The Computational Complexity Column

by

Lance Fortnow

Department of Computer Science, University of Chicago
1100 East 58th St., Chicago, IL 60637 USA
fortnow@cs.uchicago.edu
http://www.cs.uchicago.edu/~fortnow/beatcs

In the June issue Jacobo Torán will take over as editor of this column and I look forward to many exciting columns under his direction.

I would like to thank Grzegorz Rozenberg who served as BEATCS editor during most of my tenure as editor and convinced me to stay on an extra year. I also would like to thank my predecessors Eric Allender and Juris Hartmanis for building a strong reputation for this column and the current BEATCS editor Vladimiro Sassone. Most of all I would like the to thank all of the authors who have produced wonderful survey articles while I was editor: Scott Aaronson, Eric Allender, Stephen Fenner, Eldar Fischer, William Gasarch, Evan Golub, Clyde Kruskal, Steve Homer, Valentine Kabanets, Dieter van Melkebeek, Kenneth Regan and Ronen Shaltiel.

How can one query a database to get information without revealing the question asked? Bill Gasarch surveys this area of Private Information Retrieval.

A Survey on Private Information Retrieval

William Gasarch*

University of Maryland at College Park

Abstract

Alice wants to query a database but she does not want the database to learn what she is querying. She can ask for the entire database. Can she get her query answered with less communication? One model of this problem is *Private Information Retrieval*, henceforth PIR. We survey results obtained about the PIR model including partial

*University of Maryland, Dept. of Computer Science and Institute for Advanced Computer Studies, College Park, MD 20742. gasarch@cs.umd.edu

answers to the following questions. (1) What if there are k non-communicating copies of the database but they are computationally unbounded? (2) What if there is only one copy of the database and it is computationally bounded?

1 Introduction

Consider the following scenario. Alice wants to obtain information from a database but does not want the database to learn which information she wanted. One solution is for Alice to ask for the entire database. Can she obtain what she wants with less communication?

The earliest references for problems of this sort are Rivest et al. [55], Blakely [15] and Feigenbaum [30]. The model in [30] was refined by Abadi et al. [1]. This refined model was the basis for several later papers [7, 6].

We will consider a later formulation by Chor et al. [23]. We model a database as (1) an n-bit string $x = x_1x_2 \cdots x_n$, together with (2) a computational agent that can do computations based on both x and queries made to it. Alice wants to obtain x_i such that the database does not learn i. Actually Alice wants more than that– she wants the database to have absolutely no hint as to what i is. For example, if the database knows that $i \neq 98$ then Alice will be unhappy. Alice can achieve this level of privacy by asking for all n bits. Can she obtain x_i with complete privacy by a scheme that uses fewer than n bits of communication? We assume that the database knows Alice's scheme and can simulate it. This question has several answers.

1. If Alice uses a deterministic scheme then n bits are required. (This is folklore.) This holds even if there are several non-communicating copies of the database. Hence, for the rest of the paper, we assume Alice can flip coins. Despite this, we will require that she always obtains the correct answer.

2. If the database has unlimited computational power and there is only one copy of the database then n bits are required [23].

3. Let $k \geq 2$. Assume there are k non-communicating copies of the database. We also assume that the databases have unlimited computational power. The following are known.

    (a) Chor et al. [23] have a scheme that uses $O((k \lg k)n^{1/\lg k})$ bits.

    (b) Chor et al. [23] have a scheme that probably uses $O((k \lg k)n^{1/(\lg k + \lg\lg k)})$ bits. (The status of the number of bits depends on some open problems in coding theory.)

    (c) Chor et al. [22] have a scheme that uses $O((k^2 \log k)n^{1/k})$ bits.

    (d) Ambainis [2] have a scheme that uses $O(2^{k^2} n^{1/(2k-1)})$ bits.

    (e) Ishai and Kushilevitz [40, 10] have a scheme that uses $O(k^3 n^{1/(2k-1)})$ bits.

    (f) Beimel et al. [12] have a scheme that uses $n^{O(\lg\lg k/k \lg k)}$ bits.

    The last result is currently the best known even for small k (see Table 1).

4. Chor and Gilboa [19] show that if there exists one-way functions then there exists a scheme that uses two copies of the database and $O(n^\epsilon)$ bits where ϵ can be taken arbitrarily small.

5. Kushilevitz and Ostrovsky [47] show that if the database cannot solve the Quadratic Residue problem (QR problem) then there is 1-DB scheme that uses $O(n^\epsilon)$ bits, where ϵ can be taken arbitrarily small.

6. Cachin et al. [17] show that if the database cannot solve the ϕ-hiding problem then there is a 1-DB probabilistic scheme that uses $O((\lg n)^a)$ bits. where a depends on how hard the ϕ-hiding problem is. (The ϕ-hiding problem was first defined in [17].)

7. Kushilevitz and Ostrovsky [48] show that if there exist one-way permutations with a trapdoor then there is a 1-DB scheme that uses $n - o(n)$-bits.

8. Beimel et al. [11] showed that

9. Di-Crescenzo et al. [29] showed that if there is a sublinear 1-DB scheme then there exist an oblivious transfer protocol.

Asonov [3] has a short survey of PIR results. Lin [49] has a survey of some of the information-theoretic results, complete with many examples. Castner [18] has a survey of some of the schemes based on number theory, complete with many examples. By the time you read this I will have a website of PIR papers at www.cs.umd.edu/˜gasarch. In addition I will have an extended version of this paper, with more proofs added, at www.eccc.uni-tier.de/eccc/ in 2004.

To limit the survey the following topics are omitted.

1. Locally Decodable Codes [27, 43, 44].

2. PIR's that are allowed to make errors but with low probability [43, 44].

3. Quantum PIR's [44].

4. Attempts to make PIR practical [4, 45] in the real-real world.

5. The connection between current PIR work and some of the older papers on the same theme such as [1, 7, 6].

Notation 1.1. Throughout this paper we assume that lg is $\log_2$ and returns an integer.

2 Definitions

The following definition is due to Chor et al. [23]. We present them informally.

Definition 2.1. [23] A *1-round k-DB Information Retrieval Scheme* with $x \in \{0,1\}^n$ and k databases has the following form.

1. Alice wants to know x_i. There are k copies of the database which all have $x = x_1 \cdots x_n$. The DB's do not communicate with each other.

2. Alice flips coins and, based on the coin flips and i, computes (query) strings $q_1, \ldots, q_k$. Alice sends q_j to database DB_j.

3. For all j, $1 \leq j \leq k$, DB_j sends back a (answer) string $ANS_j(q_j)$.

4. Using the value of i, the coin flips, and the $ANS_j(q_j)$, Alice computes x_i.

The *complexity* of the above PIR scheme is $\sum_{j=1}^{k} |q_j| + |ANS_j(q_j)|$.

We define two types of privacy.

Definition 2.2. [23] A *1-round k-DB Private Information Retrieval Scheme* with $x \in \{0,1\}^n$ and k databases is an information retrieval scheme such that, after the query is made and answered, the database does not have any information about what i is. The database is assumed to be computationally unbounded. Hence we need to ensure that the database does not have enough *information* to figure out anything about i. For these PIR schemes we will need multiple copies of the database.

Definition 2.3. [19] A *1-round k-DB Computationally Private Information Retrieval Scheme* with $x \in \{0,1\}^n$ and k databases is an information retrieval scheme such that, assuming some limitations on what the database can compute, after the query is made and answered, the database does not have any information about what i is. Hence we need to ensure that computing anything about i is beyond the computational limits of the database.

The definition is only for 1-round PIR schemes. This can be modified to allow more rounds; however, no PIR scheme in the literature needs more than 1-round. (Some variants of the PIR problem need multirounds- see Section 6.2.)

3 Information Theoretic PIR

Assume you have $k \geq 2$ copies of the database. Then there are PIR schemes of complexity $<< n$ which achieve complete information theoretic security. In this section we examine several of these PIR schemes. For a summary of the known results see Table 1. The last row of the table is the best known PIR scheme.

3.1 A k-DB, $O(kn^{1/\lg k})$-bit PIR Scheme

The PIR schemes in this section are from Chor et al. [23].

Definition 3.1. If σ is a string and $i \leq |\sigma|$ then $\sigma \oplus i$ is the string σ with the ith bit flipped.

Theorem 3.2. *[23] For all $k \in \mathsf{N}$ there is a k-DB, $O((k \lg k) n^{1/\lg k})$-bit PIR scheme.*

Tools	Th	Ref	2 DB	3 DB	4 DB	k DB
kth root	Th 3.2	[23]	no PIR	no PIR	$n^{1/2}$	$k \cdot n^{1/\lg k}$
Cov. Codes	Th 3.3	[23]	$n^{1/3}$	no PIR	$n^{1/4}$	$(k \lg k)n^{1/(\lg k + \lg\lg k)}$?
poly inter.		[22]	$n^{1/2}$	$n^{1/3}$	$n^{1/4}$	$(k^2 \log k)n^{1/k}$
Rec	Th 3.7	[2]	$n^{1/3}$	$n^{1/5}$	$n^{1/7}$	$2^{k^2} n^{1/(2k-1)}$
Linear Alg		[41]	$n^{1/3}$	$n^{1/5}$	$n^{1/7}$	$k! n^{1/(2k-1)}$
Linear Alg		[40, 10]	$n^{1/3}$	$n^{1/5}$	$n^{1/7}$	$k^3 n^{1/(2k-1)}$
poly-heavy		[12]	$n^{1/3}$	$n^{1/5.25}$	$n^{1/7.87}$	$n^{O(\lg\lg k / k \lg k)}$

Table 1: **Summary of Information Theoretic Schemes, up to a constant factor.**

KEY IDEA: View the database as a $\sqrt{n} \times \sqrt{n}$ bit array and use properties of $\oplus$.

Proof. We do the $k = 4$ case and leave the generalization to the reader. Each index of the database is represented as an ordered pair (i_1, i_2), where i_1 and i_2 are written in base $\lceil \sqrt{n} \rceil$. The databases are labeled DB_{00}, DB_{01}, DB_{10} and DB_{11}.

4-DB, $O(\sqrt{n})$-bit, Information Theoretic PIR Scheme

1. Alice wants to know bit x_{i_1,i_2}.
2. Alice generates $\sigma, \tau \in \{0,1\}^{\sqrt{n}}$.
3. Alice then generates two additional $\sqrt{n}$ bits strings from the first two strings: $\sigma' = \sigma \oplus i_1$ and $\tau' = \tau \oplus i_2$.
4. Alice sends two strings to each database. DB_{00} receives σ, τ. DB_{01} receives σ and τ'. DB_{10} receives σ' and τ. DB_{11} receives σ' and τ'.
5. D_{00} sends $\oplus_{\sigma(j_1)=1, \tau(j_2)=1} x_{j_1,j_2}$. D_{01} sends $\oplus_{\sigma(j_1)=1, \tau'(j_2)=1} x_{j_1,j_2}$. D_{10} sends $\oplus_{\sigma'(j_1)=1, \tau(j_2)=1} x_{j_1,j_2}$. D_{11} sends $\oplus_{\sigma'(j_1)=1, \tau'(j_2)=1} x_{j_1,j_2}$.
6. Alice XORs the four bits. Since x_{i_1,i_2} is the only bit that appeared an odd number of times, the result is x_{i_1,i_2}.

Note that the number of bits sent is $8\sqrt{n} + 4$. □

3.2 A k-DB, $O((k \lg k) n^{1/\lg k + \lg\lg k})$(?)-bit PIR Scheme

The PIR schemes in this section are from Chor et al. [23]. We show a 2-DB $O(n^{1/3})$-bit PIR scheme, which contains most of the ideas. We will then sketch a k-DB case, which is similar but whose bit complexity depends on open questions involving covering sets.

In the PIR scheme from Theorem 3.2, Alice sends many more bits than the database sends. By making the databases send a comparable number of bits as Alice, the total number of bits communicated between Alice and the databases can be reduced.

Theorem 3.3. *[23] There is a a 2-DB $O(n^{1/3})$-bit PIR scheme.*

KEY IDEA: Two databases can do the work of eight. Covering codes help to organize who does what.

Proof. By the $n = 8$ case of Theorem 3.2 there is an 8-DB $O(n^{1/3})$-bit PIR scheme. We can decrease the number of databases from eight to two by having two databases simulate the work of eight databases. In particular, DB_A simulates DB_{000}, DB_{001}, DB_{010}, DB_{100}; and DB_B simulates DB_{111}, DB_{011}, DB_{101} and DB_{110}. The simulation is designed so that DB_A (DB_B) simulates databases whose 3-bit labels are of *Hamming distance* ≤ 1 from 000 (111).

2-DB, $O(n^{1/3})$, Information-Theoretic PIR Scheme

1. Alice views the database as a $n^{1/3} \times n^{1/3} \times n^{1/3}$ grid. Alice wants x_{i_1,i_2,i_3}. The database is x.

2. Alice generates $\sigma, \tau, \eta \in \{0,1\}^{n^{1/3}}$ and creates $\sigma' = \sigma \oplus i_1$, $\tau' = \tau \oplus i_2$, $\eta' = \eta' \oplus i_3$.

3. Alice sends σ, τ, η to DB_A. Clearly DB_A can simulate DB_{000} (from the original PIR scheme) and send back the needed single bit. Consider what DB_A must do to simulate DB_{100}. DB_A knows that DB_{100} would have received the following strings: σ', τ, η. DB_A already has two of the three strings that DB_{100} has (namely, τ, η) but does not have σ'; however, it knows that σ' and σ (which it does have) differ by only one bit. DB_A can create and use all $n^{1/3}$ possible values of σ'. Specifically, DB_A generates $\sigma \oplus i$ for $0 \leq i < n^{1/3}$. Each of the strings generated is a candidate for σ'. For each candidate DB_A simulates what DB_{100} would have done. Note that there are $O(n^{1/3})$ candidates for σ' and each one leads to a 1-bit answer. Hence DB_A sends $O(n^{1/3})$ bits to simulate DB_{100}. Similarly, it can simulate DB_{010} and DB_{001}. The total number of bits sent back is $3n^{1/3} + 1$.

4. Alice sends σ', τ', η' to DB_B. Similar to the last step, DB_B simulates DB_{111}, DB_{110}, DB_{101}, and DB_{011}.

5. Alice XOR's the relevant bits. That is, she ignores all of the bits send back except those corresponding to $\{\sigma, \tau, \eta\}$, $\{\sigma, \tau, \eta'\}$, $\{\sigma, \tau', \eta\}$, $\{\sigma', \tau, \eta\}$, (which DB_A sends back) and $\{\sigma', \tau', \eta'\}$, $\{\sigma, \tau', \eta'\}$, $\{\sigma', \tau, \eta'\}$, $\{\sigma', \tau', \eta\}$, (which DB_B sends back).

Alice sends $6n^{1/3}$ bits and each database sends back $3n^{1/3}+1$ bits, for a total of $12n^{1/3}+2$ bits. $\square$

Note 3.4. Itoh [41] presents a slightly different PIR scheme yields $12n^{1/3}$. Beimel and Ishai [9, 10] use a different approach which yields $7.27n^{1/3}$. Improving this constant may be important in that the techniques employed may lead to a PIR scheme that uses $<< n^{1/3}$ bits.

The key to Theorem 3.3 is that we took an 8-DB, $O(n^{1/3})$-bit database and got two databases to do the work of eight since there are two vectors $\vec{v}_1, \vec{v}_2 \in \{0,1\}^3$ that cover $\{0,1\}^3$ in that every $\vec{v}$ is at most one bit away from either $\vec{v}_1$ or $\vec{v}_2$. This is called a *covering set*. >From Theorem 3.2 we know there is a 2^d-DB, $O(n^{1/d})$-bit PIR scheme. If we can find

k vectors that cover $\{0,1\}^d$ then we can generalize the PIR scheme from the above theorem. The problem is that the status of k, called the problem of covering numbers, is not resolved (see [24, 25, 37, 58]). Even so, we have the following theorem and speculation.

Theorem 3.5. *[23]*

1. *Assume there are k vectors in $\{0,1\}^d$ that cover $\{0,1\}^d$. Then there is a k-DB, $O(n^{1/d})$-bit PIR scheme.*

2. *There is a 4-DB, $O(n^{1/4})$-bit PIR scheme.*

3. *This technique can lead to, at best, a k-DB, $O(k \lg k)n^{1/(\lg k + \lg\lg k)})$-bit PIR scheme.*

Proof sketch:
1) This is similar to the proof of Theorem 3.3
2) This follows from part a using the vectors $\{0000, 1000, 0111, 1111\}$
3) This follows from the *volume bound* of Gallager [32], though it is not hard to prove. □

We tend to think that the lower bound on covering sets is equal to the upper bound; hence, we think there is a k-DB, $O(k \lg k)n^{1/(\lg k + \lg\lg k)})$-bit PIR scheme. However, this is unknown as of this time. It is also not important for PIR since, as we will see in the next section, there exist much better PIR schemes.

3.3 A k-DB $O(2^{k^2} n^{1/2k-1})$-bit PIR Scheme

The PIR scheme in this section is by Ambainis [2]. The main new idea is to use recursion; however, to set up the recursion we need a lopsided protocol.

Lemma 3.6. *[2] There is a 2-DB PIR scheme where the following hold.*

1. *Both databases receive $O(kn^{1/2k-1})$ bits.*

2. *One of the databases sends back $O(kn^{1/2k-1})$ bits.*

3. *The other database sends back $O(2^{2k} n^{2k-3/2k-1})$ bits.*

4. *Alice only needs $k+1 = \Theta(k)$ of the bits sent back by DB_A and $2^{2k-1} - k - 1 = \Theta(2^{2k})$ bits send back by DB_B.*

Proof. We will be simulating the 2^{2k-1}-DB $O(n^{1/2k-1})$-bit PIR Scheme of Theorem 3.2 with two databases. Hence we will be viewing the database as a $2k-1$-dimensional array of 0's and 1's. DB_A will simulate the $k+1$ databases that are of Hamming distance ≤ 1 from $00\cdots0$ DB_B will simulate the remaining $2^{2k-1} - k - 1$ databases.

2-DB, Lopsided Information-Theoretic PIR Scheme

1. Alice wants bit $x_{i_1,\ldots,i_{2k-1}}$.

2. Alice generates $\sigma_1, \sigma_2, \ldots, \sigma_{2k-1} \in \{0,1\}^{n^{1/2k-1}}$.

3. For $1 \le j \le 2k-$ Alice forms $\sigma'_j = \sigma_j \oplus i_j$.

4. Alice sends $\sigma_1, \ldots, \sigma_{2k-1}$ to DB_A. Alice sends $\sigma'_1, \ldots, \sigma'_{2k-1}$ to DB_B.

5. DB_A simulates $DB_{0\cdots0}$ and all databases of Hamming distance of one from $DB_{0\cdots0}$ (using the $n^{1/2k-1}$ PIR scheme described in Section 3.2). This takes $O(k \times n^{1/(2k-1)})$ bits (similar to the proof of Theorem 3.3). Alice only uses the $k+1$ bits corresponding to the correct guesses as to the queries that would have been asked.

6. DB_B simulates $DB_{1\cdots1}$ and all databases with index Hamming distance less than or equal to $2k-3$ from $1\cdots1$, which means that it simulates the rest of the databases that DB_A does not simulate. A database of Hamming distance h transmits $O(2^h n^{1/2k-1})$ bits. Hence DB_B will transmit $O(2^{2k-3} n^{(2k-3)/(2k-1)})$ bits back to Alice. Alice only uses the $2^{2k-3} - k - 1$ bits corresponding to the correct guesses as to the queries that would have been asked.

□

We use the lopsided PIR scheme to build a PIR scheme of the desired complexity.

Theorem 3.7. *[2] For all k there is a k-DB $O(2^{k^2} n^{1/2k-1})$-bit scheme.*

KEY IDEAS: The k databases simulate the two databases from the lopsides scheme. Since Alice only needs one bit of what the database is going to send her, apply the PIR scheme recursively to get that bit.

Proof. We build the PIR scheme by induction. The base case is the 2-DB $O(n^{1/3})$-bit PIR scheme from Theorem 3.3. Assume inductively that there is a $k-1$-DB, $O(2^{(k-1)^2} n^{1/(2k-3)})$-bit scheme.

k-DB, $O(2^{k^2} n^{1/2k-1})$ Information-Theoretic PIR Scheme

1. Alice has i and wants x_i.

2. Alice begins to simulate the lopsided protocol by generating $\sigma_1, \sigma_2, \ldots, \sigma_{2k-1}$ in $\{0,1\}^{n^{1/2k-1}}$ and forming for $1 \le j \le 2k-1$, $\sigma'_j = \sigma_j \oplus i_j$.

3. Alice sends $\sigma_1, \ldots, \sigma_{2k-1}$ to DB_1 (who will simulate DB_A) and $\sigma'_1, \ldots, \sigma'_{2k-1}$ to all of $DB_2, \ldots, DB_k$ (who will collectively simulate DB_B. $\sigma_1, \ldots, \sigma_{2k-1} \in$ to DB_1. (Alice sends a total of $O(k^2 n^{1/2k-1})$ bits.)

4. DB_1 runs the lopsided PIR scheme as DB_A and hence sends Alice $O(kn^{1/(2k-1)})$ bits.

5. Each of these databases $DB_2, \ldots, DB_k$ runs the lopsided PIR scheme playing the role of DB_B, and computes $O(2^{2k} n^{(2k-3)/(2k-1)})$ bits. These bits are not sent back to Alice, but are left at the database.

6. Alice and $DB_2, \ldots, DB_k$ treat the $O(2^{2k} n^{(2k-3)/(2k-1)})$ bits as a new database. Alice privately retrieves the $\Theta(2^{2k})$ bits from the new database, using the $(k-1)$-DB PIR scheme inductively. This takes

$$O(2^{2k} \times 2^{(k-1)^2} (n^{(2k-3)/(2k-1)})^{1/2k-3}) = O(2^{2k+k^2-2k+1} n^{1/2k-1}) = O(2^{k^2} n^{1/2k-1})$$

bits.

This PIR scheme may appear to take more than two rounds. But note that the bits Alice sends in each round do not depend on previous rounds; hence the PIR scheme can be done in one round. $\square$

Note 3.8. Note that the dependence on k is large since the PIR scheme takes $O(2^{k^2} n^{1/2k-1})$ bits. Itoh [41] has a different protocol that has constant $k!$. Ishai and Kushilevitz use an entirely different technique (without recursion) and reduce it to $O(k^3 n^{1/2k-1})$ bits. These improvements are important since the new techniques they used eventually lead to an PIR scheme using $<< n^{1/2k-1}$ bits.

3.4 A k-DB $n^{O(\lg\lg k/k \lg k)}$-bit PIR Scheme

The PIR scheme in this chapter is from Beimel et al. [12].

KEY IDEA: View the database as a polynomial.

Definition 3.9. Let $x \in \{0,1\}^n$. Let d, m be such that $\binom{m}{d} \geq n$. Hence $m \geq dn^{1/d}$.

1. For all $i \in [n]$, let $E(i)$ be the ith element of $\{0,1\}^m$ that has exactly d ones.
2. Let $P_x(z_1, \ldots, z_m)$ be the polynomial in Z_2 of degree d such that $(\forall i)[P_x(E(i)) = x_i]$. Formally $P_x(z_1, \ldots, z_m) = \sum_{i=1}^{n} x_i \prod_{E(i)_j=1} z_j$.
3. Let $(z_1, \ldots, z_m) = (\sum_{j=1}^{k} y_{1,j}, \ldots, \sum_{j=1}^{k} y_{m,j})$. Let

$$Q_x(\{y_{j,h}\}, 1 \leq j \leq k, 1 \leq h \leq m) = P_x(\sum_{j=1}^{k} y_{1,j}, \ldots, \sum_{j=1}^{k} y_{m,j}).$$

For each $j \in [k]$ let $V_j = \{y_{1,j}, \ldots, y_{m,j}\}$.

The PIR problem is equivalent to the following problem:

1. Alice has $E(i)$.
2. The k databases have P_x.
3. Alice wants to know $P_x(E(i))$ without the databases knowing anything about i.

We present the first few steps of a PIR scheme for this and then restate the problem.

Partial PIR Scheme

1. Alice has $E(i)$.

2. Alice generates $Y_1, \ldots, Y_{k-1} \in \{0,1\}^m$ and then forms Y_k such that $\sum_{j=1}^{k} Y_j = E(i)$.

3. For all $j \in [k]$, Alice sends $\{Y_1, \ldots, Y_k\} - Y_j$ to DB_j. Hence Alice sends $O(km)$ bits to each database, $O(k^2 m)$ bits total.

Each database has all but m variables of Q_x. Can they send Alice information so that she can evaluate $Q_x(\{y_{j,h}\}, 1 \le j \le k, 1 \le h \le m)$?

We use this reformulation to obtain a $O(k^3 n^{1/2k-1})$ PIR scheme. Alas, it is not the case that all of the key ideas are contained here. We will discuss how to modify it to obtain the desired PIR scheme.

Theorem 3.10. *[9, 40, 10, 12] For all k, there is a k-DB $O(k^3 n^{1/2k-1})$-bit scheme.*

Proof. Let $d = 2k - 1$. Let $m = \Theta(kn^{1/d})$ be such that $\binom{m}{d} \ge n$. Let $Q_x, V_1, \ldots, V_k$ be as in Definition 3.9. We assign to each monomial M of Q_x the database that can best evaluate it. This is done before the PIR scheme begins.

1. If there exists $j_0 \in [k]$ such that no variable of M is in V_{j_0} then assign M to DB_{j_0}. Note that DB_{j_0} will be able to evaluate M.

2. Assume for all $j \in [k]$ some variable of V_j is in M. If there exists $j_0 \in [k]$ such that only one variable of V_{j_0} is in M, then assign M to DB_{j_0}.

3. Assume for all $j \in [k]$ two variables of V_j are in M. Then M has $2k > d$ variables. Since Q_x is of degree d this cannot occur.

Let p_j be the sum of all the monomials assigned to DB_j. Note that once the $y_{j',h}$ for $j' \ne j$ are known, p_j is linear in $\{y_{j,1}, \ldots, y_{j,m}\}$. Hence p_j can be represented by an element of $\{0,1\}^m$, and is m bits long.

k-DB $O(k^3 n^{1/2k-1})$ PIR Scheme

1. Alice has $E(i)$.

2. Alice generates $Y_1, \ldots, Y_{k-1} \in \{0,1\}^m$ and then forms Y_k such that $\sum_{j=1}^{k} Y_j = E(i)$.

3. For all $j \in [k]$ Alice sends $\{Y_1, \ldots, Y_k\} - Y_j$ to DB_j. Hence Alice sends $O(k^2 m)$ bits.

4. For each $j \in [k]$ DB_j finds p_j and sends it back to Alice. Each database is sending m bits, so this is $O(km)$ bits total.

5. Alice can evaluate all of the p_j that are sent and XOR them. This is the answer.

This takes $O(k^2 m) = O(k^3 n^{1/d}) = O(k^3 n^{1/2k-1})$ bits. ∎

To extend this proof to general k we will need to take $d \ge 2k$. However, if $d \ge 2k$ and we assign monomials to the database best able to compute it, the polynomial that a DB has been assigned may be quadratic in m variables and hence requires $m^2 = n^{2d}$ bits to communicate. We sketch the ideas that are needed.

1. Let k', λ be parameters to be chosen carefully. The polynomial $P_x(\vec{z})$ is broken up into several pieces, some of which use $\vec{z}$ and some of which use the y's. In particular, for each $V \subseteq [k]$ such that $|V| \geq k'$ we have a polynomial $P_V(z_1, \ldots, z_m)$, and we have linear polynomials $p_j(y_{*,j})$ such that
   (a) $P_x(z_1, \ldots, z_m) = P_x(\sum_{j=1}^{k} y_{1,j}, \ldots, \sum_{j=1}^{k} y_{m,j}) =$
   $\sum_{V \subseteq [k], |V| \geq k'} P_V(z_1, \ldots, z_m) + \sum_{j=1}^{k} p_j(y_{1,j}, \ldots, y_{m,j})$.
   (b) P_V and p_j both take m variables.
   (c) The degree of P_V is $\leq \lambda|V|$.
   (d) The degree of p_j is one (so p_j is linear).
   (e) Note that $P_x(E(i)) = \sum_{V \subseteq [k], |V| \geq k'} P_V(E(i)) + \sum_{j=1}^{k} p_j(y_{1,j}, \ldots, y_{m,j})$.
2. For each $V \subseteq [k]$, $|V| = k'$, the databases in V will be able to evaluate $P_V(E(i))$. Alice cannot give them $E(i)$; however, it will turn out that there are not that many coefficients of P_V that Alice needs and she will be able to get these by a recursive call to the PIR scheme.

4 Conjectures that Imply sublinear PIR

In Section 3, we examined PIR schemes where the databases had unlimited computing power; hence we needed to replicate the database to achieve sublinear communication complexity. In this section we will look at sublinear PIR's where the database has computational limits.

4.1 Number Theoretic Conjectures

The Quadratic Residue Problem (see Definition 4.1) is thought to be hard. Kushilevitz and Ostrovsky [47] show that, assuming QR is hard, there is a 1-DB $O(n^{\epsilon})$-bit PIR scheme where ϵ can be taken to be arbitrarily small. We present that PIR scheme. Cachin et al. [17] assume that the Φ-*Hiding Problem* is hard and, from that, obtain a polylog PIR scheme. (The Φ-hiding problem is defined in| [17].) We do not formalize or prove that theorem.

Definition 4.1. Let $z, m \in \mathsf{N}$. Assume z is relatively prime to m. The number z is a *Quadratic Residue mod* m if there exists a number a such that $a^2 \equiv z \pmod{m}$. The *Quadratic Residue Problem* is, given (z, m), determine if z is a quadratic residue mod m.

We state the next theorem informally and only present the PIR scheme, not the proof that it is correct or private.

Definition 4.2. Z_n^* is the group of integers with underlying set $\{x \mid \text{gcd(x,n)=1}\}$ and the operation of multiplication mod n.

Theorem 4.3. *[47] Assume that the quadratic residue problem is 'hard' for m the product of two primes and $|m| \geq n^{\delta}$ ($|m|$ is the length of m, not its absolute value). Then there exists a 1-DB, $O(n^{1/2+\delta})$-bit PIR scheme.*

KEY IDEA: View the database as a $\sqrt{n} \times \sqrt{n}$ array. A new database is formed which relates to QR.

Proof sketch: The database is viewed as a $\sqrt{n} \times \sqrt{n}$ array of bits.

1-DB PIR Scheme

1. Alice wants bit $x_{i,j}$.
2. Alice generates two primes p_1, p_2 of the same length such that $m = p_1 p_2$ has length n^δ.
3. Alice generates $\sqrt{n}$ elements of Z_m^* which we call $r_1, \ldots, r_{\sqrt{n}}$. Alice makes sure that all of them are quadratic residues except r_i. Make sure that r_i has Jacobi symbol 1 (i.e., it is a non-square modulo both p_1 and p_2.)
4. Alice sends $m, r_1, \ldots, r_{\sqrt{n}}$ to the database. Note that this takes $O(n^\delta \sqrt{n}) = O(n^{1/2+\delta})$ bits.
5. The database computes the following matrix.
   (a) $c_{a,b} = z_b^2$ if $x_{ab} = 1$,
   (b) $c_{a,b} = z_b$ if $x_{ab} = 0$.
6. The database computes the products of the rows. In particular, for $1 \le a \le \sqrt{n}$ the database computes $r_a = \prod_{b=1}^{\sqrt{n}} c_{a,b}$.
7. The database sends over $r_1, \ldots, r_{\sqrt{n}}$. This takes $O(n^{1/2+\delta})$ bits.
8. Alice sees if r_j is a QR. If it is then $x_{i,j} = 1$, otherwise $x_{i,j} = 0$.

We leave the proof that this is correct to the reader. The proof that this is private depends on a careful definition of what it means for the QR problem to be hard. □

In the last step of the PIR scheme Alice receives $n^{1/2+\delta}$ bits but only uses n^δ of them. Hence we can do the last step recursively. In the PIR schemes current form this does not help; however, if we start with different dimensions and use the $n^{1/2+\delta}$ protocol as a base case we can obtain a PIR scheme which takes $n^{1/4+f(\delta)}$ bits. By repeating this we can obtain a PIR scheme with $n^{\epsilon+f_\epsilon(\delta)}$ bits. We leave this to the reader.

Note 4.4. The PIR scheme above uses that Z_m^* is a group. Yamamur and Saito have generalized this scheme to any group in [59]. Mann [50] has a similar scheme that is based on general assumptions.

4.2 One-way Functions Imply $O(n^\epsilon)$ 2-DB PIRs

Chor and Gilboa [19] show that if one-way functions exist then there is a 2-DB $O(n^\epsilon)$-bit PIR scheme. Having a one-way function is equivalent to having a pseudorandom generator. We phrase the theorem in those terms and prove a scaled down version of it.

Theorem 4.5. *[19] Let $1 \le m \le n$. Assume there is a function $G : \{0,1\}^{\lg n} \rightarrow \{0,1\}^{(n/m)^{1/3}}$ such that Alice and the databases can compute G but the databases cannot deduce anything about z from $G(z)$. Then there is a 2-DB $O((n/m)^{1/3} + m \lg n)$-bit PIR scheme. By taking $m = n^{1/4}$ we obtain an $O(n^{1/4} \lg n)$ PIR scheme.*

KEY IDEA: Alice does the $O(n^{1/3})$-bit PIR scheme from theorem 3.3 on each row, but she sends short seed instead of long message.

Proof. We view the database as an $m \times n/m$ bit matrix. We will determine m later.

2-DB $O((n/m)^{1/3} + m \lg n)$-bit PIR scheme

1. Alice wants bit $x_{i,j}$.
2. Alice generates $\sigma \in \{0,1\}^m$ and lets $\sigma' = \sigma \oplus i$.
3. Alice acts as though she is going to run the PIR scheme in Theorem 3.3 on the ith row to get the j bit. Alice prepares the queries q_1, q_2 of length $(n/m)^{1/3}$ that she would send to DB_1 and DB_2 but does not send them.
4. For each column index b, $1 \le b \le n/m$, Alice generates $s_b \in \{0,1\}^{\lg n}$.
5. Alice finds $M_1, M_2 \in \{0,1\}^{(n/m)^{1/3}}$ such that $G(s_j) \oplus M_1 = q_1$ and $G(s_j) \oplus M_2 = q_2$. (This is not a typo- it really is $G(s_j)$ both times.)
6. Alice sends to DB_1 the following: $\sigma, M_1, M_2, s_1, s_2, \ldots, s_m$. Alice sends to DB_2 the following: $\sigma', M_1, M_2, s_1, s_2, \ldots, s_m$. (The total is $O(m+(n/m)^{1/3}+m \lg n) = O((n/m)^{1/3}+ m \lg n)$.)
7. DB_1 sends back $U_1 = \oplus_{\sigma(a)=1} ANS_1(M_1 \oplus G(s_a))$ and $U_2 = \oplus_{\sigma(a)=0} ANS_2(M_1 \oplus G(s_a))$. This is of length $O((n/m)^{1/3})$. (Recall that $ANS_1(q)$ is the answer that DB_1 gives when sent question q.)
8. DB_2 sends back $V_1 = \oplus_{\sigma'(a)=1} ANS_2(M_1 \oplus G(s_a)$ and $V_2 = \oplus_{\sigma'(a)=0} ANS_2(M_1 \oplus G(s_a)$. This is of length $O((n/m)^{1/3})$.
9. Note that the PIR scheme in Theorem 3.3 that we are using has the following important property: if you give the two databases the same query, they will return the same answer. Hence we have, for all a, $ANS_1(M_1 \oplus G(s_a)) = ANS_2(M_1 \oplus G(s_a))$ and $ANS_1(M_2 \oplus G(s_a)) = ANS_2(M_2 \oplus G(s_a))$. Assume $\sigma(i) = 1$ (the other case is similar). Hence $U_1 \oplus V_1$ will mostly cancel out just leaving $ANS_1(M_1 \oplus G(s_j)) = ANS_1(q_j)$, and $U_2 \oplus V_2$ will mostly cancel out just leaving $ANS_2(M_2 \oplus G(s_j)) = ANS_2(q_j)$, >From these Alice can complete the simulation and recover x_i.

□

The following theorem follows from the proof of Theorem 4.5. (A result that follows from a Theorem is called a Corollary. A result that follows from a proof is called a Porism.)

Porism 4.6. Assume that $\alpha(n)$ is a function and $1 \le m \le n$. Assume that $\mathcal{P}$ is a 2-DB $\alpha(n)$-bit PIR scheme (possibly based on computational limits on the databases) where, for all queries q that that Alice could make, $ANS_1(q) = ANS_2(q)$. Assume there is a function $G : \{0,1\}^{\lg n} \to \{0,1\}^{\alpha(n/m)}$ such that Alice and the databases can compute G but the databases cannot deduce anything about z from $G(z)$. Then there is a 2-DB $O(\alpha(n/m) + m \lg n)$ PIR scheme (based on the same limits as the of the original scheme plus the limits about G). Applying this to the 2-DB $O(n^{1/4} \lg n)$-bit PIR scheme from Theorem 4.5, with $m = n^{1/5}$, yields a 2-DB $O(n^{1/5} \lg n)$-bit PIR scheme.

Corollary 4.7. *Assume that, for all $\delta < 1$, there exists $G : \{0,1\}^{n^\delta} \to \{0,1\}^n$ such that Alice and the databases can compute G but the databases cannot deduce anything about z from $G(z)$. Then, for all ϵ, there is a 2-DB, $O(n^\epsilon)$-bit PIR scheme.*

If 1-DB PIR's are desired then a stronger assumption is needed. In particular, the following are known:

1. Stern [57] and Mann [50] have shown that any homomorphic encryption scheme implies n^ϵ (any $\epsilon > 0$) 1-DB PIR schemes exist.
2. Kushilevitz and Ostrovsky [48] show that if there exists a One-way Trapdoor Permutation then there is an $n - o(n)$ 1-DB PIR scheme.
3. Certain assumptions about oblivious transfer imply 1-DB polylog-bits PIR schemes [46]. (Added at last minute- this Scheme has recently been broken. Details will appear

5 What Do 1-DB Sublinear PIRs Imply?

In this section we sketch a proof that 1-DB Sublinear PIR Implies OneWay Functions Exist and then summarize what else is known.

5.1 1-DB Sublinear PIR Implies OneWay Functions Exist

In Section 4 we show that one-way functions imply sublinear 2-DB PIR schemes exist. We also noted that some conjectures imply 1-DB sublinear PIR schemes exist. The question arises as to what primitives are necessary. Beimel et al. [11] show that if 1-DB sublinear PIR's exist then one-way functions exist. It is known that bit-commit (see [35]) implies one-way functions [38]. We sketch a weak version of 'sublinear 1-DB PIR's imply one-way' by showing the following.

Theorem 5.1. *[11] If there is a 1-DB $(n/2)$-bit PIR scheme then there is a weak bit-commitment scheme.*

Proof sketch:

Recall that $IP(x, y)$ is the inner product mod 2 of x and y.

We will have Carol committing to a bit and David be the one she commits to (we do not use Alice and Bob since Alice is being used in another capacity throughout this paper.)

Assume that there is 1-DB $(n/2)$-bit PIR scheme $\mathcal{P}$. We use it to build the following bit-commit scheme.
PHASE ONE: Carol commits to bit b.

1. Carol has bit b.
2. Carol generates $x, y \in \{0,1\}^n$. David generates $i \in [n]$.
3. Carol and David exercise the PIR scheme $\mathcal{P}$ with Carol having database x and David having index i. Note that at the end David knows x_i and Carol does not know i.
4. Carol sends David y and $\mathrm{IP}(x, y) \oplus b$.

Before giving phase two we claim that David cannot possibly deduce anything about b after Phase one. Assume that he could. Then the following is a communication protocol (no privacy involved) for the IP problem where Carol has x, David has y, and at the end of the protocol they both know something about $\mathrm{IP}(x, y)$. The protocol takes $n/2$ bits, which violates the lower bound on the randomized communication complexity of IP of Chor and Goldreich [21].

1. Carol has x, David has y.
2. David generates $i \in [n]$.
3. Carol and David exercise the PIR scheme $\mathcal{P}$ with Carol having database x and David having index i. Note that at the end David knows x_i and Carol does not know i.
4. David generates $c \in \{0,1\}$ (independent of everything else) and uses it as the bit sent from Carol at step 4 of the Commit protocol.
5. Using the above together with y, David outputs his prediction b' for b as we are assuming he can.
6. David computes $b' \oplus c$ as a prediction for $\mathrm{IP}(x, y)$ and transmits this prediction to Carol. (Since Carols choice of b in the commit protocol is uniformly distributed, Davids view here is identical to his view in the commit protocol. Hence c conveys just as much information as $b \oplus \mathrm{IP}(x, y)$ did.)

We now exhibit
PHASE TWO

1. Carol has $x, y \in \{0,1\}^n$ and $b \in \{0,1\}$. David has i, x_i, y, and $\mathrm{IP}(x, y) \oplus b$.
2. Carol sends David x.
3. David verifies that x_i is what it should be. (Carol did not know i so she must give David the correct x.)
4. David computes $\mathrm{IP}(x, y)$ and can then deduce b easily.

□

5.2 Summary of What is Known about Computational PIR

Notation 5.2.

1. *One-Way* means there exists a one-way function. This is known to be equivalent to the existence of pseudorandom generators.

2. *One-Way-Perm-Trap* means that there exists a one-way permutation with a trapdoor. Intuitively this means that if you know the trapdoor (e.g., the factors of a number) then you can compute the inverse.

3. *HES* means that there exists a homomorphic encryption scheme.

4. *OT* is oblivious transfer. It is known that 1-out-of-2 OT and 1-out-of-n OT are equivalent [26]. It is clear that 1-out-of-n sublinear OT and SPIR (see Section 7.2) are equivalent.

The following summarizes what is known about assumptions for sublinear 1-DB PIR.
One-Way-Perm-Trap $\Longrightarrow$ 1-DB $(n - o(n))$-bit PIR [48]
$(n - o(n))$-bit PIR $\Longrightarrow$ OT [29]
OT $\Longrightarrow$ One-Way
One-Way $\Longrightarrow$ 2-DB $o(n)$-bit PIR [19]

HES $\Longrightarrow$ 1-DB n^{ϵ}-bit PIR [50, 57]

Impagliazzo and Rudich [39] show that a proof that OT can be implemented using one-way functions only (without trapdoor), which does not relativize, would, roughly speaking, lead to a proof that $P \neq NP$ that does not relativize. They consider this evidence that proving such a result is going to be difficult. Since OT is equivalent to SPIR, and SPIR is close to PIR, it is unlikely that we can obtain sublinear PIR from one-way functions.

6 Retrieving Different Types of Data

6.1 PIR by Blocks

In the standard model Alice only wants one bit. It is more realistic that Alice wants a block of bits. What if the data is partitioned into blocks of m each and Alice wants an entire block. She could invoke a PIR scheme m times. Can she do better? This question was raised by Chor et al. [23].

Definition 6.1. [23] Let $\ell, n, k, n \in \mathsf{N}$. The $\mathcal{PIR}(\ell, n, k)$ problem is as follows: There are k databases each with the same $x \in \{0,1\}^n$. The x is broken up into n/ℓ blocks of ℓ each. Alice wants to privately retrieve ℓ consecutive bits. Note that $\mathcal{PIR}(\ell, n, k)$ problem can be solved by ℓ iterations of a k-DB PIR scheme.

The following theorem appeared in [23]. We give a different proof, from [42], which we will need in Section 8

Theorem 6.2. *[23, 42] The $\mathcal{PIR}(\ell, n, 2)$ problem can be solved with $O(n/\ell + \ell)$ bits.*

KEY IDEA: Use $\oplus$ on blocks

Proof. The database consists of n/ℓ blocks of ℓ bits each. View it as an n/ℓ by ℓ array. We denote the blocks $B_1, \ldots, B_{n/\ell}$.

$\mathcal{PIR}(\ell, n, 2)$ **Scheme**

1. Alice wants the ith row.

2. Alice generates $\sigma \in \{0,1\}^{n/\ell}$. Alice sets $\sigma' = \sigma \oplus i$. Alice sends σ to DB_1 and σ' to DB_2. (Alice sends $O(n/\ell + \ell)$ bits.)

3. DB_1 returns $\tau = \oplus_{\sigma(j)=1} B_j$. DB_2 returns $\eta = \oplus_{\sigma'(j)=1} B_j$. (The databases send $O(\ell)$ bits.

4. Alice computes $B_i = \tau \oplus \eta$.

The number of bits communicated is $O(n/\ell + \ell)$. □

Note 6.3. If $\ell = n^\delta$ then the above PIR scheme takes $O(n^{\max\{\delta, 1-\delta\}})$. Contrast this to using the $O(n^{1/3})$-bit PIR scheme (Theorem 3.3) n^δ times which results in a $O(n^{1/2+\delta})$-bit PIR scheme, which is clearly worse.

The proof can be generalized to obtain the following.

Theorem 6.4. *[23, 20]*

1. *For any constant $k \geq 2$, and for any ℓ, $\ell \geq n^{1/k-1}$, there exists an $O(\ell)$-bit $\mathcal{PIR}(\ell, n, k)$ scheme.*

2. *For any constant $k \geq 2$, and for any ℓ, there exists an $O(n^{1/2k-1}\ell^{k/2k-1})$-bit $\mathcal{PIR}(\ell, n, k)$ PIR scheme.*

6.2 PIR by Keyword

What if the database is a list of good stocks to buy and Alice just wants to know if BEATCS Inc. is a good stock? This does not fit our framework since she does not know exactly where in the database that information would be. This problem was considered by Chor and Gilboa [20].

Definition 6.5. [20] Let $\ell, N, k \in \mathsf{N}$. The PrivatE Retrieval by KeYwords problem with parameters, (henceforth $\mathcal{PERKY}(\ell, N, k)$) is as follows. There are k databases and they each have the same list of N strings of length ℓ. Alice has a string $w \in \{0,1\}^\ell$. Alice wants to determine if w is on the list without the databases knowing anything about w.

Theorem 6.6. *[20] There exists an $O((N+\ell)(\lg N))$-bit $\mathcal{PERKY}(\ell,N,k)$ scheme.;*

KEY IDEA: The words are sorted. Alice uses block PIR and binary search

Proof. The databases store the strings in lexicographic order. Both Alice and the database can view the set of strings as one string of length $N\ell$. Alice will first retrieve the middle string on the list using the PIR-block scheme of Theorem 6.2. (This takes $O(N+\ell)$ bits.) If the string is retrieved is lexicographically less than w then Alice knows that w is in the second half. If the string is retrieved is lexicographically more than w then Alice knows that w is in the second half. If the string is retrieved is w then Alice knows that w is in the list but cannot stop here or else the database will know what she was looking for (so she flips a coin to decide to go right or left). In all three cases Alice proceeds on either a real or fake binary search to determine if w is in the database. The entire process takes $O((N/\ell+\ell)(\lg N))$ bits. □

Note 6.7. Using perfect hash functions $\mathcal{PERKY}(\ell,N,k)$ can be solved in $O(N+\ell)$ bits [20].

7 Variants of PIR and CPIR

7.1 Robust PIR Schemes

In the standard PIR model the databases never break down (return no answer) and are never Byzantine (return a false answer). Beimel and Stahl [14] consider what can be done if some of the databases break down or return false answers.

Definition 7.1. [14] A *k-out-of-m PIR scheme* is an m-DB PIR scheme that works even if only k of the databases send back answers (the rest return nothing). Note that a standard k-DB PIR scheme is a k-out-of-k database PIR scheme. Note also that if there is a k-DB $b(n)$-bit PIR scheme then there is an easy $\binom{m}{k}b(n)$-bit k-out-of-m PIR scheme (have each k-sized subset of the m databases execute the original PIR scheme). Note also that the following 2-round solution works: in the first round send one bit to each DB and ask it to return that bit, which suffices to see which DB's are functioning. In this section we only consider 1-round solutions.

Theorem 7.2. *[14] If there is a 2-DB 1-round $b(n)$-bits PIR scheme then there is a 2-out-of-m databases $O(b(n)m\lg m)$-bit PIR scheme. Hence, using Theorem 3.3, there is a 2-out-of-m $O(n^{1/3}m\lg m)$-bit PIR scheme.*

Proof. We assume m is a power of 2. Number the databases DB_σ as $\sigma\in\{0,1\}^{\lg m}$.

1. Alice wants x_i.
2. Alice generates questions for two databases as though she is going to execute the 2-DB PIR scheme. Repeat this $(\lg m)-1$ times. Now Alice has $\lg m$ query pairs $(Q_j[0],Q_j[1])$ as $j=1,\ldots,\lg m$. Note that Alice has not sent anything yet.

3. For each σ Alice sends database DB_σ one query from the pair $(Q_j[0], Q_1[1])$ by sending
$$Q_1[\sigma(1)]Q_2[\sigma(2)] \cdots Q_{\lg m}[\sigma(\lg m)].$$
This takes $O(b(n)m \lg m)$ bits.

4. Each DB_σ sends back the answers it would send back to those queries. This takes $O(b(n)m \lg m)$ bits.

Note that, for all σ and j, database DB_σ does not get both a $Q_j[0]$ and $Q_j[1]$. Hence this is private. Also note that even if only two databases DB_σ and DB_τ respond, and if j is such that $\sigma(j) \neq \tau(j)$, then these two databases will give you some pair of queries $(Q_j[0], Q_1[1])$. This will suffice to find x_i. □

Note 7.3. There is an alternative proof of Theorem 7.2 that uses Shamir's secret sharing [56].

For the general case perfect hash families are used to obtain the following.

Theorem 7.4. *[14] There is a k-out-of-m* $2^{\tilde{O}(k)} n^{2 \lg\lg k / k \lg k} m \lg m$*-bit PIR scheme.*

We now look at the case where some databases can answer with the wrong information.

Definition 7.5. [14] Let $b, k, m \in \mathsf{N}$. A b-Byzantine k-out-of-m PIR scheme is an m-DB PIR scheme that works even if only k of the PIR schemes return answers and $\leq b$ of them return incorrect answers. (Note that the b bad databases do not collude.)

Theorem 7.6. *[14] There is a $k/3$-Byzantine robust k-out-of-m $O(kn^{1/(\lfloor k/3 \rfloor m \lg m)})$-bit PIR scheme.*

What if the b bad databases collude? In this case we will allow any b databases to collude and hence we can use the terminology of Section 7.4.

Theorem 7.7. *[14] Assume $b < k/3$. There is a b-private b-Byzantine k-out-of-m $O(\frac{k}{3b} n^{1/\lfloor (k-1)/3t \rfloor} m \lg m)$-bit PIR scheme.*

These last two theorems use polynomial interpolation.

7.2 Symmetric PIR Schemes

In the standard model Alice may end up learning more than the one bit she is curious about. Gertner et al. [34] considered the the question of preventing Alice from learning any more than x_i.

Definition 7.8. [34] A *Symmetric PIR* scheme (henceforth SPIR) is a PIR scheme where, at the end, Alice learns nothing more than x_i. We will allow the databases to share a common random string; however, the length of that string will be one of our parameters. There are two types of SPIR:

1. Those where Alice is honest-but-curious (she will follow the PIR scheme but will try to use the information gathered to find out more information).

2. Those where Alice is dishonest (she may choose to not follow the PIR scheme in order to find out some information).

We will need to look at the complexity of a PIR scheme slightly differently than usual to state the next theorem.

Definition 7.9. A *1-round $(\alpha(n), \beta(n))$-bit PIR scheme* is a PIR scheme where Alice sends a string of length $\alpha(n)$ and then receives, from each database, a string of length $\beta(n)$.

Theorem 7.10. *[34] Let $k \geq 2$. Assume there exists $\mathcal{P}$, a k-DB 1-round $(\alpha(n), \beta(n))$-bit PIR scheme. Then there exists a 1-round $(k+1)$-DB $(\alpha(n)+(k+1)\lceil \lg n \rceil, \beta(n)+1)$-bit SPIR scheme $\mathcal{P}'$ that uses a shared random string of length n. $\mathcal{P}'$ works in the honest-but-curious model. We obtain, using Theorem 3.3, a 3-DB $O(n^{1/3})$-bit SPIR scheme.*

Proof sketch:

We prove the $k = 2$ case; the extension is obvious. We do not include the proofs of security. The databases are DB_0, DB_1, DB_2 and all have $x \in \{0,1\}^n$ as well as a shared random string $r \in \{0,1\}^n$. It will turn out that DB_0 does not need r.

1. Alice has i. We take $i \in \{0, \ldots, n-1\}$ since we will be using mod n arithmetic. Alice sends queries to DB_1 and DB_2 as she would in PIR scheme $\mathcal{P}$ (this takes $2\alpha(n)$ bits). She then generates $\triangle \in \{0, \ldots, n-1\}$. Alice sends $\triangle$ to DB_1 and DB_2, and sends $i' \equiv i - \triangle \pmod{n}$ to DB_0 (this takes $\lg n$ bits).

2. DB_1 and DB_2 compute r' which is r shifted cyclically $\triangle$ places to the right. Then DB_1 and DB_2 compute $x' = x \oplus r'$. DB_1 and DB_2 answer the query Alice sent to them as if the database was x'. (This takes $\beta(n)$ bits.)

3. DB_0 sends $r_{i'}$. (This takes one bit.)

4. Alice reconstructs x'_i and then computes $x_i = x'_i \oplus r_{i'}$. (Note that $x'_i = x_i \oplus r_{i'}$ so $x'_i \oplus r_{i'} = x_i \oplus r_{i'} \oplus r_{i'} = x_i$.)

□

For the case where Alice is dishonest a new primitive is introduced called *Conditional Disclosure of Secrets* which is a generalization of t-out-of-m secrets sharing [56]. It is used to obtain the following results.

Theorem 7.11. *[34] Assume there exists $\mathcal{P}$, a 1-round k-DB $(\alpha(n), \beta(n))$-bit PIR scheme. Then there exists a 1-round $(k+1)$-DB $(\alpha(n)+(k+1)\lceil \lg n \rceil, 2\beta(n))$-bit SPIR scheme $\mathcal{P}'$ that uses a shared random string of length $O(n + \beta(n))$. $\mathcal{P}'$ and works when Alice is dishonest. We obtain, using Theorem 3.3, a 3-DB $O(n^{1/3})$-bit SPIR scheme.*

The above theorems are very general in that they take *any* PIR schemes and modify them to form a SPIR scheme. The next theorem is proven by taking a particular PIR scheme, the one from Theorem 3.7, and modifying it.

Theorem 7.12. *[34] For every constant $k \geq 2$ there exists a k-DB SPIR scheme with communication complexity and shared randomness $O(n^{1/2k-1})$ which works when Alice is dishonest.*

Theorem 7.13. *[34] There exists a $\lceil \lg n + 1 \rceil$ database $O(\lg^2 n \lg\lg n)$-bit SPIR scheme with communication complexity and shared randomness $O(\lg^2 n \lg\lg n)$ which works when Alice is dishonest.*

The notion of SPIR has also been looked at in the context of computational PIR by Mishra and Sarkar [51, 52]. Their main result assumes that both quadratic residue (see Definition 4.1) is hard, and the XOR assumption (to be defined below) is true. The XOR assumption was first articulated in [51]. They claim to have theoretical results and simulations as evidence for it.

Definition 7.14. [51, 52] The following is the XOR assumption. Let N be the product of two primes that are roughly the same length. Let x, y be picked from $\{0, \ldots, N\}$ at random. Let $z = x \oplus y$. Then

$$\begin{aligned}
\mathrm{Prob}(x \in \mathrm{QR} \wedge y \in \mathrm{QR} \mid z) &= 1/4 \\
\mathrm{Prob}(x \in \mathrm{QR} \wedge y \notin \mathrm{QR} \mid z) &= 1/4 \\
\mathrm{Prob}(x \notin \mathrm{QR} \wedge y \in \mathrm{QR} \mid z) &= 1/4 \\
\mathrm{Prob}(x \notin \mathrm{QR} \wedge y \notin \mathrm{QR} \mid z) &= 1/4
\end{aligned}$$

We state the following informally.

Theorem 7.15. *[51, 52] If the quadratic residue problem is hard and the XOR assumption is true then there is a 1-DB SPIR of complexity $O(n^\epsilon)$ where ϵ depends on the particular hardness assumption for quadratic residue problem. This scheme works when Alice is dishonest.*

The above Theorem follows more generally (and under weaker assumptions) from a general PIR to SPIR transformation by Naor and Pinkas [53]. This transformation takes any PIR scheme and, using a logarithmic number of oblivious transfers, turns it into a (computational) SPIR scheme. Since PIR implies OT, we get that in the computational setting PIR implies SPIR with no further assumptions and with a minor increase to the communication complexity. (NOTE- the above paragraph is quoted word for word from an email from Yuval Ishai.)

7.3 Information-Theoretic PIR without Replication

In the standard model there are several copies of the database, which may be a security risk. This problem was addressed by Gertner et al. [33]. Ideally we would like the databases themselves to not be able to (separately) deduce anything about x. Even more ideal– we want no t databases to be able to collude to find out anything about x.

Theorem 7.16. *[33] Assume there exists a k-DB $\alpha(n)$-bit PIR scheme. Assume further that the only queries that Alice asks are of the form "give me $\oplus_{a \in T} x_a$" Then there is a $(t+1)k$-DB $2\alpha(n)$-bit PIR scheme such that if any t databases collude they still cannot deduce anything about x.*

Proof sketch: We will do the $t = 2$ case; the generalization is obvious. The databases will be called $DB_1^1, DB_2^1, DB_1^2, DB_2^2, DB_1^3, DB_2^3, \ldots, DB_1^k, DB_2^k$. For $1 \le j \le k$ the database DB_1^j will have a random string $r_1^j \in \{0,1\}^n$ and DB_2^j will have r_2^j such that $r_1^j \oplus r_2^j = x$. Note that none of the databases have any information about x.

1. Alice has i.

2. Alice simulates the PIR scheme $\mathcal{P}$ as follows: if she wants to make the query $\oplus_{a \in T} x_a$ of database j, she makes it to both DB_1^j and DB_2^j. She gets back bits b_1 and b_2. The bit $b = b_1 \oplus b_2$ is the answer to her query. Note that this takes $2\alpha(n)$ bits.

□

The paper also considers the case where one of the databases has x but the others, even if they work together, cannot obtain any information about x. This is called "total independence"

Theorem 7.17. *[33] Assume there exists $\mathcal{P}$ k-DB $\alpha(n)$-bit PIR scheme. Then there is a $2k+1$-DB $\alpha(n)\lg n$-bit SPIR scheme such that one of the database has x and all the rest, even if they collude, cannot learn anything about x.*

7.4 t-private PIR Schemes

In the basic model we assumed that none of the databases talk to each other. Chor et al. raised the question of what happens if some of the databases talk to each other. A PIR scheme is t-private [22] if no subset of t of them can determine anything about i. Note that standard PIR schemes are 1-private.

Let $k, t \in \mathsf{N}$.

1. Chor et al. [22] show that there is a t-private, k-DB, $O(tn^{t/k})$ PIR scheme [22]. This paper uses polynomial interpolation.

2. Ishai and Kushilevitz [40, 10] have shown the following. Let d be such that

$$k = \min\left\{ \left\lfloor dt - \frac{d+t-3}{2} \right\rfloor, dt - t + 1 - (d \bmod 2) \right\}.$$

   Then there is a t-private, k-DB, $O(k^2\binom{k}{t}n^{1/d})$-bit PIR scheme. This paper uses linear algebra and secret sharing [56].

3. Beimel and Ishai [9, 10] show that there is a t-private, k-DB, $O(n^{1/\lfloor(2k-1)/t\rfloor})$-bit PIR scheme. This papers uses polynomials in a manner similar to that of Theorem 3.10, combined with secret sharing [56]. The technique can be seen as a precursor to the proof of Theorem 3.10.

4. Blundo et al [16] show that there is a t-private, k-DB, $O(k\sqrt{n})$-bit PIR scheme. This uses blocks of bits and XOR. The result is of interest when $t > k/2$.

7.5 PIR's with Preprocessing

In all of the PIR schemes discussed in the prior sections the database has to do $O(n)$ work, usually taking the XOR of n bits. Can the amount of work the database does be cut down? This question was raised and partially answered by Beimel et al. [13]. The key is that some XORs of blocks of bits are precomputed and prestored. This requires additional space. The following are known and are proven in [13].

Let $k \geq 2$ and $0 < \epsilon < 1$.

1. There is a k-DB, $O(k^3 n^{1/(2k-1)})$-bit PIR scheme where the databases does $O(n/\epsilon(\lg n)^{2k-2})$ work and use $O(n^{1+\epsilon})$ additional storage. This is a variant of the PIR scheme in [10, 40]. It is possible that a variant of Theorem 3.10 yields better results.

2. If there is a k-DB PIR scheme in which the length of the query sent to each database is α and the length of the answer of each database is β, then there is a k-DB PIR scheme with $k\beta$ work, $k(\alpha+\beta)$ communication, and $k2^\alpha$ extra bits to store.

3. There is a k-DB, $O(n^{1/(k+\epsilon)})$-bit PIR scheme where the databases do $O(n^{1/(k+\epsilon)})$ work and use $O(n^{O(1)})$ additional storage. This follows from Item 2 and a construction from [9, 10].

4. Suppose that homomorphic encryption exists. Then there exists a k-DB CPIR scheme with polynomially many extra bits, $O(n^\epsilon)$ communication, and $O(n^{1/k+\epsilon})$ work. This follows from Item 2 and a generalization of the PIR scheme from Theorem 4.3.

7.6 Commodity Based PIR

In the standard model of PIR there is a lot of communication between Alice and the databases. Beaver [5] began a line of research which aimed at minimizing direct communication between parties in cryptographic schemes. The main ideas was that a third party would be able to help facilitate the scheme but would not learn anything (e.g., the third party might just supply random bits to all parties). DiCrescenzo [28] applied this approach to PIR's.

In the results below the third party gives to Alice and the databases a random string. The length of that string is called the *Commodity Complexity*. We want the number of bits communicated between Alice and the databases to be low and we are willing to make the commodity complexity high to obtain that (though we also want to keep it low).

Theorem 7.18. *[5] Let $k \in \mathsf{N}$. There is a k-DB PIR scheme where (1) The bits sent between Alice and the databases is $O(\lg n)$, and (2) the commodity complexity is $O(n^{1/k-1})$.*

Theorem 7.19. *[5] Assume the Quadratic Residue problem is hard (see Definition 4.1). Let κ be a security parameter. There is a 1-DB PIR scheme where (1) The bits send between Alice and the databases is $O(\lg n + poly(\kappa))$, and (2) the commodity complexity is $O(\kappa n)$.*

8 Lower Bounds

Lower bounds on Private Information Retrieval Protocols have been hard to obtain. Lower bounds are (mostly) only known for 2-DB protocols with one round and restrictions on the number of bits returned by the database. Even then, prior to Kerenidis and de Wolf [44] all lower bounds had restrictions on the type of answers the database could return.

8.1 Lower Bounds For 2-DB 1-Round PIR Schemes

The following list summarizes lower bounds results for 2-DB 1-round PIR schemes.

1. Assume only linear answers are allowed. (That is, the answer is an XOR of some of the bits of the database). Goldreich et al. [36] show that if the database sends back a query of length a then Alice must send a query of length $\Omega(\frac{n}{2^a})$. This proof uses the equivalence between PIR's and locally decodable codes.

2. Assume only linear queries are allowed. Chor et al. [23] show that if the database sends back an answer of length one then each database must get a query of length at least $n-1$ bits. This matches an upper bound also in [23].

3. (No restrictions on the query.) Kerenidis and de Wolf [44] show that if Alice only uses a of the bits send back then Alice must send a query of length at least $\Omega(n/2^{6a})$. In the case $a = 1$ at least $(1 - H(11/14))n - 4 \sim 0.25n$ bits are required. Their proof first converts a 2-DB PIR scheme to a 1-DB quantum PIR scheme and then they show lower bounds on the quantum PIR scheme.

4. (No restrictions on the query.) Beigel et al. [8] show that if the database sends back a query of length one then Alice must send a query of length least $n-2$, which nearly matches an $n-1$ upper bound (upper bound in [23]). The lower bound proof avoids quantum techniques of [44]. Rather it builds on classical tools developed by Yao [60] and Fortnow and Szegedy [31] for studying locally-random reductions, a complexity-theoretic tool for information hiding that predates private information retrieval.

8.2 Other Lower Bounds

If privacy was not a concern, then Alice could obtain the bit she wants in $\lg n$ communication. Hence the next result, by Mann [50], is important in that it shows that privacy does increase the costs. It is also the only bound that holds for multi-round and one of the few bounds (the only other one is later in this section) that holds for k databases instead of just two.

Theorem 8.1. *[50] Let $k \geq 2$ and $\epsilon > 0$. Every k-DB $\alpha(n)$-bit PIR scheme where every database receives the same number of bits has $\alpha(n) \geq (\frac{k^2}{k-1} - \epsilon)\lg n$. In particular, taking $k = 2$ and $\epsilon = 1/2$, any 2-DB PIR scheme where every database receives the same number of bits has complexity at least $3.5 \lg n$.*

Itoh [42] proves lower bounds on certain types of PIR schemes.

Definition 8.2. Let $k, n \in \mathsf{N}$. Let $((q_1, \ldots, q_k), (ANS_1, \ldots, ANS_k), \phi)$ be a *k-DB 1-round r-random bit PIR for databases of size n with m-bit queries and a-bit answers.*

1. The PIR scheme is *linear* if, for all j, $1 \le j \le k$, the function ANS_j, viewed as a function from Z_2^m to Z_2^a is linear in each variable. That is, if $b_1, \ldots, b_{p-1}, b, c, b_{p+1}, \ldots, b_m \in \{0,1\}$ then

$$\begin{aligned} ANS_j(b_1, \ldots, b_{p-1}, b+c, b_{p+1}, \ldots, b_m) = \quad & ANS_j(b_1, \ldots, b_{p-1}, b, b_{p+1}, \ldots, b_m) \\ + \quad & ANS_j(b_1, \ldots, b_{p-1}, c, b_{p+1}, \ldots, b_m) \end{aligned}$$

2. Let $\ell \in \mathsf{N}$. The PIR scheme is *ℓ-multilinear* if, for all j, $1 \le j \le k$, the function ANS_j, viewed as a function from $((Z_2)^\ell)^{m/\ell}$ to Z_2^a is linear in each variable. That is, if $b_1, \ldots, b_{p-1}, b, c, b_{p+1}, \ldots, b_{m/\ell} \in \{0,1\}^\ell$ then

$$\begin{aligned} ANS_j(b_1, \ldots, b_{p-1}, b+c, b_{p+1}, \ldots, b_{m/\ell}) = \quad & ANS_j(b_1, \ldots, b_{p-1}, b, b_{p+1}, \ldots, b_{m/\ell}) \\ + \quad & ANS_j(b_1, \ldots, b_{p-1}, c, b_{p+1}, \ldots, b_{m/\ell}) \end{aligned}$$

3. Let $\ell \in \mathsf{N}$. The PIR scheme is *ℓ-affine* if, for all j, $1 \le j \le k$, the function ANS_j, viewed as a function from $((Z_2)^\ell)^{m/\ell}$ to Z_2^a is affine with constant 0^ℓ in each variable. That is, if $b_1, \ldots, b_{p-1}, b, c, b_{p+1}, \ldots, b_{m/\ell} \in \{0,1\}^\ell$ then

$$\begin{aligned} ANS_j(b_1, \ldots, b_{p-1}, b+c, b_{p+1}, \ldots, b_{m/\ell}) \; & = ANS_j(b_1, \ldots, b_{p-1}, b, b_{p+1}, \ldots, b_{m/\ell}) \\ & + ANS_j(b_1, \ldots, b_{p-1}, c, b_{p+1}, \ldots, b_{m/\ell}) \\ & + ANS_j(b_1, \ldots, b_{p-1}, 0^\ell, b_{p+1}, \ldots, b_{m/\ell}). \end{aligned}$$

Note 8.3. There is a k-DB $O(k^3 n^{1/k})$-bit PIR scheme from [40, 10, Section 3.2] is ℓ-multilinear with $\ell = (k-1)^2$. There is a k-DB $O(k^3 n^{1/2k-1})$-bit PIR scheme from [40, 10, Section 3.3] is ℓ-affine with $\ell = (2k-1)(k-1)$.

We prove a weak version of a theorem in [42] and then state several other theorems from [42].

The following is an information-theoretic argument that we leave to the reader.

Theorem 8.4. *[23] In a 1-DB PIR scheme the complexity is at least n.*

Theorem 8.5. *[42] Any k-DB linear PIR scheme has complexity at least* $\sqrt{\frac{n}{2k}}$.

Proof. Assume, by way of contradiction, that there is a k-DB linear PIR scheme
KEY IDEAS: Using linearity Alice can reconstruct the answers to any queries she wants. This enables her to obtain a 1-DB sublinear PIR scheme, which contradicts Theorem 8.4.
$((q_1, \ldots, q_k), (ANS_1, \ldots, ANS_k), \phi)$ with complexity $\sqrt{\frac{n}{2k}}$. We will assume that q_j returns a string of length m_j and that ANS_j returns a string of length a_j. We will use this to build a 1-DB PIR scheme of complexity $< n$, which contradicts Theorem 8.4.

1-DB PIR Scheme

1. Alice has i. The database has x. Alice generates ρ at random and forms the queries $q_1(i,\rho), q_2(i,\rho), \ldots, q_k(i,\rho)$. Alice does not send anything!

2. The database returns the following:

$$
\begin{array}{l}
ANS_1(x, 10^{m_1-1}),\ ANS_1(x, 010^{m_1-2}),\ ANS_1(x, 0010^{m_1-3}),\ \ldots,\ ANS_1(x, 0^{m_1-1}1),\\
ANS_2(x, 10^{m_2-1}),\ ANS_2(x, 010^{m_2-2}),\ ANS_2(x, 0010^{m_2-3}),\ \ldots,\ ANS_2(x, 0^{m_2-1}1),\\
\ldots\\
ANS_k(x, 10^{m_k-1}),\ ANS_k(x, 010^{m_k-2}),\ ANS_k(x, 0010^{m_k-3}),\ \ldots,\ ANS_k(x, 0^{m_k-1}1).
\end{array}
$$

3. (This is the real key.) Since the PIR scheme is linear Alice can, for every j, $1 \le j \le k$, deduce $ANS_j(x, q_j(i,\rho))$.

4. Alice can easily compute ϕ and hence x_i.

This PIR scheme sends a total of $m_1a_1+m_2a_2+\cdots+m_ka_k$ bits. Hence we are interested in the maximum value $\sum_{j=1}^k m_ja_j$ can take on. We know that $\sum_{j=1}^k (m_k+a_k) \le \sqrt{\frac{n}{2k}}$. One can show that the maximum value that $m_1a_1+m_2a_2+\cdots+m_ka_k$, given $\sum_{j=1}^k (m_k+a_k) \le \sqrt{\frac{n}{2k}}$, occurs when, for all j, $1 \le j \le k$, $a_j = m_j = \sqrt{\frac{n}{4k^2}}$. In this case we get
$m_1a_1+m_2a_2+\cdots+m_ka_k = k(n/4k) = n/4 < n$.
This is a contradiction. □

Theorem 8.5 is tight since Theorem 6.2 has a 2-DB $O(\sqrt{n})$-bit PIR linear scheme.
In Theorem 8.5 the k can be replaced by an $k-1$ by allowing the first query to be sent and answered. Using this, and generalizing the proof, one can prove the following.

Theorem 8.6. *[42] Let $k, \ell \in \mathsf{N}$. Let $\epsilon > 0$. Let $\mathcal{P}$ be any k-DB ℓ-multilinear PIR scheme. Let $\alpha(n)$ be its complexity. For almost all n, $\alpha(n) \ge (1/(k-1)^{1/\ell+1} - \epsilon)n^{1/\ell+1}$.*

Theorem 8.6 is not tight. There is an $(k-1)^2$- multilinear PIR scheme in [40, 10, Section 3.2] that has complexity $O(k^3n^{1/k})$. The lower bound implied by Theorem 8.6 is

$$\Big(\frac{1}{(k-1)^{1/(k-1)^2+1}} - \epsilon\Big)n^{1/(k-1)^2+1}.$$

The proof can be further generalized to show the following.

Theorem 8.7. *[42] Let $k, \ell \in \mathsf{N}$. Let $\epsilon > 0$. Let $\mathcal{P}$ be any k-DB ℓ-affine PIR scheme. Let $\alpha(n)$ be its complexity. For almost all n $\alpha(n) \ge (\frac{1}{(k-1)^{1/\ell+1}} - \epsilon)n^{1/\ell+1}$.*

Theorem 8.6 is not tight. There is an $(2k-1)(k-1)$- affine PIR scheme in [40, 10, Section 3.3] that has complexity $O(k^3n^{1/k})$. The lower bound implied by Theorem 8.6 is

$$\Big(\frac{1}{(k-1)^{1/(2k-1)(k-1)+1}} - \epsilon\Big)n^{1/(2k-1)(k-1)^2+1}.$$

9 Open Problems

1. Find a k-DB PIR scheme that uses less than $n^{O(\lg\lg k/k\lg k)}$ bits. The authors of [12] claim that their method, properly formalized, might yield a k-DB $n^{O(1/k^2)}$ scheme; however, it cannot be pushed further than that. Hence one plausible goal is to use their method (or others) to obtain a k-DB $n^{O(1/k^2)}$ scheme. We conjecture that this can be done.

2. The only lower bounds known are on fairly restrictive models. It is open to prove any bounds on an unrestricted model. We conjecture that $n^{\Theta(1/k^2)}$ is both an upper and lower bound.

3. All known PIR schemes are 1-round. We conjecture that if there is a k-DB, $n^{\alpha(k)}$-bit PIR scheme then there is a 1-round k-DB, $n^{O(\alpha(k))}$-bit PIR scheme. It may even be that there is a 1-round k-DB $n^{\alpha(k)}$-bit PIR.

4. What conjecture (e.g., the existence of 1-way functions) is equivalent to 1-DB $o(n)$-bit PIR? 1-DB $(n - o(n))$-bit PIR? 1-DB $(n - c)$-bit PIR? We conjecture that these questions do not have nice answers.

The biggest frustration about PIR's is the lack of good lower bounds. This is particularly striking since we are dealing with communication complexity where lower bounds are possible and plentiful (see [54]). We also note that hard results from Communication Complexity are not used that much in PIR (the only exception known to the author is Theorem 5.1 which uses randomized lower bounds for IP). Perhaps a more extensive use of these techniques would help; however many people work in both fields so it's not as though those results are unknown to the researchers.

10 Commentary

I have been asked "Having read 27 papers on PIR what do you think of the field?" Well the word 'read' may be overly generous; however, I do have the following impressions:

(1) Some of the results are simple enough to present in an undergraduate cryptography class. I have taught Theorems 3.3 and 4.3. towards the end of such a course (after the mandatory material was covered) and it worked well.

(2) PIR is interesting in that it is a simple model and yet proving things about it seems to require knowing material from other fields. Communication Complexity, Computational Number Theory, Complexity Theory, Cryptography, Combinatorics, all play a role. Hence a course on it would be an excellent and motivated way to get into these other subjects.

(3) How interesting is PIR? A field is interesting if it answers a fundamental question, or connects to other fields that are interesting, or uses techniques of interest. While I don't see PIR as being fundamental, I do see it as both connecting to fields of interest and using interesting techniques.

11 Acknowledgments

I would like to thank Clyde Kruskal and Charles Lin for proofreading. I would also like to thank Eyal Kushilevitz for useful email exchanges and for writing a constant fraction of the papers on PIR. In addition I would like to thank Amos Beimel, Yuval Ishai, and Eyal Kusilevitz, for proofreading and catching some subtle errors.

References

[1] M. Abadi, J. Feigenbaum, and J. Killian. On hiding information from an oracle. *Journal of Computer and Systems Sciences*, 39, 1989.

[2] A. Ambainis. Upper bound on the communication complexity of private information retrieval. In *Proc. of the 24th ICALP*, 1997.

[3] D. Asonov. Private information retrieval. In *GI Jahrestagung (2)*, pages 889–894, 2001.

[4] D. Asonov and J.-C. Freytag. Almost optimal private information retrieval. In *2nd Workshop on Privacy Enhancing Technologies (PET2002)*, 2002.

[5] D. Beaver. Commodity-based cryptography. In *Proc. of the 29th ACM Sym. on Theory of Computing*, pages 446–455, 1997.

[6] D. Beaver and J. Feigenbaum. Hiding instances in mulit-oracle queries. In *Proc. of the 7th Sym. on Theoretical Aspects of Computer Science*, volume 415 of *Lecture Notes in Computer Science*, pages 37–48, Berlin, 1990. Springer-Verlag.

[7] D. Beaver, J. Feigenbaum, J. Kilian, and P. Rogaway. Locally random reductions: improvements and applications. *Journal of Cryptology*, 10, 1997. Earlier version in 1990 CRYPTO.

[8] R. Beigel, L. Fortnow, and W. Gasarch. A nearly tight lower bound for private information retrieval protocols. Technical Report TR03-087, Electronic Colloquim on Computational Complexity (ECCC), 2003. See www.ecc.uni-trier.de/eccc/.

[9] A. Beimel and Y. Ishai. Information retrieval private information retrieval: A unified construction. In *ICALP01*, 2001. Also in ECCCTR, 2001. Also available at springer verlag website www.springerlink.com if you or your school is registered. Also part of the paper *General constructions for information-theoretic private information retrieval* by Beimel, Ishai, and Kushilevitz.

[10] A. Beimel, Y. Ishai, and E. Kushilevitz. General constructions for information-theoretic private information retrieval, 2003. Unpublished manuscripte available at www.cs.bgu.ac.il/ beimel/pub.html.

[11] A. Beimel, Y. Ishai, E. Kushilevitz, and T. Malkin. One-way functions are essential for single-server private information retrieval. In *Proc. of the 31th ACM Sym. on Theory of Computing*, 1999.

[12] A. Beimel, Y. Ishai, E. Kushilevitz, and J.-F. Rayomnd. Breaking the $o(n^{1/(2k-1)})$ barrier for information-theoretic private information retrieval. In *Proc. of the 43st IEEE Sym. on Found. of Comp. Sci.*, 2002.

[13] A. Beimel, Y. Ishai, and T. Malkin. Reducing the servers' computation in private information retrieval: Pir with preprocessing. *Journal of Cryptology*. To appear. Prelim version was in CRYPTO00.

[14] A. Beimel and Y. Stahl. Robust information-theoretic private information retrieval. In *Proceedings of the 3rd conference on security in Communications networks*, pages 326–341, 2002.

[15] G. Blakely and C. Meadows. A database encryption scheme which allows computation of statistics using encrypted data. In *Proceedings of the Symposium on Security and Privacy*, pages 116–122, 1985.

[16] C. Blundo, P. DArco, and A. DeSantis. A t-private k-database information retrieval scheme. *International Journal of Information Security*, 1(1):64–68, 2001.

[17] C. Cachin, S. Micali, and M. Stadler. Computationally private information retrieval with polylog communication. In *EUROCRYPT99*, 1999.

[18] A. Castner. Survey of single database private information retrieval systems, 2002. Available at www.cs.umd.edu/~gasarch/papers under Undergraduate Students.

[19] B. Chor and N. Gilboa. Computationally private information retrieval. In *Proc. of the 32th ACM Sym. on Theory of Computing*, 2000.

[20] B. Chor, N. Gilboa, and M. Naor. Private information retrieval by keywords, 1998. Unpublished manuscripte available at www.cs.technion.ac.il/~gilboa.

[21] B. Chor and O. Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal of Computing*, 17, 1988. Prior version in *I*EEE Sym on Found. of Comp. Sci., 1985 (FOCS).

[22] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan. Private information retrieval. In *Proc. of the 36st IEEE Sym. on Found. of Comp. Sci.*, 1995. We are using the conference version since the item referenced did not appear in the journal version.

[23] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan. Private information retrieval. *Journal of the ACM*, 45, 1998. Earlier version in FOCS 95.

[24] G. Cohen, M. Karpovsky, and H. Mattson. Covering radius — survey and recent results. *IEEE Trans. Inform. Theory*, IT-31:338–343, 1985.

[25] G. Cohen, A. Lobstein, and N. Sloane. Further results on the covering radius of codes. *IEEE Trans. Inform. Theory*, IT-32:680–694, 1986.

[26] C. Crepeau. Equivalent between two flavors of oblivious transfers. In *Proc. of the 8th IACR Conf. on Crypto (LNCS Vol 403)*, 1988.

[27] Deshpande, Jain, Kavitha, Lokam, and Radhakrishnan. Better lower bounds for locally decodable codes. In *Proc. of the 17th IEEE Conf on Complexity Theory*. IEEE Computer Society Press, 2002.

[28] G. DiCrescenzo, Y. Ishai, and R. Ostrovsky. Universal service-providers for private information retrieval. *Journal of Cryptology*, 14(1), 2001. Earlier Version in PODS 1998.

[29] G. DiCrescenzo, T. Malkin, and R. Ostrovsky. Single database private information retrieval implies oblivious transfer. In *EUROCRYPT00*, 2000.

[30] J. Feigenbaum. Encrypting problem instances, or, ... can you take advantage of someone without having to trust him? In *Proc. of the 5th IACR Conf. on Crypto (LNCS Vol 218)*, pages 477–488, 1985.

[31] L. Fortnow and M. Szegedy. On the power of two-local random reductions. *Information Processing Letters*, 44:303–306, 1992.

[32] P. E. Gallager. *Information Theory and Reliable Communication.* Wiley, New York, 1968.

[33] Y. Gertner, S. Goldwasser, and T. Malkin. A random server model for private information retrieval or information theoretic pir avoiding database replication. In *Proc. of the 2nd RANDOM*, 1998.

[34] Y. Gertner, Y. Ishai, E. Kushilevitz, and T. Malkin. Protecting data privacy in private information retrieval schemes. *Journal of Computer and Systems Sciences*, 60, 2000. Prelimary version in STOC98.

[35] O. Goldreich. *Foundations of Cryptography: Basic Tools.* Cambridge University Press, 2001. Fragments of this are at the Electronic Colloquim on Computational Complexity, 1995, under monographs.

[36] O. Goldreich, H. Karloff, L. Schulman, and L. Trevisan. Lower bounds for linear local decodable codes and private information retrieval systems. In *Proc. of the 17th IEEE Conf on Complexity Theory.* IEEE Computer Society Press, 2002.

[37] I. Honkala. Modified bounds for covering codes. *IEEE Trans. Inform. Theory*, IT-37:351–365, 1991.

[38] R. Impagliazo and M. Luby. One-way functions are essential for cryptography. In *Proc. of the 30th IEEE Sym. on Found. of Comp. Sci.*, pages 230–235. IEEE Computer Society Press, 1989.

[39] R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. In *Proc. of the 21th ACM Sym. on Theory of Computing*, pages 44–61, 1989.

[40] Y. Ishai and E. Kushilevitz. Improved upper bounds on information-theoretic private information retrieval. In *Proc. of the 31th ACM Sym. on Theory of Computing*, 1999. Part of the paper *General constructions for information-theoretic private information retrieval* by Beimel, Ishai, and Kushilevitz.

[41] T. Itoh. Efficient private information retrieval. *IEICE Trans. Fundamentals*, ES2-A(1), 1999.

[42] T. Itoh. On lower bounds for the communication complexity of private information retrieval. *IEICE Trans. Fundamentals*, ES4-A(1), 2001. This journal is at http://search.ieice.org/2001/index.htm.

[43] J. Katz and L. Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *Proc. of the 32th ACM Sym. on Theory of Computing*, 2000.

[44] I. Kerenidis and R. de Wolf. Exponential lower bound for 2-query locally decodable codes. In *Proc. of the 35th ACM Sym. on Theory of Computing*, pages 106–115, 2003.

[45] D. Kesdogan, M. Borning, and M. Schmeink. Unobservable surfind on the world wide web: is private information retrieval an alternative to the MIX based approach? In *2nd Workshop on Privacy Enhancing Technologies (PET2002)*, 2002.

[46] A. Kiayias and M. Yung. Secure games with polynomial expressions. In *ICALP01*, 2001.

[47] E. Kushilevitz and R. Ostrovsky. Replication is not needed: Single database, computationally-private information retrieval (extended abstract). In *Proc. of the 38st IEEE Sym. on Found. of Comp. Sci.*, pages 364–373, 1997.

[48] E. Kushilevitz and R. Ostrovsky. One-way trapdoor permutations are sufficient for non-trivial single-server private information retrieval. In *EUROCRYPT00*, 2000.

[49] C. Lin. Survey of private information retrieval systems, 2001. Available at www.cs.umd.edu/~gasarch/papers under Masters Students.

[50] E. Mann. *Private access to distributed information.* PhD thesis, Technion – Israel Institute of Technology, Haifa, 1998. Masters Thesis.

[51] S. Mishra. *Symmetrically Private Information Retrieval.* PhD thesis, Indian Statistical Institute, Calcutta, 2000. Available at citeseer.nj.nec.com/kumarmishra00symmetrically.html.

[52] S. Mishra and P. Sarkar. Symmetrically private information retrieval. In *Proc. of the 1st INDOCRYPT (LNCS 1977)*, 2000.

[53] M. Naor and B. Pinkas. Oblivious transfer and polynomial evaluation. *Proc. of the 31th ACM Sym. on Theory of Computing*, 1999.

[54] N. Nisan and A. Wigderson. Hardness vs randomness. *Journal of Computer and Systems Sciences*, 49, 1994. Prior version in *I*EEE Sym on Found. of Comp. Sci., 1988 (FOCS).

[55] R. Rivest, L. Adelman, and M. Dertouzos. On databanks and privacy homomorphism. In D. et al, editor, *Foundations of secure computation*, pages 168–177, 1978.

[56] A. Shamir. How to share a secret. *Communications of the ACM*, 22, 1979.

[57] J. Stern. A new and efficient all-or-nothing disclosure of secrets protocol. In *Asia Crypt 1998 (LNCS 1514))*, pages 357–371, 1998.

[58] G. V. Wee. Improved sphere bounds on the covering radius of codes. *IEEE Trans. Inform. Theory*, IT-34:237–245, 1988.

[59] A. Yamamura and T. Saito. Private information retrieval based on subgroup membership problem. In *Proc. of the 6th Australasian Conf., ACISP 2001*, 2001.

[60] A. Yao. An application of communication complexity to cryptography, 1990. Lecture DIMACS Workshop on Structural Complexity and Cryptography.