Information-Theoretic Private Information Retrieval: A Unified Construction
(Extended Abstract)

Amos Beimel[1] and Yuval Ishai[2]

[1] Ben-Gurion University, Israel. beimel@cs.bgu.ac.il.
[2] DIMACS and AT&T Labs – Research, USA. yuval@dimacs.rutgers.edu.

Abstract. A Private Information Retrieval (PIR) protocol enables a user to retrieve a data item from a database while hiding the identity of the item being retrieved. In a *t-private*, *k-server* PIR protocol the database is replicated among k servers, and the user's privacy is protected from any collusion of up to t servers. The main cost-measure of such protocols is the *communication complexity* of retrieving a single bit of data.
This work addresses the *information-theoretic* setting for PIR, in which the user's privacy should be unconditionally protected from collusions of servers. We present a unified general construction, whose abstract components can be instantiated to yield both old and new families of PIR protocols. A main ingredient in the new protocols is a generalization of a solution by Babai, Kimmel, and Lokam to a communication complexity problem in the so-called *simultaneous messages* model.
Our construction strictly improves upon previous constructions and resolves some previous anomalies. In particular, we obtain: (1) t-private k-server PIR protocols with $O(n^{1/\lfloor(2k-1)/t\rfloor})$ communication bits, where n is the database size. For $t > 1$, this is a substantial asymptotic improvement over the previous state of the art; (2) a constant-factor improvement in the communication complexity of 1-private PIR, providing the first improvement to the 2-server case since PIR protocols were introduced; (3) efficient PIR protocols with logarithmic query length. The latter protocols have applications to the construction of efficient families of *locally decodable codes* over large alphabets and to PIR protocols with reduced work by the servers.

1 Introduction

A Private Information Retrieval (PIR) protocol allows a user to retrieve a data item of its choice from a database, such that the server storing the database does not gain information on the identity of the item being retrieved. For example, an investor might want to know the value of a specific stock without revealing which stock she is interested in. The problem was introduced by Chor, Goldreich, Kushilevitz, and Sudan [11], and has since then attracted a considerable amount of attention. In formalizing the problem, it is convenient to model the database by an n-bit string x, where the user, holding some *retrieval index* i, wishes to learn the i-th data bit x_i.

A trivial solution to the PIR problem is to send the entire database x the user. However, while being perfectly private, the *communication complexity* of this solution may be prohibitively large. Note that if the privacy constraint is lifted, the (non-private) retrieval problem can be solved with only $\lceil \log_2 n \rceil + 1$ bits of communication. Thus, the most significant goal of PIR-related research has been to minimize the communication overhead imposed by the privacy constraint. Unfortunately, if the server is not allowed to gain *any* information about the identity of the retrieved bit, then the linear communication complexity of the trivial solution is optimal [11]. To overcome this problem, Chor et al. [11] suggested that the user accesses k replicated copies of the database kept on different servers, requiring that each *individual* server gets absolutely no information on i. PIR in this setting is referred to as *information-theoretic* PIR.[1] This naturally generalizes to t-*private* PIR, which keeps i private from any collusion of t servers.

The best 1-private PIR protocols known to date are summarized below: (1) a 2-server protocol with communication complexity of $O(n^{1/3})$ bits [11]; (2) a k-server protocol with $O(n^{1/(2k-1)})$ communication bits, for any constant k (Ambainis [1] improving on [11], see also Ishai and Kushilevitz [15]); and (3) an $O(\log n)$-server protocol with $O(\log^2 n \log\log n)$ communication bits ([11], and implicitly in Beaver and Feigenbaum [4]). For the more general case of t-private PIR, the best previous bounds were obtained in [15], improving on [11]. To present these bounds, it is convenient to use the following alternative formulation of the question:

> Given positive integers d and t, what is the least number of servers for which there exists a t-private PIR protocol with communication complexity $O(n^{1/d})$?

In [15] it was shown that $k = \min\left(\lfloor dt - (d+t-3)/2 \rfloor\ ,\ dt - t + 1 - (d \bmod 2)\right)$ servers are sufficient. If t is fixed and d grows, the number of servers in this bound is roughly $(t - \frac{1}{2})d$.

No strong general lower bounds on PIR are known. Mann [20] obtained a constant-factor improvement over the trivial $\log_2 n$ bound, for any constant k. In the 2-server case, much stronger lower bounds can be shown under the restriction that the user reconstructs x_i by computing the exclusive-or of a *constant* number of bits sent by the servers, whose identity may depend on i (Karloff and Schulman [17]). These results still leave an exponential gap between known upper bounds and lower bounds. For a list of other PIR-related works the reader can consult, e.g., [7].

A different approach for reducing the communication complexity of PIR is to settle for *computational* privacy, i.e., privacy against computationally bounded servers. Following a 2-server solution by Chor and Gilboa [10], Kushilevitz and Ostrovsky [19] showed that in this setting a *single* server suffices for obtaining sublinear communication, assuming a standard number-theoretic intractability assumption. The most communication efficient single-server PIR protocol to date is due to Cachin, Micali, and Stadler [9]; its security is based on a new number-theoretic intractability assumption, and its communication complexity is polynomial in $\log n$ and the security parameter. From a practical point of view, single-server protocols have obvious advantages over multi-server protocols.[2] However, they have some *inherent* limitations which can only be avoided in a multi-server setting. For instance, it is impossible for a (sublinear-communication) single-server PIR protocol to have very short queries (say, $O(\log n)$-bits long) sent from the user to the server, or very short answers (say, one bit long) sent in return. These two extreme types of PIR protocols, which can be realized in the

[1] The term "information-theoretic PIR" may also refer to protocols which leak a negligible amount of information on i. However, there is no evidence that such a relaxation is useful.

[2] However, for practical sizes of databases and security parameter, known multi-server (and even 2-server) protocols are much more efficient in computation and are typically even more communication-efficient than single-server protocols.

information-theoretic setting, have found different applications (Di-Crescenzo, Ishai, and Ostrovsky [12], Beimel, Ishai, and Malkin [7]) and therefore serve as an additional motivation for information-theoretic PIR. A different, coding-related, motivation is discussed below.

Our results. We present a unified general framework for the construction of PIR protocols, whose abstract components can be instantiated to meet or beat all previously known upper bounds. In particular we obtain:

- t-private k-server PIR protocols with communication complexity $O(n^{1/\lfloor(2k-1)/t\rfloor})$. In other words, $k > dt/2$ is sufficient for the existence of a t-private k-server PIR protocol with $O(n^{1/d})$ communication. For $t > 1$, this is a substantial asymptotic improvement over the previous state of the art [15]. For example, for $t = 2$ the communication complexity of our protocol is $O(n^{1/(k-1)})$, while the communication complexity of the best previous protocol [15] is $O(n^{1/\lfloor 2k/3\rfloor})$. Our bound is essentially the best one could hope for without asymptotically improving the bounds for the 1-private case.
- A constant-factor improvement in the communication complexity compared to the 2-server protocol of [11] and its 1-private k-server generalizations from [1, 15]. In the 2-server case, this provides the first improvement since the problem was introduced in [11].
- Efficient PIR protocols with logarithmic query length: We construct a t-private k-server PIR protocol with $O(\log n)$ query bits and $O(n^{t/k+\epsilon})$ answer bits, for every constant $\epsilon > 0$. The 1-private protocols from this family were used in [7] to save computation in PIR via preprocessing, and have interesting applications, discussed below, to the construction of efficient *locally decodable codes* over large alphabets.

It is interesting to note that in contrast to previous PIR protocols, in which the user can recover x_i by reading only a *constant* number of answer bits (whose location depends only on i), most instances of our construction require the user to read *all* answer bits and remember either the queries or the randomness used to generate them. It is open whether the previous constructions of [15] (in particular, the t-private protocols for $t > 1$) can be improved if one insists on the above "easy reconstruction" feature, which allows the user's algorithm to be implemented using logarithmic space.

Locally decodable codes. Information-theoretic PIR protocols have found a different flavor of application, to the construction of *locally decodable codes*. A locally decodable code allows to encode a database x into a string y over an alphabet Σ, such that even if a large fraction of y is adversarially corrupted, each bit of x can still be decoded *with high probability* by probing *few* (randomly selected) locations in y. More formally, a code $C : \{0,1\}^n \to \Sigma^m$ is said to be (k, δ, ρ)-locally decodable, if every bit x_i of x can be decoded from $y = C(x)$ with success probability $1/2+\rho$ by probing k entries of y, even if up to a δ-fraction of the m entries of y are corrupted. Katz and Trevisan [18] have shown an intimate relation between such codes and information-theoretic PIR. In particular, any information-theoretic PIR protocol can be converted into a locally decodable code with related efficiency by concatenating the answers of all servers on all possible queries. This motivates the construction of PIR protocols with short queries.

The short-query instantiations of our PIR construction have an interesting interpretation in terms of locally decodable codes. The main focus in the works [18, 14] has been on the following question. Suppose that ρ, δ are restricted to be greater than some positive constant. Given a constant number of queries k and a *constant-size* (say, binary) alphabet Σ, what is the minimal asymptotic growth of the code length? Generalizing a PIR lower bound of [20], it is proved in [18] that for any constant k the code length must be super-linear. For the case of a linear code with $k = 2$ (non-adaptive) queries, an exponential lower bound on $m(n)$ has been obtained by Goldreich and Trevisan [14]. While no super-polynomial lower bounds are known for the case $k > 2$, the best known upper bound (obtained from PIR protocols with a single answer bit per server, see Section 6) is $m(n) = 2^{O(n^{1/(k-1)})}$, which is exponential in n. Our construction answers the following dual question: Suppose that we insist on the code being *efficient*, namely of polynomial length. Then, how small can the alphabet Σ be? More precisely, given a constant k, how small can $\Sigma_k(n)$ be such that the code length $m(n)$ is polynomial and, as before, $\rho(n), \delta(n)$ are kept constant? The short-query variants of our construction imply the following upper bound: for any constants $k \geq 2$ and $\epsilon > 0$ it suffices to let $\Sigma_k(n) = \{0,1\}^{\beta(n)}$, where $\beta(n) = O(n^{1/k+\epsilon})$.

ORGANIZATION. In Section 2 we give an overview of our unified approach for constructing PIR protocols. In Section 3 we provide some necessary definitions. In Section 4 we describe a meta-construction of PIR protocols, in Section 5 we instantiate one of its crucial ingredients, and in Section 6 we derive new and old families of PIR protocols as instances of the meta-construction from Section 4. For lack of space we omitted most of the proofs. These can be found in the full version of this paper [6].

2 Overview of Techniques

At the heart of our constructions is a combination of two techniques.[3]

Reduction to polynomial evaluation. A first technique is a reduction of the retrieval problem to the problem of multivariate polynomial evaluation. Specifically, the retrieval of x_i, where the servers hold x and the user holds i, is reduced to an evaluation of a multivariate polynomial p_x, held by the servers, on a point $E(i)$, which the user determines based on i. We refer to $E(i)$ as the *encoding* of i. As observed in [5] and, more generally, in [11], the degree of p_x can be decreased by increasing the length of the encoding $E(i)$ (i.e., the number of variables in p_x). Originating in [4], different variants of this reduction have been implicitly or explicitly used in virtually every PIR-related construction. Interestingly, encodings realizing the optimal length-degree tradeoff, which were utilized in [11, 12] to obtain special families of PIR protocols with short answer length, could not be used in protocols optimizing the *total* communication complexity [11, 1, 15]. We remedy this situation in the current work, and consequently get a constant-factor improvement to the communication complexity even in the 2-server case.

Simultaneous messages protocols for polynomial evaluation. A main ingredient in our new protocols is a generalization of a solution by Babai, Kimmel, and Lokam [3] to a communication complexity problem of computing the *generalized addressing function* in the so-called *simultaneous messages* (SM) model. Interestingly, this problem was motivated by circuit lower bounds questions, completely unrelated to privacy or coding. Towards solving their problem, they consider the following scenario. A degree-d m-variate polynomial p is known to k players, and k points $y_1, y_2, \ldots, y_k$ (each being an m-tuple of field elements) are distributed among them such that player j knows all points except y_j. An external referee knows *all* k points y_j but does not know p. How efficiently can the value $p(y_1 + y_2 + \ldots + y_k)$ be communicated to the referee if the players are restricted to simultaneously sending messages to the referee?

[3] A restricted use of the same approach has been made in the companion work [7].

A naive solution to the above problem is to have one of the players send an entire description of p to the referee. Knowing all y_j, the referee can then easily compute the required output. A key observation made in [3] is that it is in fact possible to do much better. By decomposing $p(y_1 + y_2 + \ldots + y_k)$ into terms and assigning each term to a player having the least number of unknown values, it is possible to write $p(y_1+\ldots+y_k)$ as the sum of k *lower degree* polynomials in the inputs, each known to one of the players. More precisely, player j can locally compute from its inputs a degree-$\lfloor d/k \rfloor$ polynomial p_j with its *unknown* inputs as indeterminates, such that $p(y_1 + \ldots + y_k) = p_1(y_1) + p_2(y_2) + \ldots + p_k(y_k)$. Then, by letting player j communicate the (much shorter) description of p_j, the referee can compute the required output. The amount of savings obtained by this degree reduction technique depends on the values of the parameters m,d, and k. In [3, 2], due to constraints imposed by the specific problem they consider, the degree-reduction technique is applied with rather inconvenient choices of parameters. Thus, in their setting the full savings potential of the technique has not been realized. It turns out that in the PIR context, where there is more freedom in the choice of parameters, the full spectrum of possible tradeoffs is revealed.

It is instructive to look at three useful choices of parameters: (1) If $d = 2k-1$, then the degree of each polynomial p_j is only $\lfloor (2k-1)/k \rfloor = 1$. When $m \gg d$, this $2k-1$ savings factor in the degree makes the description size of each p_j roughly the $(2k-1)$-th root of the description size of p. (2) If $d = k-1$, the degree of each p_j becomes 0, and consequently communicating each p_j requires sending a single field element. (3) Finally, if $m \gg d$ and $d \gg k$, then the cost of communicating p_j is roughly the k-th root of that of communicating p. These three examples, respectively, turn out to imply the existence of k-server PIR protocols with: (1) both queries and answers of length $O(n^{1/(2k-1)})$; (2) queries of length $O(n^{1/(k-1)})$ and answers of length $O(1)$; (3) queries of length $O(\log n)$ and answers of length $O(n^{1/k+\epsilon})$, for an arbitrarily small constant $\epsilon > 0$.

Combining the two techniques. In the case of 1-private PIR, the two techniques can be combined in the following natural way. On input i, the user computes an encoding $y = E(i)$ and the servers compute a degree-d polynomial p_x such that $x_i = p_x(E(i))$. To generate its queries, the user "secret-shares" $E(i)$ among the servers by first breaking it into otherwise-random vectors $y_1, \ldots, y_k$ which add up to y, and then sending to each server $\mathcal{S}_j$ all vectors except y_j. Using the SM communication protocol described in the previous section, the servers communicate $x_i = p_x(y)$ to the user.

This simple combination of the two techniques is already sufficient to yield some of the improved constructions. In the remainder of this work we generalize and improve the above solution in several different ways. First, we abstract its crucial components and formulate a generic "meta-construction" in these abstract terms. Second, we instantiate the abstract components to accommodate more general scenarios, such as t-private PIR. In the full version of this paper [6], we attempt at optimizing the amount of replication in the setting of [3], i.e., use a more efficient secret-sharing scheme for distributing $E(i)$, while maintaining the quality of the solution. This motivates various extensions of the SM communication model, which may be of independent interest.

3 Definitions

Notation. By $[k]$ we denote the set $\{1, \ldots, k\}$, and by $\binom{[k]}{t}$ all subsets of $[k]$ of size t. For a k-tuple v and a set $T \subseteq [k]$, let v_T denote the restriction of v to its T-entries. By Y_j for some j we represent a variable, while by the lower letter y_j we represent an assignment to the former variable. By H we denote the binary entropy function; that is, $H(p) = -p \log p - (1-p)\log(1-p)$, where all logarithms are taken to the base 2.

Polynomials. Let GF(q) denote the finite field of q elements. By $F[Y_1, \ldots, Y_m]$ we denote the linear space of all polynomials in the indeterminates $Y_1, \ldots, Y_m$ over the field F, and by $F_d[Y_1, \ldots, Y_m]$ its subspace consisting of all polynomials whose *total* degree is *at most* d, and whose degree in each indeterminate is at most $|F| - 1$. (The last restriction guarantees that each polynomial in $F_d[Y_1, \ldots, Y_m]$ represents a distinct function $p : F^m \to F$.) A natural basis for this linear space consists of all *monic monomials* satisfying the above degree restrictions. The case $F = \text{GF}(2)$ will be the most useful in this work. In this case, the natural basis consists of all products of at most d distinct indeterminates. Hence, $dim(F_d[Y_1, \ldots, Y_m]) = \sum_{w=0}^{d} \binom{m}{w}$ for $F = \text{GF}(2)$. We denote this dimension by $\Lambda(m, d) \stackrel{\text{def}}{=} \sum_{w=0}^{d} \binom{m}{w}$. We will also be interested in $F_d[Y_1, \ldots, Y_m]$ where $|F| > d$. In this case, the dimension of the space is $\binom{m+d}{d}$.

PIR protocols. We define single-round information-theoretic PIR protocols. A k-server PIR protocol involves k servers $\mathcal{S}_1, \ldots, \mathcal{S}_k$, each holding the same n-bit string x (the database), and a user who wants to retrieve a bit x_i of the database.

Definition 1 (PIR). *A k-server PIR protocol $\mathcal{P} = (\mathcal{R}, \mathcal{Q}_1, \ldots, \mathcal{Q}_k, \mathcal{A}_1, \ldots, \mathcal{A}_k, \mathcal{C})$ consists of a probability distribution $\mathcal{R}$ and three types of algorithms: query algorithms $\mathcal{Q}_j$, answering algorithms $\mathcal{A}_j$, and a reconstruction algorithm $\mathcal{C}$. At the beginning of the protocol, the user picks a random string r from $\mathcal{R}$. For $j = 1, \ldots, k$, it computes a query $q_j = \mathcal{Q}_j(i, r)$ and sends it to server $\mathcal{S}_j$. Each server responds with an answer $a_j = \mathcal{A}_j(q_j, x)$. Finally, the user computes the bit x_i by applying the reconstruction algorithm $\mathcal{C}(i, r, a_1, \ldots, a_k)$. A protocol as above is a* t-private *PIR protocol, if it satisfies: (1)* **correctness.** *The user always correctly computes x_i; (2)* t**-privacy.** *For every $i_1, i_2 \in [n]$ and $T \subseteq [k]$ such that $|T| = t$, the distributions $\mathcal{Q}_T(i_1, \mathcal{R})$ and $\mathcal{Q}_T(i_2, \mathcal{R})$ are identical.*

Linear secret-sharing. A *t-private secret-sharing scheme* allows a dealer to distribute a secret s among k players, such that any set of at most t players learns nothing on s from their joint shares, and any set of at least $t+1$ players can completely recover s from their shares. A secret-sharing scheme is said to be *linear over a field F* if $s \in F$, and the share received by each player consists of one or more linear combinations of the secret and r independently random field elements (where the same random field elements are used for generating all shares). A linear secret-sharing scheme is formally defined by a k-tuple $L = (L_1, \ldots, L_k)$ such that each L_j is a linear mapping from $F \times F^r$ to F^{ℓ_j}, where ℓ_j is the j-th player's share length. Finally, given a linear secret-sharing scheme as above, a vector in F^m will be shared by independently sharing each of its m entries. We next define two linear secret-sharing schemes that will be useful in the paper.

Definition 2 (The CNF scheme [16]). This scheme may work over any finite field (in fact, over any finite group), and proceeds as follows. To t-privately share a secret $s \in F$:

- *Additively* share s into $\binom{k}{t}$ shares, each labeled by a different set from $\binom{[k]}{t}$; that is, $s = \sum_{T \in \binom{[k]}{t}} r_T$, where the shares r_T are otherwise-random field elements.
- Distribute to each player P_j all shares r_T such that $j \notin T$.

The t-privacy of the above scheme follows from the fact that every t players miss exactly one additive share r_T (namely, the one labeled by their index set) and every set of $t+1$ players views all shares. The share vector of each party consists of $\binom{k-1}{t}$ field elements.

Definition 3 (Shamir's scheme [21]). Let $F = \mathrm{GF}(q)$, where $q > k$, and let $\omega_1, \ldots, \omega_k$ be distinct nonzero elements of F. To t-privately share a secret $s \in F$, the dealer chooses t random elements $a_1, \ldots, a_t$, which together with the secret s define a univariate polynomial $p(Y) \stackrel{\text{def}}{=} a_t Y^t + a_{t-1} Y^{t-1} + \ldots + a_1 Y + s$, and sends to the the j-th player the value $p(\omega_j)$.

4 The Meta-Construction

We describe our construction in terms of its abstract general components, and specify some useful instantiations for each of these components. In Section 5 several combinations of these instantiations are used for obtaining different families of PIR protocols.

Building blocks. There are three parameters common to all of our constructions: (1) a finite *field* F, (2) a *degree* parameter d, and (3) an *encoding length* parameter m. The database x is always viewed as a vector in F^n. Some variants of our construction use an additional *block length* parameter ℓ. All variants of our construction (as well as previous PIR protocols) can be cast in terms of the following abstract building blocks:

Linear space of polynomials. Let $V \subseteq F_d[Y_1, \ldots, Y_m]$ be a linear space of degree-d m-variate polynomials such that $\dim(V) \geq n$. The three most useful special cases are:
$\mathbf{V1}$**:** The space $F_d[Y_1, \ldots, Y_m]$ where $F = \mathrm{GF}(2)$; m and d must satisfy $\Lambda(m, d) \geq n$.
$\mathbf{V2}$**:** The space $F_d[Y_1, \ldots, Y_m]$ where $|F| > d$; here, m and d must satisfy $\binom{m+d}{d} \geq n$.
$\mathbf{V3}$**:** The linear subspace of $F_d[Y_1, \ldots, Y_m]$ such that $F = \mathrm{GF}(2)$ and V is spanned by the following basis of monomials. Let ℓ be an additional *block length* parameter, and let $m = \ell d$. We label the m indeterminate by $Y_{g,h}$, where $g \in [d]$ and $h \in [\ell]$. The basis of V will include all monic monomials containing exactly one indeterminate from each block, i.e., all monomials of the form $Y_{1,h_1} Y_{2,h_2} \cdots Y_{d,h_d}$. Since the number of such monomials is ℓ^d, the restriction on the parameters in this case is $\ell^d \geq n$.

Low-degree encoding. A *low-degree encoding* (with respect to the polynomial space V) is a mapping $E : [n] \to F^m$ satisfying the following requirement: There exist m-variate polynomials $p_1, p_2, \ldots, p_n \in V$ such that $\forall i, j \in [n]$, $p_i(E(j))$ is 1 if $i = j$ and is 0 otherwise. By elementary linear algebra, $\dim(V) \geq n$ is a necessary and sufficient condition for the existence of such an encoding. Given a low-degree encoding E and polynomials $p_1, p_2, \ldots, p_n$ as above, we will associate with each database $x \in F^n$ the polynomial $p_x \in V$ defined by $p_x(Y_1, \ldots, Y_m) = \sum_{i=1}^{n} x_i p_i$. Here x is fixed, and $x_1, \ldots, x_n$ are fixed coefficients (and not variables). Note that $p_x(E(i)) = x_i$ for every $i \in [n]$ and $x \in F^n$. With each of the above linear spaces we associate a natural low-degree encoding (see [6] for proofs of validity):[4]
$\mathbf{E1}$**:** $E(i)$ is the i-th vector in $\mathrm{GF}(2)^m$ of Hamming weight *at most* d.
$\mathbf{E2}$**:** Let $\omega_0, \ldots, \omega_d$ be distinct field elements. Then, $E(i)$ is the i-th vector of the form $(\omega_{f_1}, \ldots, \omega_{f_m})$ such that $\sum_{j=1}^{m} f_j \leq d$.
$\mathbf{E3}$**:** Let $(i_1, \ldots, i_d)$ be the d-digit base-ℓ representation of i (that is, $i = \sum_{j=1}^{d} i_j \ell^{j-1}$). Then, $E(i)$ is a concatenation of the length-ℓ unit vectors $e_{i_1}, e_{i_2}, \ldots, e_{i_d}$. The validity of this encoding follows by letting $p_i = Y_{1,i_1} \cdot \ldots \cdot Y_{d,i_d}$.

Linear secret-sharing scheme. Denoted by L. We will use either $L = \mathbf{CNF}$ (defined in Definition 2) or $L = \mathbf{Shamir}$ (defined in Definition 3). In the full version of this paper we also utilize a third scheme, which slightly optimizes $\mathbf{CNF}$.

[4] Since the *existence* of an appropriate encoding is implied by dimension arguments, the specific encoding being employed will usually not matter. In some cases, however, the encoding can make a difference. Such a case is discussed in Section 5.

Simultaneous messages communication protocol (abbreviated SM protocol). The fourth and most crucial building block is a communication protocol for the following promise problem, defined by the instantiations of the previous components V, E, and L. The problem generalizes the scenario described in Section 2. The protocol, denoted P, involves a user $\mathcal{U}$ and k servers $\mathcal{S}_1, \ldots, \mathcal{S}_k$.

- *User's inputs:* Valid L-shares $y^1, \ldots, y^k$ of a point $y \in F^m$. (That is, the k vectors y^j can be obtained by applying L to each entry of y, and collecting the shares of each player.) Moreover, it may be useful to rely on the following additional promise: $y = E(i)$ for some $i \in [n]$. Most of the protocols constructed in this paper do not make use of this additional promise.
- *Servers' inputs:* All k servers hold a polynomial $p \in V$. In addition, each $\mathcal{S}_j$ holds the share vector y^j.
- *Communication pattern:* Each server $\mathcal{S}_j$ sends a single message to $\mathcal{U}$ based on its inputs p, y^j. We let β_j denote a bound on the length of the message sent by $\mathcal{S}_j$.
- *Output:* $\mathcal{U}$ should output $p(y)$.

In Section 5 we will describe our constructions of SM protocols P corresponding to some choices of the space of polynomials V, the low degree encoding E, and the linear secret-sharing scheme L.

Putting the pieces together. A 4-tuple (V, E, L, P) instantiating the above 4 primitives defines a PIR protocol $\mathrm{PIR}(V, E, L, P)$. The protocol proceeds as follows.

- $\mathcal{U}$ lets $y = E(i)$, and shares y according to L among the k servers. Let y^j denote the vector of shares received by $\mathcal{S}_j$.
- Each server $\mathcal{S}_j$ lets $p = p_x$, and sends a message to $\mathcal{U}$ as specified by protocol P on inputs (p, y^j).
- $\mathcal{U}$ reconstructs $x_i = p(y)$ by applying the reconstruction function specified in P to $y^1, \ldots, y^k$ and the k messages it received.

The following lemma summarizes some properties of the above protocol.

Lemma 1. $\mathrm{PIR}(V, E, L, P)$ *is a t-private k-server PIR protocol, in which the user sends $m\ell_j$ field elements to each server $\mathcal{S}_j$ and receives β_j bits from each server (where ℓ_j is the share size defined by L and β_j is the length of message sent by $\mathcal{S}_j$ in P).*

Note that the only information that a server gets is a share of the encoding $E(i)$; the t-privacy of the secret sharing scheme ensures that a collusion of t servers learns nothing on i. For the query complexity, recall that $y = E(i) \in F^m$ and the user shares each of the m coordinates of y independently. Thus, the share size of server $\mathcal{S}_j$ is $m\ell_j$, where ℓ_j is the share size defined by L for sharing one coordinate (field element).

Some perspective concerning a typical choice of parameters is in place. In the typical case where k is viewed as a constant, all ℓ_j are also constant, and so the query complexity of $\mathrm{PIR}(V, E, L, P)$ is $O(m)$. If d is constant then, for any of the three vector spaces $\mathbf{V1}, \mathbf{V2}, \mathbf{V3}$, letting $m = O(n^{1/d})$ suffices to meet the dimension requirements. Thus, when both d, k are constants, the length of the queries in $\mathrm{PIR}(V, E, L, P)$ is $O(n^{1/d})$ and the length of the answers is determined by P.

In principle, the SM component in our construction could be replaced by a more general interactive protocol. However, there is yet no evidence that such an additional interaction may be helpful. Moreover, in defining an interactive variant of the fourth primitive one would have to take special care that the privacy requirement is not violated by the interaction. In the current non-interactive framework, the privacy property is guaranteed by the mere use of a t-private secret-sharing scheme.

5 Simultaneous Messages Protocols

We next describe SM protocols corresponding to useful combinations of V, E, and L. These may be viewed as the core of the PIR protocol.

Protocol P1. Protocol $\mathbf{P1}$ will serve as our default protocol. It may be viewed as a natural generalization of the protocol from [3]. The ingredients of this protocols are the polynomial space $\mathbf{V1} = F_d[Y_1, \ldots, Y_m]$ where $F = \mathrm{GF}(2)$, the encoding $\mathbf{E1}$ which encodes i as a vector in $\mathrm{GF}(2)^m$ of weight $\le d$, and the secret-sharing scheme $\mathbf{CNF}$.

Lemma 2. *For $V = \mathbf{V1}$, $E = \mathbf{E1}$, and $L = \mathbf{CNF}$, there exists an SM protocol $\mathbf{P1}$ with message complexity $\beta_j = \Lambda(m, \lfloor dt/k \rfloor)\binom{k-1}{t-1}^{\lfloor dt/k \rfloor}$.*

Proof. Let $y = \sum_T y_T$ be an additive sharing of y induced by the CNF sharing, such that the input y^j of $\mathcal{S}_j$ is $(y_T)_{j \notin T}$. The servers' goal is to communicate $p(y) = p(\sum_T y_T)$ to $\mathcal{U}$. Let $Y = (Y_{T,b})_{T \in \binom{[k]}{t}, b \in [m]}$, where each variable $Y_{T,b}$ corresponds to the input bit $(y_T)_b$, whose value is known to all servers $\mathcal{S}_j$ such that $j \notin T$. Define a $\binom{k}{t}m$-variate polynomial $q(Y) \stackrel{\text{def}}{=} p(\sum_{T \in \binom{[k]}{t}} Y_{T,1}, \ldots, \sum_{T \in \binom{[k]}{t}} Y_{T,m})$. Note that q has the same degree as p, and $q((y_T)_{T \in \binom{[k]}{t}}) = p(y)$. We consider the explicit representation of q as the sum of monomials, and argue that for every monomial $Y_{T_1,b_1} Y_{T_2,b_2} \ldots Y_{T_{d'},b_{d'}}$ of degree $d' \le d$ there exist some $j \in [k]$ such that at most $\lfloor dt/k \rfloor$ variables $Y_{T,b}$ with $j \in T$ appear in the monomial: Consider the multi-set $T_1 \cup T_2 \cup \ldots \cup T_{d'}$. This multi-set contains $d't \le dt$ elements, thus there must be some $j \in [k]$ that appears at most $\lfloor dt/k \rfloor$ times in the multi-set. We partition the monomials of q to k polynomials $q_1, \ldots, q_k$ such that q_j contains only monomials in which the number of the variables $Y_{T,b}$ with $j \in T$ is at most $\lfloor dt/k \rfloor$. Each monomial of q is in exactly one polynomial q_j, and therefore $q(Y) = \sum_{j=1}^{k} q_j(Y)$.

We are now ready to describe the protocol $\mathbf{P1}$. Denote by $\bar{Y}^j$ the set of variables whose values are *unknown* to the server $\mathcal{S}_j$ (that is, $\bar{Y}^j = (Y_{T,b})_{T \in \binom{[k]}{t}, j \in T, b \in [m]}$) and by $\bar{y}^j$ the corresponding values. Each $\mathcal{S}_j$ substitutes the values y^j of the variables it knows in q_j to obtain a polynomial $\hat{q}_j(\bar{Y}^j)$. The message of server $\mathcal{S}_j$ is a description of $\hat{q}_j$. The user, who knows the assignments to all variables, reconstructs the desired value by computing $\sum_{j=1}^{k} \hat{q}_j(\bar{y}^j) = q((y_T)_{T \in \binom{[k]}{t}}) = p(y)$.

MESSAGE COMPLEXITY. Recall that $\hat{q}_j$ is a degree-$\lfloor dt/k \rfloor$ multivariate polynomial with $m\binom{k}{t-1}$ variables. By the definition of q, not all monomials are possible: no monomial contains two variables $Y_{T_1,b}$ and $Y_{T_2,b}$ for some $b \in [m]$ and $T_1 \ne T_2$. Thus, to describe a possible monomial we need, for some $w \in \{0, \ldots, \lfloor dt/k \rfloor\}$, to choose w indices in $[m]$ and w sets of size t that contain j. Therefore, the number of possible monomials of $\hat{q}_j$ is at most $\sum_{w=0}^{\lfloor dt/k \rfloor} \binom{m}{w}\binom{k-1}{t-1}^{w} \le \Lambda(m, \lfloor dt/k \rfloor)\binom{k-1}{t-1}^{\lfloor dt/k \rfloor}$. Since each coefficient is from $\mathrm{GF}(2)$, the communication is as promised. □

Protocol P2. Protocol $\mathbf{P2}$ is useful for the construction of efficient PIR protocols with short answers (see Section 6). Unlike protocol $\mathbf{P1}$, which can be used with any combination of the parameters k, d, t, the applicability of $\mathbf{P2}$ is restricted to the case $k > dt$. That is, $k = dt + 1$ is the minimal sufficient number of servers. The first part of the following lemma is implicit in [4, 5, 11] and a special case of the second part is implicit in [12, 13]. The proof of the lemma appears in the full version of this paper [6].

Lemma 3. *For $V = \mathbf{V2}$, $E = \mathbf{E2}$, and $L = \mathbf{Shamir}$, and assuming that $k > dt$ and $|F| > k$, there exists an SM protocol $\mathbf{P2}$ in which each server sends a single field element. Moreover, given the promise that $p(y) \in F'$ for some subfield F' of F, it suffices for each server to send a single element of F'.*

A special case of interest is when $F' = \mathrm{GF}(2)$ and F is a sufficiently large extension field of F'. In this case, each message in the SM protocol consists of a single bit.

Protocol $\mathbf{P3}$. Special cases of the protocol $\mathbf{P3}$ are implicit in the 2-server PIR construction from [11] and its k-server generalization from [15]. A useful feature of this protocol is that it allows the user to compute his output by probing a small number of bits from the received messages. We only formulate this protocol for the 1-private case. Restricted generalizations to t-privacy may be obtained, using the approach of [15]. However, unlike the previous protocols, we do not know a "smooth" generalization to t-privacy. A proof of the following Lemma appears in the full version [6].

Lemma 4. *For $V = \mathbf{V3}$, $E = \mathbf{E3}$, and $L = \mathbf{CNF}$, there exists an SM protocol $\mathbf{P3}$ with message complexity $\beta_j = \ell^{\lfloor d/k \rfloor} \binom{d}{\lfloor d/k \rfloor}$ such that the user needs to read only $\binom{d}{\lfloor d/k \rfloor}$ bits from each message.*

6 Families of PIR Protocols

We now derive several explicit families of PIR protocols from the meta-construction.

Main family. Our main family of PIR protocols uses $\mathbf{V1}$, $\mathbf{E1}$, $\mathbf{CNF}$, and $\mathbf{P1}$. Protocols from this family yield our main improvements to the known upper bounds. We start with the general result, which follows from Lemmas 1 and 2, and then consider some interesting special cases.

Theorem 1. *Let m and d be positive integers such that $\Lambda(m, d) \geq n$. Then, for any k, t such that $1 \leq t < k$, there exists a t-private k-server PIR protocol with $\binom{k-1}{t} m$ query bits and $\Lambda(m, \lfloor dt/k \rfloor) \binom{k-1}{t-1}^{\lfloor dt/k \rfloor}$ answer bits per server.*

The total communication is optimized by letting $d = \lfloor (2k-1)/t \rfloor$ and $m = \Theta(n^{1/d})$. Substituting these parameters in Theorem 1 gives the following explicit bounds.

Corollary 1. *For any $1 \leq t < k$ there exist:*

- *A t-private k-server PIR protocol with $O_{k,t}(n^{1/\lfloor (2k-1)/t \rfloor})$ communication bits; this is a significant asymptotic improvement over the previous state of the art [15]. More precisely, the communication complexity is $O(\frac{k^2}{t} \binom{k}{t} n^{1/\lfloor (2k-1)/t \rfloor})$.*
- *A 1-private k-server PIR protocol with $k^2((2k-1)!n)^{1/(2k-1)} + k + k^3 = O(k^3 n^{1/(2k-1)})$ communication bits; this improves the previous best construction [15] by a constant factor which tends to e as k grows.*
- *A 1-private 2-server PIR protocol with $4(6n)^{1/3} + 2 \approx 7.27n^{1/3}$ communication bits; in comparison, the communication complexity of the best previously known 2-server protocol [11] is roughly $12n^{1/3}$.*

Another interesting case, discussed and used in [7], is when queries are short, i.e., of length $O(\log n)$; such protocols are obtained by letting $d = \theta m$, where $0 \leq \theta \leq 1/2$ is some constant. Substituting $m = (1/H(\theta) + o(1)) \log n$ and $d = \lfloor \theta m \rfloor$ in Theorem 1, and relying on the facts that $\lim_{\theta \to 0} \frac{H(\theta t/k)}{H(\theta)} = t/k$ and $\lim_{\theta \to 0} \frac{\theta}{H(\theta)} = 0$, we obtain:

Corollary 2. *For any constant integers $1 \leq t < k$ and constant $\epsilon > 0$, there exists a t-private k-server protocol with $O(\log n)$ query bits and $O(n^{t/k+\epsilon})$ answer bits. More precisely, for any $0 < \theta \leq 1/2$ one can obtain query length $\binom{k-1}{t}(1/H(\theta)+o(1))\log n$ and answer length $n^{(H(\theta t/k)+\theta\frac{t}{k}\log\binom{k-1}{t-1})/H(\theta)+o(1)}$.*

As observed in [18], a 1-private k-server PIR protocol with query length α and answer length β gives rise to a locally decodable code of length $k \cdot 2^\alpha$ over the alphabet $\Sigma = \{0,1\}^\beta$: A string $x \in \{0,1\}^n$ is encoded by concatenating the answers of all servers on all possible queries, where x is viewed as the database. If $\alpha = O(\log n)$, then the code length is polynomial. By substituting $t = 1$ in Corollary 2 and applying the above transformation we get:

Corollary 3. *For any constant integer $k \geq 2$ and constant $\epsilon > 0$, there exist positive constants δ_k, ρ_k, such that the following holds: There is a family $C(n)$ of* polynomial-length *(k, δ_k, ρ_k)-locally decodable codes over $\Sigma(n) = \{0,1\}^{\beta(n)}$, where $\beta(n) = O(n^{1/k+\epsilon})$.*

Boolean family. We now derive the construction of the most efficient known PIR protocols with a single answer bit per server. These are obtained by using $\mathbf{V2}$, $\mathbf{E2}$, $\mathbf{Shamir}$, and $\mathbf{P2}$. Protocols from this family, utilized in [12, 13], optimize similar protocols from [4, 5, 11] in which each answer consists of a single element from a moderately sized field. While the asymptotic communication complexity of protocols from this family is worse than that of the best unrestricted protocols, these protocols have found various applications. In particular they imply: (1) the most efficient constructions of binary locally decodable codes known to date; (2) very efficient PIR protocols for retrieving large records or "streams" of data; (3) PIR protocols with an optimal amount of total *on-line* communication (see [12]); (4) PIR protocols with poly-logarithmic amount of *on-line* work by the servers (see [7]).

Theorem 2 (Implicit in [12]). *Let m and d be positive integers such that $\binom{m+d}{d} \geq n$. Then, for any $t \geq 1$, there exists a t-private k-server PIR protocol with $k = dt + 1$ servers, $\lceil \log(k+1) \rceil m$ query bits per server, and a* single *answer bit per server.*

Corollary 4. *For any constant $d, t \geq 1$ there is a t-private PIR protocol with $k = dt+1$ servers, $O(n^{1/d})$ query bits, and a single answer bit per server.*

Cube family. Our last family of protocols generalizes the 2-server protocol from [11] and its k-server generalization from [15]. It relies on $\mathbf{V3}$, $\mathbf{E3}$, $\mathbf{CNF}$, and $\mathbf{P3}$ as building blocks. The communication in these protocols is not optimal, but they have the advantage of requiring the user to read fewer bits from the answers. These protocols have the interpretation of utilizing the "combinatorial cubes" geometry which was first used in [11]. We start with the general result, and then consider interesting special cases.

Theorem 3 (Generalizing [11, 15]). *Let d and ℓ be positive integers such that $\ell^d \geq n$. Then, for any $k \geq 2$ there exists a 1-private k-server PIR protocol with $(k-1)d\ell$ query bits per server and $\ell^{\lfloor d/k \rfloor}\binom{d}{\lfloor d/k \rfloor}$ answer bits per server, in which the user needs to read only $\binom{d}{\lfloor d/k \rfloor}$ bits from each answer.*

Corollary 5, which already appears in [11, 15], minimizes the total communication.

Corollary 5 ([11, 15]). *For any $k \geq 2$ there exists a 1-private k-server PIR protocol with $O(k^3 \cdot n^{1/(2k-1)})$ communication bits in which the user reads only $2k-1$ bits from each answer.*

As a special case, utilized in [7], we may get protocols with logarithmic query length.

Corollary 6. *For any integer $k \geq 2$ and $\delta < 1$, there exists a 1-private k-server PIR protocol with query length $O(k2^{1/\delta}\delta \log n)$ and answer length $O(n^{1/k+H(1/k)\delta})$ in which the user reads only $O(n^{H(1/k)\delta})$ bits from each answer.*

Acknowledgments. We thank Eyal Kushilevitz, Tal Malkin, Mike Saks, Yoav Stahl, and Xiaodong Sun for helpful related discussions.

References

1. A. Ambainis. Upper bound on the communication complexity of private information retrieval. In *Proc. of the 24th ICALP*, vol. 1256 of *LNCS*, pp. 401–407. 1997.
2. A. Ambainis and S. Lokam. Improved upper bounds on the simultaneous messages complexity of the generalized addressing function. In *LATIN 2000*, vol. 1776 of *LNCS*, pp. 207 – 216. 2000.
3. L. Babai, P. G. Kimmel, and S. V. Lokam. Simultaneous messages vs. communication. In *12th STOC*, vol. 900 of *LNCS*, pp. 361–372. 1995.
4. D. Beaver and J. Feigenbaum. Hiding instances in multioracle queries. In *STACS '90*, vol. 415 of *LNCS*, pp. 37–48. 1990.
5. D. Beaver, J. Feigenbaum, J. Kilian, and P. Rogaway. Locally random reductions: Improvements and applications. *J. of Cryptology*, 10(1):17–36. 1997.
6. A. Beimel and Y. Ishai. Information-Theoretic Private Information Retrieval: A Unified Construction. TR01-15, Electronic Colloquium on Computational Complexity. 2001.
7. A. Beimel, Y. Ishai, and T. Malkin. Reducing the servers' computation in private information retrieval: PIR with preprocessing. In *CRYPTO 2000*, vol. 1880 of *LNCS*, pp. 56–74. 2000.
8. J. Benaloh and J. Leichter. Generalized secret sharing and monotone functions. In *CRYPTO '88*, vol. 403 of *LNCS*, pp. 27–35. 1990.
9. C. Cachin, S. Micali, and M. Stadler. Computationally private information retrieval with polylogarithmic communication. In *EUROCRYPT '99*, vol. 1592 of *LNCS*, 402–414. 1999.
10. B. Chor and N. Gilboa. Computationally private information retrieval. In *Proc. of the 29th STOC*, pp. 304–313. 1997.
11. B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. *J. of the ACM*, 45:965–981. 1998.
12. G. Di-Crescenzo, Y. Ishai, and R. Ostrovsky. Universal service-providers for private information retrieval. *J. of Cryptology*, 14(1):37–74. 2001.
13. Y. Gertner, Y. Ishai, E. Kushilevitz, and T. Malkin. Protecting data privacy in private information retrieval schemes. *JCSS*, 60(3):592–629. 2000.
14. O. Goldreich and L. Trevisan. On the length of linear error-correcting codes having a 2-query local decoding procedure. Manuscript, 2000.
15. Y. Ishai and E. Kushilevitz. Improved upper bounds on information theoretic private information retrieval. In *Proc. of the 31th STOC*, pp. 79 – 88. 1999.
16. M. Ito, A. Saito, and T. Nishizeki. Secret sharing schemes realizing general access structure. In *Proc. of Globecom 87*, pp. 99–102. 1987.
17. H. Karloff and L. Schulman. Manuscript, 2000.
18. J. Katz and L. Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *Proc. of the 32th STOC*, pp. 80–86. 2000.
19. E. Kushilevitz and R. Ostrovsky. Replication is not needed: Single database, computationally-private information retrieval. In *Proc. of the 38th FOCS*, pp. 364–373. 1997.
20. E. Mann. Private access to distributed information. Master's thesis, Technion – Israel Institute of Technology, Haifa, 1998.
21. A. Shamir. How to share a secret. *CACM*, 22:612–613. 1979.