP proof π on locations i, j (with probability $1/2$) or on locations i, i (with probability $1/2$).

Assume now that the prover of $BASIC^{\hat{\mathcal{P}}}$ can implement the behavior $\tilde{P}$ of Example 5.2. Recall that $\tilde{P}$ obeys the projection condition. Regardless of the graph G, the verifier always gets two distinct and random answers if it queries distinct locations (i, j) and the same random answer if it queries (i, i). This implies that under such an assumption $BASIC^{\hat{\mathcal{P}}}$ is not sound. Or in other words, *in order to prove the soundness of* $BASIC^{\hat{\mathcal{P}}}$ *one must first rule out the possibility of behavior* $\tilde{P}$ (which is a specific instance of the "spooky interactions" problem.).

6 The projection condition and arbitrary PCPs

Our counterexamples of Sections 4 and 5 were based on a specific PCP scheme and a specific way to perform consistency checks. In one case, we have shown that the resulting argument system is not sound. In the other case, proving the soundness of the argument system will require proving that a particular spooky behavior cannot be implemented by PIR schemes. Such a proof in general seems non-trivial (especially given the equivalence we have shown between PIR privacy and the projection condition). A possible alternative is to design a particular (communication efficient) PIR scheme that avoids the spooky behavior of our counterexample. An obvious obstacle in this direction, is the small number of communication efficient PIR schemes that exist in the literature (still, this may be the most promising research direction).

But – there seem to be another way out. *Forget about PIRs, and instead concentrate on PCPs ...* Perhaps by carefully choosing a PCP system (that may incorporate more sophisticated checks) we can prove the soundness of the corresponding argument system, without resolving the problem of spooky interactions. In this section we study this possibility and come up with a negative result. For every efficient PCP, there exists a behavior that challenges the soundness of the resulting protocol, while satisfying the projection condition. Moreover, there exists an *effective* procedure for computing this behavior. It is not clear, however, whether such behaviors have efficient implementation or not. Still, the results of this section show that no matter what PCP we use, proving the soundness of the resulting argument system reduces to an instance of the "spooky interactions" problem.

6.1 The class $PROJ(\ell)$

Since we are seeking a proof that avoids the problem of spooky behaviors, let us assume that there exist PIR schemes that are "fully malleable" (to borrow a term from the related context of [8])) in the sense that every behavior P that satisfies the projection condition has an efficient implementation under such PIR schemes. Under this assumption we ask the following question: Is there a PCP for a non-trivial language L such that (1) The query complexity ℓ, of the PCP is small. (2) The corresponding two-round argument system is sound?

The existence of such a PCP will imply that for every $x \notin L$, no behavior that satisfies the projection condition convinces the verifier. Otherwise, since we assumed that "fully malleable" PIR schemes exist, the cheating prover can convince the verifier that x is in L, thus contradicting the soundness of the argument system[2]. Furthermore, note that for such a PCP, for every $x \in L$ there in a behavior that satisfies the (perfect) projection condition and convinces the verifier. This is just the behavior of the honest prover. At this point, our argument system reduces to a proof system in the following setting:

Consider a verifier $\mathcal{V}$ that interacts with a (computationally unbounded) prover in the clear (i.e. the verifier just sends the queries $\bar{i}$ and receives the answers $\bar{b}$). Nevertheless, we restrict the prover's answers to be consistent with some behavior P that satisfies the projection condition. (The behavior P is not assumed to have an efficient implementation.) Let $\mathcal{V}(x, P)$ denote the random variable whose value is 0 or 1 according to whether $\mathcal{V}$ accepts the common input x when interacting with behavior P.

Definition 6.1 *A language L is in $PROJ(\ell)$ if there exists a probabilistic polynomial verifier $\mathcal{V}$ using ℓ queries such that*

1. *If $x \in L$ then there exists a behavior P that satisfies perfect projection condition s.t.* $\Pr[\mathcal{V}(x, P) = 1] \geq 2/3$.
2. *If $x \notin L$ then for all behaviors P^* that obey the projection condition it holds that* $\Pr[\mathcal{V}(x, P^*) = 1] \leq 1/3$.

The probabilities are over the random choices made by both machines.

Given our assumptions on the PCP for the language L (and on the malleability of PIRs), we can deduce that $L \in PROJ(\ell)$. The following theorem shows that if ℓ is small (equivalently, if the argument system is communication efficient), then L is relatively easy (and in particular, is unlikely to be NP-complete).

Theorem 6.1 $PROJ(\ell) \subseteq BPTime(2^{O(\ell \log(n))})$.

Proof Let L be a language in $PROJ(\ell)$ and $\mathcal{V}$ be the corresponding verifier. Given an input instance x, we create a linear program whose objective is to find a behavior P that satisfies perfect projection condition and

[2] We discuss the problem of "finding" the malicious behavior in Remark 6.2.

maximizes the acceptance probability of the verifier $\mathcal{V}$. If the acceptance probability of $\mathcal{V}$ on P is greater than $1/2$ then we conclude that $x \in L$ otherwise we conclude that $x \notin L$.

Recall the definition of a behavior P and the corresponding values $P[\bar{b}|\bar{i}], P_\alpha[\sigma|\bar{i}]$ (see Section 5.2). We abuse notation and use $P[\bar{b}|\bar{i}], P_\alpha[\sigma|\bar{i}]$ as the variables of our linear program.

Fix the common input x. Let $\mathcal{V}[\bar{i}]$ be the probability that $\mathcal{V}$ queries locations $\bar{i} = (i_1, \ldots, i_\ell)$. Given that $\mathcal{V}$ queries $\bar{i}$, let $\mathcal{V}[\bar{b}|\bar{i}]$ be the probability that $\mathcal{V}$ accepts on answers $\bar{b}$. Both probabilities are taken over the coin tosses of $\mathcal{V}$ and are implicitly defined by $\mathcal{V}$. We define the coefficients of the linear program as $\mathcal{V}^{\bar{i}}_{\bar{b}} = \mathcal{V}[\bar{i}] \cdot \mathcal{V}[\bar{b}|\bar{i}]$. The acceptance probability of the verifier $\mathcal{V}$ on P is $\mathcal{V}(x, P) = \sum_{\bar{i}} \mathcal{V}[\bar{i}] \sum_{\bar{b}} \mathcal{V}[\bar{b}|\bar{i}] \cdot P[\bar{b}|\bar{i}] = \sum \mathcal{V}^{\bar{i}}_{\bar{b}} \cdot P[\bar{b}|\bar{i}]$. We can now define the corresponding linear program (LP).

$$
\begin{array}{llll}
\textit{(LP)} & \text{Maximize} & \sum_{\bar{i},\bar{b}} \mathcal{V}^{\bar{i}}_{\bar{b}} \cdot P[\bar{b}|\bar{i}] \quad \text{s.t.} & \\
 & (1) & P_\alpha[\sigma|\bar{i}] = P_\alpha[\sigma|\bar{i}'] & \forall\, \alpha \subseteq [\ell]; \sigma \in \{0,1\}^{|\alpha|}; \bar{i}, \bar{i}' \text{ s.t. } \bar{i}_\alpha = \bar{i}'_\alpha \\
 & (2) & P_\alpha[\sigma|\bar{i}] = \sum_{\bar{b}_\alpha = \sigma} P[\bar{b}|\bar{i}] & \forall\, \alpha \subseteq [\ell]; \sigma \in \{0,1\}^{|\alpha|} \\
 & (3) & \sum_{\sigma \in \{0,1\}^l} P[\sigma|\bar{i}] = 1 & \forall\, \bar{i} \\
 & (4) & P[\sigma|\bar{i}] \geq 0 & \forall\, \sigma \in \{0,1\}^l; \bar{i}
\end{array}
$$

Conditions (1) and (2) of LP enforce perfect projection condition on P.[3] Conditions (3) and (4) restrict $P[\sigma|\bar{i}]$ to be a probability distribution.

The size of the linear program is $2^{O(\ell \log(n))}$. Thus assuming that the coefficients $\mathcal{V}^{\bar{i}}_{\bar{b}}$ are known, the LP may be solved in time $2^{O(\ell \log(n))}$.

Computing the exact values $\mathcal{V}^{\bar{i}}_{\bar{b}}$ by going over all the possible coin tosses of $\mathcal{V}$ takes time exponential in the randomness of $\mathcal{V}$, thus it may be larger than $2^{O(\ell \log(n))}$. Instead we show how to approximate $\mathcal{V}^{\bar{i}}_{\bar{b}}$ in time $2^{O(\ell \log(n))}$ such that solving the linear program with these *approximated coefficients* changes the value of the objective function by less than $1/6$. Thus using such an *approximated* linear program, membership in the language L can be decided in time $2^{O(\ell \log(n))}$.

The number of coefficients $\mathcal{V}^{\bar{i}}_{\bar{b}}$ is $2^{O(\ell \log(n))}$, each of them appears in the objective function of LP once, multiplied by a positive variable bounded by 1. Hence, it suffices to approximate each $\mathcal{V}^{\bar{i}}_{\bar{b}}$ within additive error $2^{-O(\ell \log(n))}$ with probability $1 - 2^{-O(\ell \log(n)) - n}$.

We approximate $\mathcal{V}^{\bar{i}}_{\bar{b}}$ by sampling t random strings of length r. Let $\hat{\mathcal{V}}^{\bar{i}}_{\bar{b}}$ be the fraction of strings on which $\mathcal{V}$ queries $\bar{i}$ and accepts if answered $\bar{b}$. Using the Chernoff bound if $t \geq 2^{O(\ell \log(n))}$ then the difference between $\mathcal{V}^{\bar{i}}_{\bar{b}}$ and $\hat{\mathcal{V}}^{\bar{i}}_{\bar{b}}$ is smaller than $2^{-O(\ell \log(n))}$ with probability at least $1 - 2^{-O(\ell \log(n)) - n}$. Thus for any behavior P it holds with probability at least $1 - 2^{-n}$ that

$$|\sum \mathcal{V}^{\bar{i}}_{\bar{b}} P[\bar{b}|\bar{i}] - \sum \hat{\mathcal{V}}^{\bar{i}}_{\bar{b}} P[\bar{b}|\bar{i}]| < 1/6.$$

□

Remark 6.2 *As noted, our argument above neglects the question of how the prover "finds" the malicious spooky behavior. As the proof of Theorem 6.1 shows, for every instance x it is possible to find in time*

[3] Condition (1) can be rephrased to capture the (computational) projection condition by allowing slackness. We defer the treatment of computational projection to the full version of this work.

$2^{O(\ell \log(n))}$ *the* best *behavior $\mathcal{P}$ (in terms of verifier acceptance probability). If ℓ is constant, the prover can therefore find is cheating behavior in polynomial time. More generally, consider a variant of our definition of argument systems, in which the prover is restricted to be a circuit of size $T(k)$ (which may be super-polynomial in k). (Assuming the PIR scheme is private against circuits of size $T(k)$ the projection condition still holds.) In this case, the prover can still cheat as long as $\ell \leq \log(T(k))/\log(n)$. Finally, when $\ell > \log(T(k))/\log(n)$, we can at least deduce that proving the soundness of the basic protocol cannot solely rely on the projection condition, but should use the restriction on the prover's computational power to guarantee a stronger property.*

7 Conclusions and Open Problems

We have analyzed the construction of argument systems by a composition of PCPs and PIR schemes (as suggested in [1]). Suppose that the interactive proof is based on some *sound* PCP where each verifier's query is done via an independent PIR. Then we would like to make the following argument: "every prover's strategy can be translated into a PCP proof tape (or distribution over tapes) that has a similar probability of success (with respect to the PCP verifier)." We have shown inherent difficulties in making such a simulation argument, regardless of the specific PCP in use. The problem is that the although the prover does not really understand the queries may answer then in a *correlated* manner (again, without really understanding the answers).[4] The possibility of "spooky interactions" (according to Definition 5.5) appears to be a fundamental problem that must be resolved in order to construct argument systems using the approach of [1].

In Appendix D we further discuss two issues that are closely related to the topic of our investigation: how to reduce the soundness error of two-round arguments via parallel repetition and how to achieve witness indistinguishability.

The following open problems seem to be of particular interest.

1. Does there exist a computational PIR where "spooky interactions" may occur? A negative answer would represent substantial progress in our understanding of PIRs and would be especially interesting given the observations of this paper.

   A possible direction for giving a positive answer is to construct artificial PIRs with communication complexity which is larger than the data size that allow "spooky interactions". In Appendix B, we introduce a variant of PIR, that performs several queries at once – Multi-PIR. We show that Multi-PIR does allow "spooky interactions". Furthermore, in Appendix C we show an information theoretic PIR where "spooky interactions" occur.

2. Is it possible to construct a "non-malleable computational PIR" in which such a behavior can be ruled out?

Acknowledgments

We thank the authors of [1] for providing us a preprint of their paper. We thank Bill Aiello and Tal Malkin for discussions of the basic protocol and PIR schemes, and Salil Vadhan for discussions of the results of this paper.

[4] This issue is closely related to malleability in cryptosystems (see [8]). An example where one player's independence does not protect it from the other player's coordination is in 4-round computational games (see [5]).

References

[1] W. Aiello, S. Bhatt, R. Ostrovsky and S. Rajagopalan. *Fast Verification of Any Remote Procedure Call: Short Witness-Indistinguishable One-Round Proofs for NP,* Proc. 27th International Colloquium on Automata, Languages and Programming, (ICALP 2000), Lecture Notes in Computer Science 1853, Springer, pp. 463-474 (withdrawn).

[2] S. Arora, C. Lund, R. Motowani, M. Sudan and M. Szegedy. *Proof verification and hardness of approximation problems.* J. of the ACM 45(3),1998, pp. 501-555.

[3] S. Arora and S. Safra. *Probabilistic checking of proofs: A new characterization of NP.* J. of the ACM 45(1), 1998, pp. 70–122.

[4] B. Barak, *How to Go Beyond The Black-Box Simulation Barrier*, Proc. 42nd IEEE Symposium on Foundations of Computer Science, 2001.

[5] M. Bellare, R. Impagliazzo and M. Naor. *Does Parallel Repetition Lower the Error in Computationally Sound Protocols?*, Proc. 38th IEEE Symposium on Foundations of Computer Science, 1997, pp. 374–383.

[6] M. Ben-Or, S. Goldwasser, J. Kilian and A. Wigderson. *Multi-prover interactive proofs: How to remove the intractability assumptions,* Proc. 20th ACM Symposium Theory of Computing, pp. 113-131, 1988.

[7] C. Cachin, S. Micali and M. Stadler. *Computationally Private Information Retrieval With Polylogarithmic Communication,* Advances in Cryptology – Eurocrypt '99, Lecture Notes in Computer Science 1592, Springer, 1999, pp. 402–414.

[8] D. Dolev, C. Dwork and M. Naor. *Non-malleable Cryptography.* Siam J. on Computing 30(2), 2000, pp. 391–437.

[9] C. Dwork, U. Feige, J. Kilian, M. Naor, M. Safra, *Low Communication 2-Prover Zero-Knowledge Proofs for NP,* Advances in Cryptology – CRYPTO '92, Lecture Notes in Computer Science, Springer 740, pp. 215–227.

[10] C. Dwork and M. Naor. *Zaps and Their Applications*, Proc. 41st IEEE Symposium on Foundations of Computer Science, 2000, pp. 283–293.

[11] L. Fortnow, J. Rompel and M. Sipser. *On the Power of Multi-Prover Interactive Protocols*, Theoretical Computer Science A, 134:545-557, 1994.

[12] Y. Gertner, T. Malkin and O. Reingold, *On the impossibility of basing trapdoor functions on trapdoor predicates*, Proc. 42nd IEEE Symposium on Foundations of Computer Science, 2001.

[13] O. Goldreich and J. Håstad. *On the Complexity of Interactive Proofs with Bounded Communication.* Information Processing Letters 67(4), 1998, pp. 205–214.

[14] O. Goldreich, H. Krawczyk, *On the Composition of Zero-Knowledge Proof Systems*, SIAM J. Comput. 25(1), pp. 169–192, 1996.

[15] O. Goldreich, S. P. Vadhan, A. Wigderson, *On Interactive Proofs with a Laconic Prover*, Proc. 28th International Colloquium on Automata, Languages and Programming, (ICALP 2001), Lecture Notes in Computer Science 2076, Springer 2001, pp. 334–345.

[16] R. Impagliazzo and S. Rudich, *Limits on the provable consequences of one-way permutations*, Proceedings of the ACM Symposium on Theory of Computing (STOC), pp. 44–61, 1989.

[17] J. Kilian. *A note on efficient zero-knowledge proofs and arguments,* Proc. of the Twenty-Fourth Annual ACM Symposium on the Theory of Computing, 1992, pp. 723–732.

[18] J. Kilian. *Improved efficient arguments.* Advances in Cryptology—CRYPTO '95, Lecture Notes in Computer Science, vol. 963, Springer-Verlag, 1995, pp. 311–324.

[19] E. Kushilevitz and R. Ostrovsky, *Replication Is Not Needed: Single Database, Computationally-Private Information Retrieval,* Proc. 38th IEEE Symposium on Foundations of Computer Science, 1997, pp. 364–373.

[20] E. Kushilevitz and R. Ostrovsky, *One-Way Trapdoor Permutations Are Sufficient for Non-trivial Single-Server Private Information Retrieval,* Advances in Cryptology – Eurocrypt'2000, Lecture Notes in Computer Science 1807, Springer, 2000, pp. 104–121.

[21] S. Micali, *CS proofs,* Proc. 35th IEEE Symposium on Foundations of Computer Science, 1994, pp. 436–453.

[22] E. Petrank, *The hardness of approximation: Gap location,* Computational Complexity 4, 1994, pp. 133–157.

[23] R. Raz. *A Parallel Repetition Theorem.* SIAM J. Comput. 27(3), 1998, pp. 763–803.

A Protocol PAR_{PIR}

Protocol $PAR_{PIR}(\bar{i}, \mathcal{DB})$:

Input: The chooser holds a vector of (not necessarily distinct) locations $\bar{i} = (i_1, i_2, \ldots, i_\ell)$ where $i_j \in [n]$. The sender holds a database $\mathcal{DB}$ of length n.

Step 1: The chooser prepares, for each $j \in [\ell]$, a corresponding query $(q_j, s_j) \leftarrow Q(n, i_j, 1^k)$.
The random coins used in the construction of each query are chosen independently.
The chooser sends all queries $q_1, \ldots, q_\ell$ to the sender.

Step 2: The sender prepares for each query q_j an answer $a_j \leftarrow D(\mathcal{DB}, q_j, 1^k)$.
The sender sends the answers $a_1, \ldots, a_\ell$ to the chooser.

Step 3: The chooser recovers the entries of $\mathcal{DB}$ in locations $i_1, i_2, \ldots, i_\ell$, i.e. the sequence of bits $\bar{b} = (b_1, b_2, \ldots, b_\ell)$ where $b_j \leftarrow R(n, i_j, (q_j, s_j), a_j, 1^k)$.

B Spooky interactions for Multi-PIR

The parties of a Multi-PIR are a chooser and a sender. The sender holds a database $\mathcal{DB}$ of length n and the chooser holds a vector of l indices $\bar{i} = (i_1 \ldots i_l) \in [n]^l$. Intuitively, the protocol enables the chooser to read locations $\bar{i}$ of $\mathcal{DB}$ without the sender learning anything about $\bar{i}$. As in the PIR setup, the Multi-PIR scheme is defined by three algorithms: Q and R are executed by the chooser and D by the sender. Algorithm Q is the query generator that maps the indices $\bar{i}$ into a query q using the chooser's secret random string s. Algorithm D is the one executed by the (honest) prover in response to query q and as a function of the database $\mathcal{DB}$. Finally, algorithm R allows the chooser to reconstruct the values at locations $\bar{i}$ as a function of the response it received from the sender and the secret random string s.

Definition B.1 (Multi-PIR) *Let D, Q, R be probabilistic polynomial time algorithms. Let k be a security parameter, l be the number of queries and n the length of the database. (D, Q, R) is a (computational) Multi-PIR scheme if*

Correctness: *For any n-bit string $\mathcal{DB}$ and any l locations $\bar{i} = (i_1 \ldots i_l) \in [n]^l$,*

$$\Pr\left[(q, s) \leftarrow Q(n, \bar{i}, 1^k); a \leftarrow D(\mathcal{DB}, q, 1^k) : R(n, \bar{i}, (q, s), a, 1^k) = \mathcal{DB}[\bar{i}]\right] \geq 1 - neg(k)$$

where $\mathcal{DB}[\bar{i}] = (\mathcal{DB}[i_1] \ldots \mathcal{DB}[i_l])$.

Privacy: *For any family of polynomial-size (probabilistic) circuits $\{A_k\}_{k\in\mathbb{N}}$ and any $\bar{i}, \bar{j} \in [n]^l$,*

$$\left|\Pr\left[(q,s) \leftarrow Q(n,\bar{i},1^k) : A_k(q,1^k) = 1\right] - \Pr[(q,s) \leftarrow Q(n,\bar{j},1^k) : A_k(q,1^k) = 1]\right| \leq neg(k)$$

The probabilities are taken over all choices made by all machines involved. Notice that given a PIR scheme (D,Q,R), l independent executions of (D,Q,R) are a Multi-PIR.

Theorem B.1 *Let $\mathcal{P}$ be the PCP system defined in Observation 4.1, enhanced with consistency checks. There exists a Multi-PIR $\mathcal{MP}$ such that protocol $BASIC$ with $\mathcal{P}$ and $\mathcal{MP}$ is not sound.*

Proof Let (D,Q,R) be the PIR scheme presented in [19]. Roughly speaking, this PIR scheme is based on the Quadratic Residue Assumption and is structured as follows.

1. Chooser: Pick a random number N which is the multiplication of two primes. On query $i \in [n]$ pick n random numbers $y_1 \ldots y_n$ (with Jacobi symbol of value 1) such that y_i in QNR_N (quadratic non-residues mod$N$) and y_j for $j \neq i$ in QR_N (quadratic residues mod$N$). Define $Q(i) = (q,s)$ where $q = (y_1 \ldots y_n; N)$ and s are the factors of N.

2. Sender : $D(\mathcal{DB},q) = a = \prod_{i=1}^n y_i^{2-\mathcal{DB}[i]}$.

3. Chooser : $R(n,i,(q,s),a) = 1$ if $a \in QNR_N$ and 0 otherwise.

We will construct a Multi-PIR $(\hat{D},\hat{Q},\hat{R})$ that runs two parallel copies of (D,Q,R), with a slight twist. In addition to the standard encrypted queries that are sent from the chooser to the sender, we would like the chooser to send additional information we call *advise*. This advise is to be ignored by an honest $\hat{D}$ but is constructed such that it can (and will) be used by a malicious sender S^* in order to fool the chooser.

1. For $\bar{i} = (i_1,i_2)$ let $\hat{Q}(\bar{i}) = (q,s)$ where q and s are defined below. Let $(q_1,s_1) = Q(i_1)$, and $(q_2,s_2) = Q(i_2)$, and denote the modulae used in the instances of the PIR schemes by N_1, N_2 respectively. To construct the advice pick a random $\sigma \in \{0,1\}$. If $i_1 = i_2$ and $\sigma = 0$ let $advice_1 \in_R QR_{N_1}$ and $advice_2 \in_R QR_{N_2}$. If $i_1 = i_2$ and $\sigma = 1$ let $advice_1 \in_R NQR_{N_1}$ and $advice_2 \in_R NQR_{N_2}$. Similarly, if $i_1 \neq i_2$ and $\sigma = 0$ let $advice_1 \in_R QR_{N_1}$ and $advice_2 \in_R NQR_{N_2}$. finally if $i_1 \neq i_2$ and $\sigma = 1$ let $advice_1 \in_R NQR_{N_1}$ and $advice_2 \in_R QR_{N_2}$. Define q to be $(q_1,q_2,advice_1,advice_2)$ and s to be (s_1,s_2).

2. For a database $\mathcal{DB}$, and an encrypted query $q = (q_1,q_2,advice_1,advice_2)$ let $\hat{D}(\mathcal{DB},q) = a = (a_1,a_2)$ where $a_i = D(\mathcal{DB},q_i)$.

3. For $a = (a_1,a_2)$, $q = (q_1,q_2,advice_1,advice_2)$, and $s = (s_1,s_2)$ let $\hat{R}(n,\bar{i},(q,s),a)$ be the pair $R(n,i_1,(q_1,s_1),a_1), R(n,i_2,(q_2,s_2),a_2)$.

It is not hard to verify that the above is a Multi-PIR. Consider the behavior we obtain by Multi-PIR $(\hat{D},\hat{Q},\hat{R})$ with a malicious Sender S^* that on input $q = (q_1,q_2,advice_1,advice_2)$ answers $(advice_1,advice_2)$. If the Sender was to query locations i_1 and i_2 which differ, it would receive two answers that differ (either the pair (0,1) or the pair (1,0), each with probability 1/2). Similarly if the Sender was to query locations i_1 and i_2 which are equal, it would receive two identical answers (either (0,0) or (1,1), each with probability 1/2). We conclude that behavior $\tilde{\mathcal{P}}$ of Example 5.1 has an efficient implementation under the Multi-PIR we have presented, thus implying our assertion. □

Theorem B.1 can be extended to general PCP systems when we consider defining Multi-PIRs under a non-uniform setting.

Theorem B.2 *Let $\mathcal{P}$ be any PCP system for an NP-Complete language L with l queries and randomness r, where l and r are at most logarithmic in the instance size. Unless $NP = P$ there exists a non-uniform Multi-PIR $\mathcal{MP}$ such that protocol $BASIC$ with $\mathcal{P}$ and $\mathcal{MP}$ is not sound.*

Proof (**Sketch**) Let $\mathcal{V}$ be the PCP verifier of $\mathcal{P}$. Assume that $\mathcal{V}$ performs l queries, has randomness r, completeness δ and soundness ε. The proof idea is similar to that of Theorem B.1. We construct a Multi-PIR $\mathcal{MP}$ that helps the malicious prover convince the verifier, with probability more than ε for at least one instance $\hat{x} \notin L$.

The advice computed by the sender for a query $\bar{i}$ is the answer that maximizes the acceptance probability on $\hat{x}$. Note that given $\hat{x}$ this advice is efficiently computable by going over all possible coin tosses of $\mathcal{V}$ and all possible answers to query $\bar{i}$. This process also computes the acceptance probability given the advice. We now show that, unless $NP = P$, for infinitely many values of n there exist instances $\hat{x}$ of length n for which this probability is at least δ.

By Definition 2.1 we have that for all $x \in L$ there exists an assignment to the PCP tape so that $\mathcal{V}$ accepts with probability at least δ. Hence, if we apply the algorithm for computing the acceptance probability with the best advice (as above, using x instead of $\hat{x}$), this probability must be at least δ. If for all but a finite number of $x \notin L$ it is the case that the algorithm for computing the acceptance probability for the best advice returns a value less than δ we conclude that one can decide L in polynomial time. $\Box$

C Spooky interactions for information theoretic PIRs

Although we do not know how to construct *computational* PIR schemes with spooky interactions, it is fairly easy to do that with information theoretic schemes. Consider the following information theoretic PIR scheme with two senders (S_1, S_2). The chooser selects a random vector $r = r_1 \ldots r_n \in \{0,1\}^n$, sends this vector to S_1 and the same vector with the i'th bit flipped $\hat{r}$ to S_2. Each sender returns the bitwise multiplication of its input and the database B. The chooser compares the i'th bit in both answers and outputs 1 if they differ and zero otherwise.

We construct a malicious sender strategy S_1^*, S_2^* that implements the behavior of example 5.2, for two parallel queries. Although the chooser send two queries, both senders ignore the second one. On input r and r' the first sender S_1^* returns two vectors identical to r. On input $\hat{r}$ and $\hat{r}'$, the second sender S_2^*, flips a random coin $\sigma \in \{0,1\}$, and returns two vectors identical to $\sigma \oplus \hat{r}$ (i.e each bit of $\hat{r}$ is XORed with σ).

It is easy to see that the behavior implied by S_1^*, S_2^* is that of example 5.2.

D Parallel Repetition and Witness Indistinguishability

We now discuss two issues related to the execution of arguments systems. First, regarding soundness, we have concentrated on achieving a constant probability of error. Is this justified? The natural answer is yes, since we can rapidly decrease the probability of error by running several such argument systems in parallel multiplying the communication complexity by $-\log$ the error for which we are aiming. However, one must be careful, since we are discussing a computational game and perhaps there are "spooky interactions" between the copies of the game (as well as inside each game). However here we can use the fact that what we have is a 2-round computational game. For such games Bellare, Impagliazzo and Naor [5] showed that by parallel repetition the probability of failure goes down exponentially with the number of repetitions. (This holds for any computational game with up to 3 rounds, but is not necessarily true for 4-round games.) Also note that the *two-prover* parallel repetition Theorem of [23] is not applicable here since the prover has a (non-efficient) strategy of winning in each individual game.

Another goal of [1] was to achieve low communication witness indistinguishability in two rounds, i.e. that the verifier would not be able to distinguish which witness the prover is using (obtaining zero-knowledge in two-rounds is much harder and cannot be done via black-box reductions [14].) A possible approach is to base it on a multi-prover proof system that is zero-knowledge, such as [9] as well as a Secure PIR scheme. However, all the difficulties encountered in the previous sections must be resolved.

A different approach is to base it on Zaps [10], which are two-round witness-indistinguishable proof system where the verifier's first message σ can be fixed once and for all. Every language $L \in NP$ has a zap (provided some cryptographic assumption is true). The problem with using them is that the verifier should send a long message. However, suppose that the zap part σ of the verifier's message has been fixed (either by a clever combinatorial construction or by the system somehow). Then we have reduced the problem of constructing a low-communication witness-indistinguishable argument system for a language $L \in NP$ to the problem of constructing an argument system for another language $L' \in NP$ where L' consists of pairs (x, σ) where there is a proof y that $x \in L$ with first message σ by the verifier.