PIR schemes with small download complexity and low storage requirements*

Simon R. Blackburn, *Senior Member, IEEE*†, Tuvi Etzion, *Fellow, IEEE*‡, and Maura B. Paterson§

August 22, 2019

Abstract

In the classical model for (information theoretically secure) Private Information Retrieval (PIR) due to Chor, Goldreich, Kushilevitz and Sudan, a user wishes to retrieve one bit of a database that is stored on a set of n servers, in such a way that no individual server gains information about which bit the user is interested in. The aim is to design schemes that minimise the total communication between the user and the servers. More recently, there have been moves to consider more realistic models where the total storage of the set of servers, or the per server storage, should be minimised (possibly using techniques from distributed storage), and where the database is divided into R-bit records with $R > 1$, and the user wishes to retrieve one record rather than one bit. When R is large, downloads from the servers to the user dominate the communication complexity and so the aim is to minimise the total number of downloaded bits. Work of Shah, Rashmi and Ramchandran shows that at least $R + 1$ bits must be downloaded from servers in the worst case, and provides PIR schemes meeting this bound. Sun and Jafar have considered the download cost of a scheme, defined as the ratio of the message length R and the total number of bits downloaded. They determine the best asymptotic download cost of a PIR scheme (as $R \to \infty$) when a database of k messages is stored by n servers. This paper provides various bounds on the download complexity of a PIR scheme, generalising those of Shah et al. to the case when the number n of servers is bounded, and providing links with classical techniques due to Chor et al. The paper also provides a range of constructions for PIR schemes that are either simpler or perform better than previously known schemes. These constructions include explicit schemes that achieve the best asymptotic download complexity of Sun and Jafar with significantly lower upload complexity, and general techniques for constructing a scheme with good worst case download complexity from a scheme with good download complexity on average.

*Parts of this paper were presented at the International Symposium on Information Theory, Aachen, Germany, June 2017. Some of the research was performed while the second author visited Royal Holloway University of London under EPSRC Grant EP/N022114/1. The research was partly supported by the NSF-BSF grant 2016692.

†Simon R. Blackburn is with the Department of Mathematics, Royal Holloway University of London, Egham, Surrey TW20 0EX, United Kingdom, e-mail: s.blackburn@rhul.ac.uk

‡Tuvi Etzion is with the Department of Computer Science, Technion, Haifa 3200003, Israel, e-mail: etzion@cs.technion.ac.il.

§Maura B. Paterson is with the Department of Economics, Mathematics and Statistics, Birkbeck, University of London, Malet Street, London WC1E 7HX, United Kingdom, e-mail: m.paterson@bbk.ac.uk.

1 Introduction

1.1 The PIR Model

In the classical model for private information retrieval (PIR) due to Chor, Goldreich, Kushile-

vitz and Sudan [14], a database $\mathbf{X}$ is replicated across n servers $S_1, S_2, \ldots, S_n$. A user wishes to retrieve one bit of the database, so sends a query to each server and downloads their reply. The user should be able to deduce the bit from the servers' replies. Moreover, no single server should gain any information on which bit the user wishes to retrieve (without collusion). The resulting protocol is known as an (information-theoretic) *PIR scheme*; there are also computational variants of the security model [30]. The goal of PIR is to minimise the total communication between the user and the servers.

In practice, the assumption that the user only wishes to retrieve one bit of the database, and the assumption that there is no shortage of server storage seem unrealistic. Because of this, many recent papers assume that the database $\mathbf{X}$ consists of k *records*, each of which is R bits in length, so that the number of possible databases is 2^{kR}. We denote the value of Record i by X_i, and we write X_{ij} for the jth bit of X_i. The aim of the protocol is for the user to retrieve the whole of X_i, rather than a single bit. We also, following Shah, Rashmi and Ramchandran [38], drop the assumption that the whole database is replicated across the n servers $S_1, S_2, \ldots, S_n$ and so, for example, there is the possibility of using techniques from coding theory in general and from distributed storage codes in particular to reduce the total storage of the scheme. No restrictions are made on the particular encoding used to distribute the database across the servers other than to assume it is deterministic, *i.e.* that there is a unique way to encode each database. This important generalisation of the model has led to very interesting recent work which we discuss in Subsection 1.3 below. Our work is a follow-up to [38] with modifications, improvements, complementary results, simplifications, constructions, and additional aspects which were not considered in their paper.

More combinatorially, we define a private information retrieval scheme as follows.

Definition 1.1 (PIR scheme)**.** Suppose a database $\mathbf{X}$ is distributed across n servers $S_1, S_2, \ldots, S_n$. A user who wishes to learn the value X_ℓ of Record ℓ submits a *query* $(q_1, q_2, \ldots, q_n)$. For each $r \in \{1, 2, \ldots, n\}$, server S_r receives q_r and responds with a value c_r that depends on q_r and on the information stored by S_r. The user receives the *response* $(c_1, c_2, \ldots, c_n)$. This system is a *private information retrieval (PIR) scheme* if the following two properties are satisfied:

- **(Privacy)** For $r = 1, 2, \ldots, n$ the value q_r received by server S_r reveals no information about which record is being sought.
- **(Correctness)** Given a response $(c_1, c_2, \ldots, c_n)$ to a query $(q_1, q_2, \ldots, q_n)$ for Record ℓ, the user is unambiguously able to recover the value X_ℓ of this record.

Note that while the query is drawn randomly according a pre-specified distribution on a set of potential queries, the response is assumed to be deterministic.

Example 1.1. In the case of a single server, a trivial method for achieving PIR is for the user to download the entire kR-bit database.

Chor, Goldreich, Kushilevitz and Sudan showed that in the case of single-bit records ($R = 1$), if there is a single server then PIR is only possible if the total communication is at least k bits (*i.e.* the size of the entire database) [14], and so the solution above is best possible. We are interested in finding solutions such as the scheme below, which transmit significantly fewer than kR bits.

Example 1.2. [14] Suppose there are two servers, each storing the entire database. Suppose $R = 1$.

- A user who requires Record ℓ chooses a k-bit string $(\alpha_1, \alpha_2, \ldots, \alpha_k)$ uniformly at random.
- Server 1 is requested to return the value $c_1 = \bigoplus_{i=1}^{k} \alpha_i X_i$, and Server 2 is requested

to return $c_2 = \left(\bigoplus_{i=1}^{k} \beta_i X_i\right)$, where

$$\beta_i = \begin{cases} \alpha_i \oplus 1 & \text{when } i = \ell, \\ \alpha_i & \text{otherwise.} \end{cases}$$

- The user computes $c_1 \oplus c_2$ to recover the value X_ℓ of Record ℓ.

The strings $(\alpha_1, \alpha_2, \ldots, \alpha_k)$ and $(\beta_1, \beta_2, \ldots, \beta_k)$ are both uniformly distributed, and are independent of the choice of ℓ, hence neither server receives any information as to which record is being recovered by the user.

We note that the scheme above works unchanged when the records are R-bit strings rather than single bits. The download complexity of the scheme, in other words the total number of bits downloaded from the servers, is $2R$. The upload complexity is $2k$, since each server receives a k-bit string from the user. Thus the total communication of the scheme is $2R + 2k$ bits, which is significantly less than kR bits for most parameters.

Note that the upload complexity of this scheme does not depend on R, and so is an insignificant proportion of the total communication when R is large. This is a general phenomenon: Chan, Ho and Yamamoto [13, Remark 2] observe the following. Let $m > 1$ be an integer. Suppose we have an n-server PIR scheme for a database of k records, each R bits long. Suppose the scheme requires u upload bits and d download bits. Then we can construct an n-server PIR scheme for a database of k records, each mR bits long, which requires md bits of download but still needs just u bits to be uploaded. Note that when m is large (so records are long) the communication complexity of the new scheme is dominated by the download complexity of the given scheme.

Because of the observation of Chan et al., it is vital to find PIR schemes with low download complexity. We formalise the download complexity as follows.

Definition 1.2. A PIR scheme *uses binary channels* if the response c_j sent by server S_j is a binary string of length d_j, where d_j depends only on the query q_j it receives. The *download complexity* is the maximum of the sum $\sum_{j=1}^{n} d_j$ over all possible queries $(q_1, q_2, \ldots, q_n)$.

So the download complexity is the number of bits downloaded in the worst case. We emphasise that the length d_j in the definition above does not depend on the database $\mathbf{X}$, but could depend on the query q_j received by server S_j. We note that we allow for the possibility that $d_j = 0$, so the server does not reply to the query. Finally, we note that if we know that there are more than 2^x distinct possibilities for c_j as the database varies, we may deduce that $d_j \geq x+1$. Although it is possible to use non-binary channels for PIR as was done recently (see Subsection 1.3), we restrict our exposition to PIR schemes using binary channels in this paper. Most schemes in the literature before the work of [38] implicitly use this model, as they use fields of characteristic 2 and transmit strings of bits. This restricted model is implicit also in Shah et al. [38] when a lower bound on the download complexity is proved. This model is only required for those results in Section 2 paper that are used to generalise their bound. Although most results in the other sections can be generalised for the non-binary case we prefer not to do so for simplicity.

We should comment that, despite the observation of Chan et al., we should not ignore upload complexity completely, as there are scenarios (for example, when R is not so large) when it might be dominant. Moreover, we cannot compare the difficulty of uploading $2k$ bits with downloading $2R$ bits just by comparing k and R, since we know (at least currently) that in practice it takes much more time to upload a bit than to download one. Of course, we don't know how the speed of downloading and uploading will change over time. But the obvious consequence of the current situation and future developments is to consider both upload and download complexities separately, and not to ignore one of them completely. This is something that will be done in this paper, although the download complexity

will be the main target for optimisation since we generally assume that the size of the database is (considerably) larger than the one bit of the classical PIR model.

We continue to another important measure that has motivated many papers in the last three years, after being introduced by Shah et al. [38]:

Definition 1.3. Suppose server S_r stores s_r bits of information about the database $\mathbf{X}$.

- The *per-server storage* of the scheme is $\max\{s_r \mid r = 1, 2, \ldots, n\}$.
- The *total storage* of the scheme is $\sum_{r=1}^{n} s_r$.
- The *storage overhead* of the scheme is the ratio between the total storage and the total size of the records in the database, i.e. kR.

The classical model of PIR ignored storage issues: it was assumed that there is enough storage to allow the replication of the database at each server. But, with the quantity of information stored today in data centres, storage is an issue today and might be an important barrier in the future. Thus, it is important to reduce the storage overhead as much as possible, while keeping reliability, fast access, fast upload and fast download at reasonable levels. This is the perspective of the current paper, which is concerned with schemes whose download complexity is as small as possible whilst keeping the total storage at reasonable levels.

Finally, it should be noted that although most of the work in this area is theoretical, there have been notable recent advances in bridging the gap between theory and practice, e.g. [24, 54] as we highlight in Subsection 1.3.

1.2 Our contributions

In Section 2, we provide combinatorial results on the structure of a PIR scheme with small download complexity:

- We generalise (Theorem 2.2) a key theorem in the foundational paper of Chor et al. [14], and use this result to generalise the lower bound of $R + 1$ on download complexity in [38]. The results imply (Theorem 2.5) that an n-server PIR scheme must have download complexity at least $\frac{n}{n-1}R$ when $k > \lceil R/(n-1) \rceil$. (This last result can also be obtained as a corollary of a recent bound due to Sun and Jafar [39].) These results provide a bridge between classical PIR and the new models that are assuming the retrieval of long records. Moreover (as often happens with a combinatorial approach), some extra structural information on schemes is provided: see Theorem 2.3.
- We provide (Corollary 2.6, Theorem 2.7) information on the structure of a PIR scheme with minimal download complexity $R+1$. In particular, Theorem 2.7 provides a rigorous statement of [38, Theorem 1].

In Section 3, we provide various constructions for PIR schemes with low download complexity:

- In Subsection 3.1, we provide two simple $(R + 1)$-server PIR schemes with download complexity $R + 1$. Both schemes have total storage which is quadratic in R. The first scheme is a natural generalisation of the scheme of Chor et al. given above. The second scheme is a close variant of the quadratic total storage PIR scheme in [38], which avoids having to design slightly different schemes depending on the parity of R. This second scheme is to be preferred due to its lower upload complexity. (Another, more complex, PIR scheme with download complexity $R + 1$ is considered in detail in [38]. This scheme has small per-server storage, but requires an exponential (in R) number of servers, and so has exponential total storage.)
- In Subsection 3.2, we describe an n-server PIR scheme with download complexity $\frac{n}{n-1}R$. The total storage of the scheme is linear in R. This shows that for any $\epsilon > 0$

there exists a PIR scheme with linear total storage and download complexity at most $(1+\epsilon)R$. (Schemes with linear total storage, but with download complexity between $2R$ and $4R$, are given in [38].)

- We describe (Subsection 3.3) schemes that provide trade-offs between increasing the number of servers and reducing the per-server storage of the scheme in Subsection 3.2.

- In Subsection 3.4, we provide explicit schemes that achieve optimal asymptotic download cost. The performance of these schemes is equal to the inductively defined schemes in Sun and Jafar [39], but the description of these schemes is more concise, and the proof that they are indeed PIR schemes is much more straightforward.

- Finally, in Subsection 3.5, we explain an averaging technique that allows a PIR scheme with good average download complexity to be transformed into a scheme with good download complexity in the worst case.

1.3 Context

We end this introduction with a discussion of some of the related literature. (Many of these papers appeared after the conference version of our paper [11] was posted. We omit results submitted after our submission of this paper.)

Private information retrieval was introduced in [14], and has been an active area ever since. See, for example, Yekhanin [55] for a fairly recent survey.

The papers by Shah et al. [38] and (independently) by Augot, Levy-Dit-Vahel, and Shikfa [3] are the first to consider PIR models where the information stored in the servers could be coded using techniques from distributed storage. Whereas [38] is mainly concerned with download complexity, and also with total storage (with per-server storage, and query size also relevant parameters), the authors of [3] emphasise measures of robustness against malicious servers, namely decoder locality and PIR locality.

More recently, the literature has addressed several parallel and related issues, which can be categorised as follows:

1. Papers dealing with the download complexity, rate, and capacity of PIR schemes.

2. Research which attempts to reduce the storage overhead of PIR schemes.

3. Papers which present coding techniques, based on various error-correcting codes, e.g. MDS codes, to store the database in a distributed fashion.

4. Papers which consider PIR schemes in the presence of unreliable servers. Servers might be colluding (so they have access to more than one query q_r), they might fail (and so do not reply with a value c_r), they might be adversarial (replying with incorrect values c_r), they might be unsynchronised (storing slightly different copies of the database) and so on.

5. Research which aims to build PIR schemes into previously known architectures for distributed storage.

6. Papers dealing with other PIR models, for example allowing broadcasting of some information, or allowing the user to possess side information such as the value of some records.

Clearly, these issues are related, and a given paper might address aspects of more than one of these topics.

In early papers, Fanti and Ramchandran [17, 18] considered unsynchronized databases; the results are the same as for synchronized PIR at the expense of probabilistic success for information retrieval, and the use of two rounds of communication. We are not aware of recent work in this model, but we mention in this context the work

of Tajeddine and El Rouayheb [47] which considers PIR schemes in the presence of some servers which do not respond to a query.

In a sequence of papers, Sun and Jafar [39, 41, 43, 44, 45] consider the capacity of the channels related to PIR codes in various models. (The *rate* of a PIR scheme is the ratio of R and the download complexity, and the *capacity* is the supremum of achievable rates.) In the model for PIR we consider, they use information theoretic techniques to show [39] that an n-server PIR scheme on a k message database has rate at most

$$\left(1-\frac{1}{n}\right)\left(1-\left(\frac{1}{n}\right)^k\right)^{-1}.$$

Their model is restricted to the special case of replication. They also provide a scheme that attains this rate. The messages in their scheme are extremely long for most values of n and k: the message length must be a multiple of n^k. Because of this, the scheme can be thought of as being tailored for the situation when $R \to \infty$. Their results show that when $R \to \infty$ with n and k fixed, there are schemes whose download complexity (and so whose communication complexity) has a leading term of the form

$$\frac{n}{n-1}\left(1-\left(\frac{1}{n}\right)^k\right)R,$$

and that this term is best possible. (We give an explicit scheme with the same download complexity in Subsection 3.4.)

The results in [39] have been generalised to the case when some of the servers collude. Sun and Jafar [43] find the capacity of the channel in this more general case. (The results in [39] can be thought of as the special case where each server can collude only with itself.) The capacity for the symmetric PIR model, where the user who retrieves a message will get no information about the other messages in the database, is determined in [45]. The optimal download complexity in the situation when the messages in the database might be of an arbitrary length (subject to a certain divisibility condition with all messages having the same length) is considered in [41]. The most recent in this sequence of papers considers an interactive model, where a user can have several rounds of queries, the queries in a given round are allowed to depend on answers from previous rounds. Moreover, colluding servers are considered in this model. It is proved [44] that for this case there is no change in the capacity, but that the storage overhead can sometimes be improved.

Banawan and Ulukus [5] also generalise the results of Sun and Jafar [39], finding the exact capacity of the PIR scheme when the database is encoded with a linear code. Another generalisation due to Banawan and Ulukus [4, 6] is to the scenario that the user is allowed to request a few records in one round of queries. They provide capacity computations and schemes for this scenario. A similar case was also discussed in [59]. Finally, Banawan and Ulukus [7] consider the capacity of PIR schemes in the scenario where servers might not be synchronised, there might be adversarial errors, and some servers might collude. They compute the capacity when some or all of these events might occur. Wang and Skoglund [51] consider the capacity of a symmetric PIR scheme when the database is stored in a distributed fashion using an MDS code.

Chan, Ho, and Yamamoto [12, 13] consider the trade-off between the total storage and the download complexity when the size of a record is large; the trade-off depends on the number of records in the system. They also consider the case where the database is encoded with an MDS code.

Fazeli, Vardy, and Yaakobi [19, 20] give a method to reduce the storage overhead based on any known PIR scheme which uses replication. Their method reduces the storage overhead considerably, without affecting the order of the download complexity or upload complexity of the overall scheme, by simulating the original scheme on a larger number of servers. Their key concept is an object they call a κ-PIR code (more generally a κ-PIR array code), where κ is the number of servers used in the orig-

inally known PIR scheme, which controls how a database can be divided into parts and encoded within servers to allow a trade-off between the number of servers and the storage overhead. In particular, for all $\epsilon > 0$, they show that there exist good schemes (in terms of communication requirements) where the amount of information stored in a server is bounded but the total storage is at most $(1+\epsilon)$ times the database size. Rao and Vardy [32] study PIR codes further, establishing the asymptotic behavior of κ-PIR codes. Vajha, Ramkumar, and Kumar [50] find the redundancy of such codes for $\kappa = 3,\ 4$ by using Reed-Muller codes. Lin and Rosnes [31] show how to shorten and lengthen PIR codes, and find the redundancy of such codes for $\kappa = 5,\ 6$. Blackburn and Etzion [9, 10] consider the optimal ratios between κ-PIR array codes and the actual number of servers used in the system. Zhang, Wang, Wei, and Ge [58] consider these ratios further, and improve some of the results from [9, 10]. We remark that though it is possible to reduce the storage overhead using the techniques of PIR array codes, it seems impossible to reduce the download complexity of the resulting schemes below $(3/2)R$ (and most codes give download complexity close to $2R$) because of restrictions on the PIR rate of such codes. It is interesting to note that Augot, Levy-Dit-Vahel, and Shikfa [3] constructed PIR schemes by partitioning the database into smaller parts, as done later in [19, 20], to reduce the storage overhead. But they applied this technique only to a certain family of multiplicity codes, and the parts of the partition were not encoded as in [19, 20].

Fazeli, Vardy, and Yaakobi [20] remark that the concept of a κ-PIR code is closely related to codes with locality and availability. Such codes were studied first by Rawat, Papailiopoulos, Dimakis, and Vishwanath [33, 34] and later also by others, for example [21, 25]. A new subspace approach for such codes was given recently in [36, 37]. Another family of related codes with similar properties are batch codes, which were first defined by Ishai, Kushilevitz, Ostrovsky, and Sahai [26] and were recently studied by many others, for example [1, 2, 35]. It is important to note that all these codes are very important in the theory of distributed storage codes. This connection between the concepts of locality and PIR codes are explored in [21].

Error-correcting codes, and in particular maximum distance separable (MDS) codes, have been considered by many authors in various PIR models. It is natural to consider MDS codes, as they are very often used in various types of distributed storage codes (especially for locally repairable codes [23] and regenerating codes [15, 16]), and we expect that the servers in our PIR scheme will be part of a distributed storage system. We will now mention various examples.

Colluding or malicious servers in PIR have been much studied over the last two years. Tajeddine and El Rouayheb [46] consider PIR schemes where the information is stored using MDS codes. Their PIR scheme based on the coded MDS achieves a retrieval rate $1-R$, where R is the code rate of the storage system. They attain the bounds for linear schemes in [12, 13], in the situation when one or two 'spies' (colluding and/or malicious servers) are present. In the case of one spy (no collusion) a generalisation to any linear code with rate greater than half was given in [29]. Freij-Hollanti, Gnilke, Hollanti, and Karpuk [22] give a PIR scheme coded with an MDS code which can be adjusted (by varying the rate of the MDS code) to combat against larger numbers of colluding servers. This scheme also attains the asymptotic bound on the related capacity of such a PIR scheme in the extreme cases, where there are no colluding servers or when the database is replicated, i.e. no coding is applied. This idea is generalised in [48]. The results in the latter paper are analysed (and one conjecture disproved) by Sun and Jafar [40, 42]. Another scheme based on MDS codes which can combat large number of colluding servers is given by Zhang and Ge [56]. A generalisation to the case where the user wants to retrieve several files

is given by the same authors in [57]. Wang and Skoglund [52] consider a symmetric PIR scheme using an MDS code, in which the user can retrieve the information about the file he wants, but can gain no information about the other files. This scheme attains the bound on the capacity which they derive earlier in [51]. They have extended their work to accommodate colluding servers in [53].

PIR can be combined with other applications in storage and communication in many ways. One example is a related broadcasting scheme in [28]. Another example is cache-aided PIR, considered by Tandon [49]. In this setup the user is equipped with a local cache which is formed from an arbitrary function on the whole set of messages, and this local cache is known to the servers. The situation when this cache is not known to the servers is considered by Kadhe, Garcia, Heidarzadeh, El Rouayheb, and Sprintson [27]. Since the user has side information in these models, the problem is closely related to index coding [8] a topic which is also of great interest.

While most of the work in this area is theoretical, there have been notable recent advances in bridging the gap between theory and practice. For example, the recent paper [24] reports on the design and implementation of a scalable and private media delivery system — called *Popcorn* — that explicitly targets Netflix-like content distribution. Another practical system for private queries on public datasets — called *Splinter* — is currently in development [54]. This system has been reported to achieve latencies below 1.20 seconds for realistic workloads including a Yelp clone, flight search, and map routing.

2 Optimal download complexity

In this section, we give structural results for PIR schemes with optimal download complexity, given that the database consists of k records of length R. For some of the results, we also assume that the PIR scheme involves n servers, where n is fixed.

In Subsection 2.1 we generalise a classical result in Private Information Retrieval due to Chor et al. We use this result to provide an alternative proof of the theorem of Shah, Rashmi and Ramchandran [38] that a PIR scheme must have download complexity at least $R+1$ when $k \geq 2$, and to prove a lower bound of $\frac{n}{n-1}R$ for the download complexity of an n-server PIR scheme whenever k is sufficiently large. In Section 2.2 we present more precise structural results when the download complexity of a PIR scheme attains the optimal value of $R+1$ bits.

Definition 2.1. We say that a response $(c_1, c_2, \ldots, c_n)$ is *possible* for a query $(q_1, q_2, \ldots, q_n)$ if there exists a database $\mathbf{X}$ for which $(c_1, c_2, \ldots, c_n)$ is returned as the response to the query $(q_1, q_2, \ldots, q_n)$ when $\mathbf{X}$ is stored by the servers.

2.1 Lower bounds on the download complexity

We aim to generalise the following theorem, which was proved by Chor et al. in the very first paper on PIR [14, Theorem 5.1]:

Theorem 2.1. *A PIR scheme that uses a single server for a database with k records of size one bit is not possible unless the number of possible responses from the server to any given query is at least 2^k.*

Our generalisation shows a server must reply with at least $k(R-d)$ bits of download, if no more than a total of d bits (where $0 \leq d \leq R$) are downloaded from the other servers. We state our generalisation as follows. Without loss of generality we will focus on server S_1, so for ease of notation we will denote the tuple $(q_1, q_2, \ldots, q_n)$ by (q_1, q_{other}), and $(c_1, c_2, \ldots, c_n)$ by (c_1, c_{other}).

Theorem 2.2. *Suppose $0 \leq d \leq R$. Let q_1 be fixed. Suppose we have a PIR scheme*

with the property that for any query of the form (q_1, q_{other})*, we have*

$$|\{c_{\text{other}} \mid \exists c_1 \text{ such that } (c_1, c_{\text{other}}) \text{ is possible for } (q_1, q_{\text{other}})\}| \leq 2^d.$$

Then for any query (q_1, q'_{other}) *we have*

$$|\{c_1 \mid \exists c_{\text{other}} \text{ such that } (c_1, c_{\text{other}}) \text{ is possible for } (q_1, q'_{\text{other}})\}| \geq 2^{k(R-d)}.$$

We remark that Theorem 2.1 is the case $d = 0$ and $R = 1$ of Theorem 2.2.

Proof. Let q_1 be fixed, and suppose we have a PIR scheme with the property that for any query (q_1, q_{other})

$$|\{c_{\text{other}} \mid \exists c_1 \text{ such that } (c_1, c_{\text{other}}) \text{ is possible for } (q_1, q_{\text{other}})\}| \leq 2^d. \tag{1}$$

Assume, for a contradiction, that there exists a query $(q_1, q^*_{\text{other}})$ for which

$$|\{c_1 \mid \exists c_{\text{other}} \text{ such that } (c_1, c_{\text{other}}) \text{ is possible for } (q_1, q^*_{\text{other}})\}| < 2^{k(R-d)}.$$

Suppose this query is for Record i.

Let c_1^* be a most common reply of S_1 to $(q_1, q^*_{\text{other}})$ as the database varies over all possibilities. So we choose c_1^* to maximise $|T|$, where T is the set of databases where S_1 replies with c_1^* to the query $(q_1, q^*_{\text{other}})$. If server S_1 receives the query q_1, it will thus return c_1^* whenever a database in T is being stored. There are 2^{kR} databases, and less than $2^{k(R-d)}$ possibilities for the reply c_1 of S_1 to the query $(q_1, q^*_{\text{other}})$. So by the pigeonhole principle, $|T| > 2^{kR}/2^{k(R-d)} = 2^{kd}$.

Since the databases consist of k records, the fact that $|T| > 2^{kd}$ implies the existence of a record, say Record ℓ, for which the number of distinct values X_ℓ that appear among the databases in T is greater than 2^d. Thus we can choose a subset of $2^d + 1$ databases $W \subseteq T$ such that the values X_ℓ of Record ℓ in the databases in W are all distinct.

The requirement for privacy against server S_1 implies that there exists a query for Record ℓ of the form $(q_1, q^\ell_{\text{other}})$, since otherwise S_1 could distinguish between queries for Record i and Record ℓ.

Suppose the query $(q_1, q^\ell_{\text{other}})$ for Record ℓ is made, and suppose that the database lies in W. Server S_1 receives q_1, and so (since $W \subseteq T$) replies with c_1^*. But there are at most 2^d possible replies c^ℓ_{other} from the remaining servers by (1), and so there are at most 2^d responses $(c_1^*, c^\ell_{\text{other}})$ to the query $(q_1, q^\ell_{\text{other}})$. Since $|W| = 2^d + 1$, there are two databases $\mathbf{X}, \mathbf{Y} \in W$ such that the servers respond identically. But this is our required contradiction, since $\mathbf{X}$ and $\mathbf{Y}$ have distinct values for Record ℓ and the query was for this record. □

The following theorem is a key consequence of Theorem 2.2.

Theorem 2.3. *Let x be non-negative, and suppose we have a PIR scheme that has download complexity at most $R + x$. If the database contains k records, where $k \geq x+2$, then the number of bits downloaded from any server is at most x.*

Proof. Without loss of generality, consider the server S_1. Suppose for a contradiction that there exists a query q_1 so that at least $x + 1$ bits are downloaded from S_1 (and so at most $(R + x) - (x + 1) = R - 1$ bits are downloaded from the other servers). Suppose that a total of d bits are downloaded from the other servers in the worst case when S_1 receives q_1. So $d \leq R - 1$. Theorem 2.2 implies that at least $k(R-d)$ bits are downloaded from S_1, and so at least $k(R-d) + d$ bits are downloaded from the servers in the worst case. But $d \leq R - 1$ and $k \geq x + 2$, so

$$\begin{aligned} k(R-d) + d &= kR - (k-1)d \\ &\geq kR - (k-1)(R-1) \\ &= R + k - 1 \geq R + (x+2) - 1 \\ &= R + x + 1, \end{aligned}$$

which is impossible as the scheme has total download complexity $R + x$. This contradiction establishes the theorem. □

We are now in a position to provide a new short proof of the following corollary. The corollary is due to Shah et al. [38].

Corollary 2.4. *Let the database contain k records with $k \geq 2$. Any PIR scheme requires a total download of at least $R + 1$ bits.*

Proof. Suppose we have a scheme with total download of R or fewer bits. Theorem 2.3 with $x = 0$ implies that 0 bits are downloaded from each server, and so the user receives no information about the desired record. Hence such a scheme cannot exist. □

The following theorem (which can also be derived from the results in [39]), improves the bound of Corollary 2.4 when $n < R + 1$ and k is sufficiently large.

Theorem 2.5. *Suppose a PIR scheme involves n servers, where $n \geq 2$. Suppose the database contains k records, where $k \geq \lceil \frac{1}{n-1} R \rceil + 1$. Then the download complexity of the scheme is at least $\frac{n}{n-1} R$ bits.*

Proof. Assume for a contradiction that the scheme has download complexity $R + x$, where x is an integer such that $x < \frac{1}{n-1} R$. Since $x \leq \lceil \frac{1}{n-1} R \rceil - 1$, we see that $k \geq x + 2$ and so Theorem 2.3 implies that the number of bits downloaded by any server is at most x. Since we have n servers, the total number of bits of download is always at most xn. Since our scheme has download complexity $R + x$, there is a query where a total of $R + x$ bits are downloaded from servers. Hence we must have that $nx \geq R + x$, which implies that $x \geq \frac{1}{n-1} R$. This contradiction establishes the result. □

2.2 Download complexity $R + 1$

The final two results of this section concentrate on the extreme case when the download complexity is exactly $R + 1$. Recall that the download complexity is a worst case measure: every query results in at most $R + 1$ bits being downloaded, and there exists a query where $R + 1$ bits are downloaded.

Corollary 2.6. *Let the database contain k records with $k \geq 3$. Any PIR scheme with a total download of exactly $R + 1$ bits requires 1 bit to be downloaded from each of R or $R + 1$ different servers in response to any query.*

Proof. The special case of Theorem 2.3 when $x = 1$ shows that no server replies with more than 1 bit. For the download complexity to be $R + 1$, no more than $R + 1$ servers can respond non-trivially. Since the user deduces the value of an R-bit record from the bits it has downloaded, at least R servers must reply to any query. □

One might hope that the Corollary 2.6 could be strengthened to the statement that exactly $R + 1$ servers must respond non-trivially. However, examples show that this is not always the case: see the comments after Construction 1 below.

Shah et al. state [38, Theorem 1] that, in the situation above, "for almost every PIR operation" $R + 1$ servers must respond, and they provide a heuristic argument to support this statement. The following result makes this rigorous, with a precise definition of 'almost every'.

Theorem 2.7. *Let the database contain k records with $k \geq 3$. Suppose we have a PIR scheme with a total download of exactly $R+1$ bits (in the worst case). Suppose a user chooses to retrieve a record chosen with a uniform probability distribution on $\{1, 2, \ldots, k\}$. Let α be the probability that only R bits are downloaded. Then*

$$\alpha \leq \frac{R + 1}{kR + 1}.$$

Proof. By Corollary 2.6, each server replies to any query with at most one bit. We may assume, without loss of generality, that if a server replies with one bit then this bit must depend on the database in some way (since otherwise we may

modify the scheme so that this server does not reply and the probability α will increase).

Let $(q_1, q_2, \ldots, q_n)$ be a query for the ℓth record where only R servers reply non-trivially. Since only R servers reply, there are at most 2^R possible replies to the query (over all databases). But the value X_ℓ of the record is determined by the reply, and there are 2^R possible values of X_ℓ. So in fact there must be exactly 2^R possible replies, and there is a bijection between possible replies and possible values X_ℓ. We claim that the replies of each of these R servers can only depend on X_ℓ, not on the rest of the database. To see this, suppose a server S_r replies non-trivially, and let $f : \{0,1\}^{kR} \to \{0,1\}$ be the function mapping each possible value of the database to the reply of S_r to query q_r. Suppose f is not a function of X_ℓ alone, so there are two databases $\mathbf{X}$ and $\mathbf{X}'$ whose ℓth records are equal and such that $f(\mathbf{X}) \neq f(\mathbf{X}')$. Let ρ be the common value of the ℓth record in both $\mathbf{X}$ and $\mathbf{X}'$. When $X_\ell = \rho$ there are at least two possible replies to the query, depending on the value of the remainder of the database. But this contradicts the fact that we have a bijection between possible replies and possible values X_ℓ. So our claim follows.

Let A be the event that exactly R servers reply, and for $r = 1, 2, \ldots, n$ let B_r be the event that server S_r replies non-trivially. Let D_r be the indicator random variable for the event B_r. So D_r is equal to 1 when S_r responds non-trivially and 0 otherwise. Note that D_r is always equal to the number of bits downloaded from S_r, thus the expected value of the sum of these variables satisfies

$$\mathrm{E}\left(\sum_{r=1}^{n} D_r\right) = \alpha R + (1-\alpha)(R+1) = R + 1 - \alpha. \tag{2}$$

Let D'_r be the indicator random variable for the event $A \wedge B_r$. When A does not occur, all the variables D'_r are equal to 0. When A occurs, D'_r is the number of bits downloaded from server S_r and a total of R bits are downloaded. So

$$\mathrm{E}\left(\sum_{r=1}^{n} D'_r\right) = (1-\alpha)0 + \alpha R = \alpha R. \tag{3}$$

Suppose a server S_r uses the following strategy to guess the value of ℓ from the query q_r it receives. If the server replies non-trivially using a function f that depends on only one record, say Record ℓ', it guesses that $\ell = \ell'$. Otherwise, the server guesses a value uniformly at random. The server guesses correctly with probability $1/k$ when it responds trivially. The argument in the paragraph above shows the server always guesses correctly if it responds non-trivially and only R servers reply. Thus the server is correct with probability at least $(1/k)\Pr(\overline{B_r}) + \Pr(A \wedge B_r)$. The privacy requirement of the PIR scheme implies that the server's probability of success can be at most $1/k$, and so we must have that $\Pr(A \wedge B_r) \leq (1/k)\Pr(B_r)$. Hence

$$\mathrm{E}(D'_r) \leq (1/k)\mathrm{E}(D_r).$$

By linearity of expectation, we see that

$$\begin{aligned}\mathrm{E}\left(\sum_{r=1}^{n} D'_r\right) &= \sum_{r=1}^{n} \mathrm{E}(D'_r) \\ &\leq \frac{1}{k}\sum_{r=1}^{n} \mathrm{E}(D_r) = \frac{1}{k}\,\mathrm{E}\left(\sum_{r=1}^{n} D_r\right).\end{aligned}$$

So, using (2) and (3), we see that

$$\alpha R \leq \frac{1}{k}(R + 1 - \alpha).$$

Rearranging this inequality in terms of α, we see that the theorem follows. □

3 Constructions

Recall the notation from the introduction: we are assuming that our database $\mathbf{X}$ consists of k records, each of R bits, and we write X_{ij} for the jth bit of the ith record.

3.1 Two schemes with download complexity $R+1$

This section describes two schemes with download complexity $R+1$. Recall that this download complexity is optimal, by Corollary 2.4. The first scheme is included because of its simplicity; it can be thought of as a variation of the scheme of Chor et al. described in Example 1.2, and achieves optimal download complexity using only $R+1$ servers. It has a total storage requirement which is quadratic in R. But the scheme has high upload complexity: $kR(R+1)$. The second scheme is very closely related to a scheme mentioned in an aside in Shah et al. [38, Section IV]. This scheme has the same properties as the first scheme, except the upload complexity is improved to just $(R+1)k\lceil\log(R+1)\rceil$.

We note that the main scheme described in Shah et al. [38, Section IV] also has optimal download complexity of $R+1$. Each server stores just R bits, and so the storage per server is low. However, their scheme uses an exponential (in R) number of servers, and so has exponential total storage.

Construction 1. *Suppose there are $R+1$ servers, each storing the whole database.*

- *A user who requires Record ℓ creates a $k \times R$ array of bits by drawing its entries α_{ij} uniformly and independently at random.*
- *Server S_{R+1} is requested to return the bit $c_{R+1} = \bigoplus_{i=1}^{k} \bigoplus_{j=1}^{R} \alpha_{ij} X_{ij}$.*
- *For $r = 1, 2, \ldots, R$, server S_r is requested to return the bit $c_r = \bigoplus_{i=1}^{k} \bigoplus_{j=1}^{R} \beta_{ij} X_{ij}$, where*

$$\beta_{ij} = \begin{cases} \alpha_{ij} \oplus 1 & \text{if } i = \ell \text{ and } j = r, \\ \alpha_{i,j} & \text{otherwise.} \end{cases}$$

- *To recover $X_{\ell r}$, namely bit r of record X_ℓ, the user computes $c_r \oplus c_{R+1}$.*

Theorem 3.1. *Construction 1 is a $(R+1)$-server PIR scheme with download complexity $R+1$. The scheme has upload complexity $kR(R+1)$ and total storage $(R+1)Rk$ bits.*

Proof. We note that

$$\alpha_{ij} \oplus \beta_{ij} = \begin{cases} 1 & \text{if } i = \ell \text{ and } j = r, \\ 0 & \text{otherwise.} \end{cases}$$

Hence

$$\begin{aligned} c_r \oplus c_{R+1} &= \bigoplus_{i=1}^{k} \bigoplus_{j=1}^{R} (\alpha_{ij} \oplus \beta_{ij}) X_{ij} \\ &= X_{\ell r}. \end{aligned}$$

So the user recovers the bit $X_{\ell r}$ correctly for any r with $1 \le r \le R$. This proves correctness.

For privacy, we note that S_{R+1} receives a uniformly distributed vector $q_{R+1} = (\alpha_{ij}) \in \{0,1\}^{kR}$ in all circumstances. Since the distribution of q_{R+1} does not depend on ℓ, no information about ℓ is received by S_{R+1}. Similarly, for any $1 \le r \le R$, the query $q_r = (\beta_{ij}) \in \{0,1\}^{kR}$ is uniformly distributed irrespective of the value of ℓ, and so no information about ℓ is received by S_r.

We note that each query q_r is kR bits long (for any $r \in \{1, 2, \ldots, R+1\}$) and so the upload complexity of the scheme is $kR(R+1)$. Each server replies with a single bit, and so the download complexity is $R+1$. The database is kR bits long, and so (since each server stores the whole database) the total storage is $(R+1)Rk$ bits. □

We note that there are situations where one of the servers is asked for an all-zero linear combination of bits from the database. In this case, that server need not reply. So the number of bits of downloaded in Construction 1 is sometimes R (though usually $R+1$ bits are downloaded). See the comment following Corollary 2.6.

We now describe a second construction with improved upload complexity. The construction can be thought of as a variant of Construction 1 where the rows of the array α are all taken from a restricted set $\{e_0, e_1, \ldots, e_R\}$ of size $R+1$. A similar idea is used in the constructions in [38].

For $i = 1, 2, \ldots, R$, let e_i be the i^{th} unit vector of length R. Let e_0 be the all zero vector. For binary vectors $\mathbf{x}$ and $\mathbf{y}$ of length R, write $\mathbf{x} \cdot \mathbf{y}$ be their inner product; so $\mathbf{x} \cdot \mathbf{y} = \oplus_{j=1}^{R} x_j y_j$.

Construction 2. *Suppose there are $R+1$ servers, each storing the whole database.*

- *A user who requires Record ℓ chooses k elements $a_1, a_2, \ldots, a_k \in \mathbb{Z}_{R+1}$ uniformly and independently at random. For $r = 1, \ldots, R+1$, server S_r is sent the vector $q_r = (b_{1r}, b_{2r}, \ldots, b_{kr}) \in \mathbb{Z}_{R+1}^k$, where*

$$b_{ir} = \begin{cases} a_i + r \bmod R+1 & \text{if } i = \ell, \\ a_i & \text{otherwise.} \end{cases}$$

- *Server S_r returns the bit $c_r = \bigoplus_{i=1}^{k} e_{b_{ir}} \cdot X_i$.*
- *To recover the j^{th} bit of X_ℓ, the user finds the integers r and r' such that $b_{\ell r} = 0$ and $b_{\ell r'} = j$. The user then computes $c_r \oplus c_{r'}$.*

Theorem 3.2. *Construction 2 is an $(R+1)$-server PIR scheme with download complexity $R+1$. The scheme has upload complexity $k(R+1)\log(R+1)$ and total storage $(R+1)Rk$ bits.*

Proof. For correctness, we first note that r and r' exist since $b_{\ell r} \in \{0, 1, 2 \ldots, R\}$ takes on each possible value once as $r \in \{0, 1, \ldots, R\}$ varies. Also note that

$$e_{b_{ir}} \oplus e_{b_{ir'}} = \begin{cases} e_j & \text{if } i = \ell, \\ e_0 & \text{otherwise.} \end{cases}$$

So, since $e_0 = 0$,

$$c_r \oplus c_{r'} = \bigoplus_{i=1}^{k} (e_{b_{ir}} \oplus e_{b_{ir'}}) \cdot X_i = e_j \cdot X_\ell = X_{\ell j}.$$

So the user recovers the bit $X_{\ell j}$ correctly for any j with $1 \le j \le R$.

For privacy, we note that S_r receives a uniformly distributed vector $q_r \in (\mathbb{Z}_{R+1})^k$ in all circumstances. Since the distribution of q_r does not depend on ℓ, no information about ℓ is received by S_r.

The calculations of the total storage and download complexity are identical to those in the proof of Theorem 3.1. For the upload complexity, note that it takes just $\log(R+1)$ bits to specify an element of $\mathbb{Z}_{R+1}$. Since each server receives k elements from $\mathbb{Z}_{R+1}$, and since there are $R+1$ servers, the upload complexity of the scheme is $k(R+1)\log(R+1)$ as claimed. □

3.2 Optimal download complexity for a small number of servers

For an integer n such that $(n-1) \mid R$, we now describe an n-server PIR scheme with download complexity $\frac{n}{n-1}R$ bits. By Theorem 2.5, this construction provides schemes with an optimal download complexity for n servers, provided the number k of records is sufficiently large. This construction is closely related to Construction 2 above. Indeed, the construction below is a generalisation of Construction 2 where we work with strings rather than single bits.

We first define an analogue of the bits $e_b \cdot X_i$ computed by servers in Construction 2. We divide an R-bit string X into $n-1$ blocks, each of size $R/(n-1)$. For $b \in \{1, 2, \ldots, n-1\}$ we write $\pi_b(X)$ for the bth block (so $\pi_b(X)$ is an $R/(n-1)$-bit string). We write $\pi_0(X)$ for the all-zero string $0^{R/(n-1)}$ of length $R/(n-1)$.

Construction 3. *Let n be an integer such that $(n-1) \mid R$. Suppose there are n servers, each storing the entire database.*

- *A user who requires Record ℓ chooses k elements $a_1, a_2, \ldots, a_k \in \mathbb{Z}_n$ uniformly and independently at random. For $r = 1, \ldots, n$, server S_r is sent the vector $q_r = (b_{1r}, b_{2r}, \ldots, b_{kr}) \in \mathbb{Z}_n^k$, where*

$$b_{ir} = \begin{cases} a_i + r \bmod n & \text{if } i = \ell, \\ a_i & \text{otherwise.} \end{cases}$$

- *Server S_r returns the $R/(n-1)$-bit string $c_r = \bigoplus_{i=1}^{k} \pi_{b_{ir}}(X_i)$.*

- *To recover the j^{th} block of X_ℓ, the user finds the integers r and r' such that $b_{\ell r} = 0$ and $b_{\ell r'} = j$. The user then computes $c_r \oplus c_{r'}$.*

Theorem 3.3. *Construction 3 is an n-server PIR scheme with download complexity $\frac{n}{n-1}R$. The scheme has upload complexity $nk \log n$ and total storage is nkR.*

Proof. Exactly as in the proof of Theorem 3.2, we first note that r and r' exist since $b_{\ell r} \in \{0, 1, 2 \ldots, n-1\}$ takes on each possible value once as $r \in \{0, 1, \ldots, n\}$ varies. Also note that when $i \neq \ell$

$$\pi_{b_{ir}}(X_i) \oplus \pi_{b_{ir'}}(X_i) = \pi_{a_i}(X_i) \oplus \pi_{a_i}(X_i) = 0^{R/(n-1)},$$

but when $i = \ell$

$$\begin{aligned} \pi_{b_{ir}}(X_i) \oplus \pi_{b_{ir'}}(X_i) &= \pi_0(X_i) \oplus \pi_j(X_i) \\ &= \pi_j(X_i) = \pi_j(X_\ell). \end{aligned}$$

Hence

$$c_r \oplus c_{r'} = \bigoplus_{i=1}^{k} (\pi_{b_{ir}}(X_i) \oplus \pi_{b_{ir'}}(X_i)) = \pi_j(X_\ell).$$

So the user recovers the jth block of X_ℓ correctly for any j with $1 \leq j \leq (n-1)$.

For privacy, we note that S_r receives a uniformly distributed vector $q_r \in (\mathbb{Z}_n)^k$ in all circumstances. Since the distribution of q_r does not depend on ℓ, no information about ℓ is received by S_r.

The total storage is nkR, since each of n servers stores the entire kR-bit database. Each query q_r is $k \log n$ bits long, since an element of $\mathbb{Z}_n$ may be specified using $\log n$ bits. Hence the upload complexity is $nk \log n$. Since each server returns an $R/(n-1)$- bit string, the download complexity is $\frac{n}{n-1}R$. □

Shah et al. [38, Section V] provide PIR schemes with linear (in R) total storage and with download complexity between $2R$ and $4R$. Their scheme requires a number of servers which is independent of R (but is linear in k). The construction above shows that for any fixed positive ϵ a PIR scheme with linear total storage exists with download complexity of $(1+\epsilon)R$ (as we just fix a value of n such that $n/(n-1) < 1+\epsilon$). This is within an arbitrarily close factor of optimality. Moreover, the number of servers in our construction is independent of both k and R. However, note that in our scheme each server stores the whole database, whereas the per server storage of the scheme of Shah et al. is a fixed multiple of R. This issue is addressed in Construction 4 below.

3.3 Schemes with small per-server storage

We make the observation that the last construction may be used to give families of schemes with lower per-server storage; see [38, Section V] for similar techniques. The point here is that we never XOR the first bit (say) from one block with the second bit (say) of any other block, so we can store these bits in separate servers without causing problems.

More precisely, let s be a fixed integer such that $s \mid R$ and let t be a fixed integer such that $(t-1) \mid s$. We divide each record X_i into R/s blocks $\pi_1(X_i), \pi_2(X_i), \ldots, \pi_{R/s}(X_i)$, each s bits long. We then divide each block $\pi_j(X_i)$ into $(t-1)$ sub-blocks $\pi_{j,1}(X_i)$, $\pi_{j,2}(X_i), \ldots, \pi_{j,t-1}(X_i)$, each $s/(t-1)$ bits long. For any $i \in \{1, 2, \ldots, k\}$ and any $j \in \{1, 2, \ldots, R/s\}$, we define $\pi_{j,0}(X_i)$ to be the all zero string $0^{s/(t-1)}$ of length $s/(t-1)$.

Construction 4. *Let s be a fixed integer such that $s \mid R$. Let t be a fixed integer such that $(t-1) \mid s$. Let $n = t(R/s)$. Suppose there are n servers. Each server will store just ks bits.*

- *Index the $t(R/s)$ servers by pairs (u, r), where $1 \leq r \leq t$ and where $1 \leq u \leq R/s$. Server $S_{(u,r)}$ stores the uth sub-block of every block. So $S_{(u,r)}$ stores $\pi_{u,j}(X_i)$ where $1 \leq i \leq k$ and $1 \leq j \leq t-1$. Note that each server stores $k(t-1)s/(t-1) = ks$ bits.*

- *A user who requires Record ℓ chooses k elements $a_1, a_2, \ldots, a_k \in \mathbb{Z}_t$ uniformly and independently at random. The server $S_{(k,r)}$ is*

sent the query $q_r = (b_{1r}, b_{2r}, \ldots, b_{kr}) \in \mathbb{Z}_t^k$, *where*

$$b_{ir} = \begin{cases} a_i + r \bmod n & \text{if } i = \ell, \\ a_i & \text{otherwise.} \end{cases}$$

(Note that many servers receive the same query.)

- *Server* $S_{(u,r)}$ *returns the* $s/(t-1)$*-bit string* $c_{(u,r)} = \oplus_{i=1}^{k} \pi_{u,b_{ir}}(X_i)$.
- *To recover the* j*th sub-block of the* u*th block of* X_ℓ*, the user finds integers* r *and* r' *such that* $b_{\ell r} = 0$ *and* $b_{\ell r'} = j$ *and computes* $c_{(u,r)} \oplus c_{(u,r')}$.

Theorem 3.4. *Construction 4 is a PIR scheme with download complexity* $\frac{R}{s}\frac{r}{t-1}s = \frac{t}{t-1}R$. *The scheme has upload complexity* $nk \log t = (tkR/s)\log t$ *and total storage* $nks = tkR$ *bits.*

Proof. As in the proofs of Theorems 3.2 and 3.3, privacy follows since $S_{u,r}$ always receives a uniformly distributed vector $q_r \in \mathbb{Z}_t^k$ as a query. For correctness, observe that when $i \neq \ell$

$$\begin{aligned} \pi_{u,b_{ir}}(X_i) \oplus \pi_{u,b_{ir'}}(X_i) &= \pi_{u,a_i}(X_i) \oplus \pi_{u,a_i}(X_i) \\ &= 0^{s/(t-1)}, \end{aligned}$$

but when $i = \ell$

$$\begin{aligned} \pi_{u,b_{ir}}(X_i) \oplus \pi_{u,b_{ir'}}(X_i) &= \pi_{u,0}(X_i) \oplus \pi_{u,j}(X_i) \\ &= \pi_{u,j}(X_i) = \pi_{u,j}(X_\ell). \end{aligned}$$

Hence

$$\begin{aligned} c_{(u,r)} \oplus c_{(u,r')} &= \bigoplus_{i=1}^{k} (\pi_{u,b_{ir}}(X_i) \oplus \pi_{u,b_{ir'}}(X_i)) \\ &= \pi_{u,j}(X_\ell). \end{aligned}$$

So the user can indeed compute the j-th sub-block of the u-th block as claimed.

It is easy to calculate the upload complexity, download complexity and total storage complexity as before, remembering that each server stores ks bits rather than the entire database. □

By fixing t and s to be sufficiently large integers, we can see that for all positive ϵ we have a family of schemes with download complexity at most $(1+\epsilon)R$, with total storage linear in the database size, with a linear (in R) number of servers, and where the per server storage is independent of R. So this family of schemes has a better download complexity and per-server storage than Shah et al. [38, Section V], and is comparable in terms of both the number of servers and total storage.

The servers may be divided into t classes $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_t$, where

$$\mathcal{S}_r = \{S_{(1,r)}, S_{(2,r)}, \ldots, S_{(R/s,r)}\}.$$

Since servers in the same class receive the same query, the above construction still works if some of the servers within a class are merged. If this is done, the storage requirements of each merged server is increased, the download complexity and total storage are unaffected, and the number of servers required and upload complexity are reduced. So various trade-offs are possible using this technique.

3.4 An explicit asymptotically optimal scheme

Sun and Jafar [45] describe a PIR scheme that has the best possible asymptotic download complexity, as $R \to \infty$. Their scheme is constructed in a recursive fashion. In this subsection, we describe an explicit, non-recursive, scheme with the same parameters as the Sun and Jafar scheme. Our scheme has the advantages of a more compact description, and (we believe) a proof that is significantly more transparent.

Our scheme is described in detail in Construction 5 below. But, to aid understanding, we first provide an overview of the scheme.

Suppose that n^k divides R. We split an R-bit string X into n^k blocks, each of length R/n^k. For $j \in \{1, 2, \ldots, n^k\}$ we write $\pi_j(X)$ for the j-th block of X, and we write $\pi_0(X)$ for the all zero block $0^{R/n^k}$.

Let $\mathcal{V}$ be the set of all non-zero strings $\mathbf{v} = v_1v_2v_3\dots v_k \in \{0,1,2,\dots,n-1\}^k$ such that $\sum_{i=1}^k v_i \equiv 0 \bmod n-1$. (Note that our sum is taken modulo $n-1$, not modulo n.) Let $\mathcal{W} = \{1,2,\dots,n\} \times \mathcal{V}$. For each record, say Record ℓ, we will define a graph $\Gamma^{[\ell]}$ on the vertex set $\mathcal{W}$ (see below).

There are n servers in the scheme, each storing the whole database. Server S_r receives a query consisting of integers $b_i(r,\mathbf{v}) \in \{1,2,\dots,n^k\}$ where $i \in \{1,2,\dots,k\}$ and $\mathbf{v} \in \mathcal{V}$. The server replies with $|\mathcal{V}|$ strings, each of length R/n^k. Each string is a linear combination of blocks, at most one block from each record (the choice of each block being determined by an integer $b_i(r,\mathbf{v})$: see (4) below). From the perspective of S_r, the distribution of the integers $b_i(r,\mathbf{v})$ does not depend on ℓ, enabling us to attain privacy. However, the user chooses these integers so that $b_i(r,\mathbf{v})$ and $b_i(r',\mathbf{v}')$ are constrained to be equal when $(r,\mathbf{v})$ and $(r',\mathbf{v}')$ lie in the same component of the graph $\Gamma^{[\ell]}$. This is done in such a way that the user can reconstruct Record ℓ from the servers' replies.

We now give details of the scheme. We begin by describing the graph $\Gamma^{[\ell]}$ (see Figure 1) and by detailing some of its structure. Let $\ell \in \{1,2,\dots,k\}$. The graph $\Gamma^{[\ell]}$ is defined on the vertex set $\mathcal{W}$, and is bipartite with parts $\mathcal{W}_1^{[\ell]}$ and $\mathcal{W}_2^{[\ell]}$: the set $\mathcal{W}_1^{[\ell]}$ consists of those elements $(r,\mathbf{v}) \in \mathcal{W}$ such that $v_\ell \neq 0$, and $\mathcal{W}_2^{[\ell]}$ consists of those elements such that $v_\ell = 0$. We draw at most one edge from each element $(r,\mathbf{v}) \in \mathcal{W}_1^{[\ell]}$ into $\mathcal{W}_2^{[\ell]}$ as follows. If v_ℓ is the only non-zero entry in $\mathbf{v}$, we draw no edge from (r,v_ℓ), so we have an isolated vertex. Suppose two or more entries of $\mathbf{v}$ are non-zero. We define $\ell_2 \in \{1,2,\dots,k\}$ to be the next entry in $\mathbf{v}$ after the ℓth that is non-zero, taken cyclically. Let $w \in \{1,2,\dots,n-1\}$ be such that $w \equiv v_\ell + v_{\ell_2} \bmod n-1$. Define $\mathbf{v}' = v_1'v_2'\cdots v_k'$ by

$$v_i' = \begin{cases} v_i & \text{if } i \in \{1,2,\dots,k\} \setminus \{\ell,\ell_2\}, \\ 0 & \text{if } i = \ell, \\ w & \text{if } i = \ell_2. \end{cases}$$

Let $r' \in \{1,2,\dots,n\}$ be such that $r' \equiv r+v_\ell \bmod n$. We join $(r,\mathbf{v})$ to $(r',\mathbf{v}')$.

Let $\mathcal{C}^{[\ell]}$ be the set of connected components of the graph $\Gamma^{[\ell]}$. We note that $\Gamma^{[\ell]}$ has exactly n isolated vertices, namely the vectors of the form $(r,\mathbf{v})$ where $r \in \{1,2,\dots,n\}$ and where $\mathbf{v}$ is the single vector defined by

$$v_i = \begin{cases} 0 & \text{if } i \neq \ell, \\ n-1 & \text{if } i = \ell. \end{cases}$$

The remaining components in $\mathcal{C}^{[\ell]}$ are stars consisting of a central vertex in $\mathcal{W}_2^{[\ell]}$ and $n-1$ other vertices all lying in $\mathcal{W}_1^{[\ell]}$. Moreover, we note that if $(r,\mathbf{v})$ and $(r',\mathbf{v}')$ are distinct vertices in the same component of $\Gamma^{[\ell]}$ then $r \neq r'$.

We claim that the number of vertices $(r,\mathbf{v}) \in \mathcal{W}_1^{[\ell]}$ is n^k. To see this, we note that there are n choices for r, and then n^{k-1} choices for $v_1, v_2, \dots, v_{\ell-1}, v_{\ell+1}, \dots, v_k$. Once these choices are made $v_\ell \in \{0,1,\dots,n-1\}$ is determined, since $v_\ell \neq 0$ and $\sum_{i=1}^k v_i \equiv 0 \bmod n-1$. This establishes our claim.

Since every component of $\Gamma^{[\ell]}$ contains a vertex in $\mathcal{W}_1^{[\ell]}$, we see that $|\mathcal{C}^{[\ell]}| \leq |\mathcal{W}_1^{[\ell]}| = n^k$. Indeed, the number of components of $\Gamma^{[\ell]}$ is:

$$\begin{aligned} |\mathcal{C}^{[\ell]}| &= n + (|\mathcal{W}_1^{[\ell]}| - n)/(n-1) \\ &= n(1 + (n^{k-1}-1)/(n-1)). \end{aligned}$$

Construction 5. *Suppose that $n^k \mid R$. Suppose there are n servers, each storing the whole database.*

- *A user who requires Record ℓ proceeds as as follows. In the notation defined above, for each $i \in \{1,2,\dots,k\} \setminus \{\ell\}$ the user chooses (uniformly and independently) a random injection $f_i : \mathcal{C}^{[\ell]} \to \{1,2,\dots,n^k\}$. The user chooses (again uniformly and independently) a random bijection $\psi : \mathcal{W}_1^{[\ell]} \to \{1,2,\dots,n^k\}$.*

  Define integers $b_i(r,\mathbf{v}) \in \{0,1,\dots,n^k\}$ for $(r,\mathbf{v}) \in \mathcal{W}$ and $i \in \{1,2,\dots,k\}$ as follows.

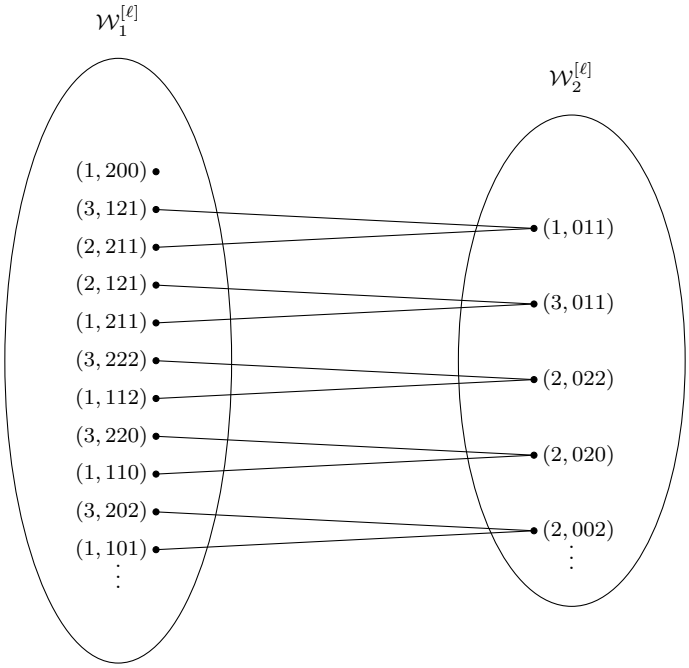

Figure 1: Part of the graph $\Gamma^{[\ell]}$ when $n = k = 3$ and $\ell = 1$.

If $i \neq \ell$, define

$$b_i(r,\mathbf{v}) = \begin{cases} 0 & \text{if } v_i = 0, \\ f_i(C) & \text{if } v_i \neq 0 \\ & \text{and } (r,\mathbf{v}) \text{ lies in the} \\ & \text{component } C \in \mathcal{C}. \end{cases}$$

Note that when $i = \ell$ we have that $v_i \neq 0$ if and only if $(r,\mathbf{v}) \in \mathcal{W}_1^{[\ell]}$. So when $i = \ell$ we may define

$$b_i(r,\mathbf{v}) = \begin{cases} 0 & \text{if } v_i = 0, \\ \psi((r,\mathbf{v})) & \text{if } v_i \neq 0. \end{cases}$$

For $r = 1, 2, \ldots, n$, server S_r is sent the vector $q_r = (b_i(r,\mathbf{v}) : \mathbf{v} \in \mathcal{V}, i \in \{1, 2, \ldots, k\})$.

- *The server S_r replies with the blocks*

$$s_{(r,\mathbf{v})} = \sum_{i=1}^{k} \pi_{b_i(r,\mathbf{v})}(X_i) \tag{4}$$

for all $\mathbf{v} \in \mathcal{V}$.

- *To recover block j of X_ℓ, the user finds $(r,\mathbf{v}) = \psi^{-1}(j) \in \mathcal{W}_1^{[\ell]}$. Let $C \in \mathcal{C}^{[\ell]}$ be the component containing $(r,\mathbf{v})$. If $|C| > 1$, let $(r',\mathbf{v}') \in C \cap \mathcal{W}_2^{[\ell]}$. Then (see below for justification)*

$$\pi_j(X_\ell) = \begin{cases} s_{(r,\mathbf{v})} & \text{if } |C| = 1, \text{ and} \\ s_{(r,\mathbf{v})} \oplus s_{(r',\mathbf{v}')} & \text{if } |C| > 1. \end{cases}$$

Theorem 3.5. *Construction 5 is an n-server PIR scheme with download complexity $(1 - 1/n^k)(n/(n-1))R$. The total storage of the scheme is nkR. The upload complexity of the scheme is $k^2 n^k \log n$ bits.*

Proof. We begin by establishing correctness of the scheme. Let $(r,\mathbf{v}) = \psi^{-1}(j)$ and let $C \in \mathcal{C}^{[\ell]}$ be the component containing $(r,\mathbf{v})$. When $|C| = 1$ we have $v_i \neq 0$ if and only if $i = \ell$ and so

$$s_{(r,\mathbf{v})} = \sum_{i=1}^{k} \pi_{b_i(r,\mathbf{v})}(X_i) = \pi_{b_\ell(r,\mathbf{v})}(X_\ell) = \pi_j(X_\ell),$$

the last equality following since $b_\ell(r, \mathbf{v}) = j$. Hence the user recovers the jth block $\pi_j(X_\ell)$ of X_ℓ correctly in this case. Suppose now that C contains two or more vertices, so there exists $(r', \mathbf{v}') \in C \cap \mathcal{W}_2^{[\ell]}$. When $i \neq \ell$, the values of $b_i(r, \mathbf{v})$ and $b_i(r', \mathbf{v}')$ are equal, since $(r, \mathbf{v})$ and $(r', \mathbf{v}')$ lie in the same component C of $\Gamma^{[\ell]}$ and since $v_i = 0$ if and only if $v'_i = 0$. Moreover, $v_\ell \neq 0$ and $v'_\ell = 0$. Hence

$$\begin{aligned} s_{(r,\mathbf{v})} \oplus s_{(r',\mathbf{v}')} &= \sum_{i=1}^{k} \left(\pi_{b_i(r,\mathbf{v})}(X_i) \oplus \pi_{b_i(r',\mathbf{v}')}(X_i)\right) \\ &= \pi_{b_\ell(r,\mathbf{v})}(X_\ell) \oplus \pi_{b_\ell(r',\mathbf{v}')}(X_\ell)) \\ &= \pi_{\psi((r,\mathbf{v}))}(X_\ell) \oplus \pi_0(X_\ell)) \\ &= \pi_j(X_\ell). \end{aligned}$$

So the user recovers the jth block $\pi_j(X_\ell)$ of X_ℓ correctly in this case also. We have established correctness.

We now aim to establish the security of the scheme. Let $\mathcal{A}$ be the set of integer vectors $(a_i(\mathbf{v}) \in \{0, 1, \ldots, n^k\} : i \in \{1, 2, \ldots, k\}, \mathbf{v} \in \mathcal{V})$ with the restrictions that $a_i(\mathbf{v}) = 0$ if and only if $v_i = 0$, and that for any fixed $i \in \{1, 2, \ldots, k\}$ the integers $a_i(\mathbf{v})$ with $v_i \neq 0$ are distinct. Let $r \in \{1, 2, \ldots, n\}$ be fixed. The query $q_r = (b_i(r, \mathbf{v}) : \mathbf{v} \in \mathcal{V}, i \in \{1, 2, \ldots, k\})$ lies in $\mathcal{A}$, since the functions f_i and ψ are injective and since (whether or not $i = \ell$) we have $b_i(r, \mathbf{v}) = 0$ if and only if $v_i = 0$. Indeed, the query is uniformly distributed in $\mathcal{A}$. To see this, first note that the functions f_i (for $i \neq \ell$) and ψ are chosen independently. The values $b_\ell(r, \mathbf{v})$ for $v_\ell \neq 0$ are uniform subject to being distinct since ψ is a randomly chosen bijection. For $i \neq \ell$, the values $b_i(r, \mathbf{v})$ for $v_\ell \neq 0$ are uniform subject to being distinct, since f_i is a uniformly chosen injection from $\mathcal{C}^{[\ell]}$, and since at most one vertex in any component $C \in \mathcal{C}^{[\ell]}$ has its first entry equal to r. Hence the distribution of query q_r is uniform on $\mathcal{A}$ as claimed. Since this distribution does not depend on ℓ, privacy follows.

Each server replies with $|\mathcal{V}|$ strings, each string of length R/n^k. Since there are n servers, the download complexity is $nR|\mathcal{V}|/n^k$. So it remains to determine $|\mathcal{V}|$. For $0 \leq s \leq k-1$, there are n^{k-s-1} elements $v_1 v_2 \cdots v_k \in \mathcal{V}$ that begin with exactly s zeros, since we may choose $v_{s+2}, v_{s+3}, \ldots, v_k \in \{0, 1, \ldots, n-1\}$ arbitrarily and then v_{s+1} is determined by the fact it is non-zero and $\sum_{j=1}^{k} v_j \equiv 0 \bmod n-1$. So

$$|\mathcal{V}| = \sum_{s=0}^{k-1} n^{k-s-1} = (n^k - 1)/(n-1)$$

and the download complexity is $(1 - 1/n^k)(n/(n-1))R$, as required.

We may argue that the total upload complexity is $k^2 n^k \log n$ as follows. Consider Server S_r. The integers $b_i(r, \mathbf{v})$ with $v_i = 0$ are zero, and so do not need to be sent. There are exactly kn^{k-1} integers $b_i(r, \mathbf{v}) \in \{1, 2, \ldots, n^k\}$ with $i \in \{1, 2, \ldots, k\}$ and $\mathbf{v} \in \mathcal{V}$ with $v_i \neq 0$. (To see this, note that there are k choices for i, and n choices for each component $\mathbf{v}$ except the ℓth. But then v_ℓ is determined by the fact that it is non-zero and $\sum_{j=1}^{k} v_j \equiv 0 \bmod n-1$.) Each integer can be specified using $k \log n$ bits, and so the query q_r is $k^2 n^{k-1} \log n$ bits long. Since there are n servers, the total upload complexity is $k^2 n^k \log n$ bits, as required. □

3.5 An averaging technique

The download complexity of both the PIR scheme due to Sun and Jafar [45] and the scheme in Construction 5 above is $(1 - 1/n^k)(n/(n-1))R$. This is only slightly smaller than the more practical scheme in Construction 3, which has download complexity $(n/(n-1))R$. In fact, the *expected* number of bits downloaded in Construction 3 is $(1-1/n^k)(n/(n-1))R$, since a server is asked for an all-zero linear combination of blocks with probability $1/n^k$ and need not reply in this case. This section describes an 'averaging' technique which transforms Construction 3 into a scheme with good (worst case) download complexity, at the price of a much stronger divisibility constraint on the length of blocks. This technique will work for a wide range of PIR schemes, but in the case of Construction 3 it

produces a scheme with optimal download complexity $(1 - 1/n^k)(n/(n-1))R$. Moreover, the upload complexity is considerably smaller than the schemes described in [45] and Construction 5.

Before giving the detail, we describe the general idea. Chan, Ho and Yamamoto [12, Remark 3], [13, Remark 2] observed that a PIR scheme with good upload complexity (but long record lengths) can be constructed by dividing each record into blocks, then using copies of a fixed PIR scheme for shorter records operating on each block in parallel. Crucially, the same randomness (and so the same queries) can be used for each parallel copy of the scheme, and so upload complexity is low. The 'averaging' construction operates in a similar way. However, rather than using the same randomness we use different but predictably varying randomness for each parallel copy. The server can calculate queries for each copy of the scheme from just one query, so upload complexity remains low. But (because queries vary over all possibilities) the resulting scheme has (worst case) download complexity equal to the average number of bits of download in the Chan, Ho and Yamamoto construction.

In more detail, we modify Construction 3 as follows. Suppose that $n^k(n-1) \mid R$. We divide an R-bit string X into $n^k(n-1)$ blocks, each of size $R/(n^k(n-1))$. We index these blocks by pairs $(b, \mathbf{x})$ where $b \in \{1, 2, \ldots, n-1\} \subseteq \mathbb{Z}_n$ and $\mathbf{x} \in \mathbb{Z}_n^k$. We write $\pi_{(b,\mathbf{x})}(X)$ for the block of X that is indexed by $(b, \mathbf{x})$. For any $\mathbf{x} \in \mathbb{Z}_n^k$, we write $\pi_{(0,\mathbf{x})}(X)$ for the all-zero string $0^{R/(n^k(n-1))}$ of length $R/(n^k(n-1))$.

Construction 6. *Let n be an integer such that $n^k(n-1) \mid R$. Suppose there are n servers, each storing the entire database.*

- *A user who requires Record ℓ chooses k elements $a_1, a_2, \ldots, a_k \in \mathbb{Z}_n$ uniformly and independently at random. For $r = 1, \ldots, n$, server S_r is sent the vector $q_r = (b_{1r}, b_{2r}, \ldots, b_{kr}) \in \mathbb{Z}_n^k$, where*

$$b_{ir} = \begin{cases} a_i + r \bmod n & \text{if } i = \ell, \\ a_i & \text{otherwise.} \end{cases}$$

- *For $r \in \{1, 2, \ldots, n\}$ and $\mathbf{x} \in \mathbb{Z}_n^k$, define the string $c_{(r,\mathbf{x})}$ of length $R/(n^k(n-1))$ by*

$$c_{(r,\mathbf{x})} = \bigoplus_{i=1}^{k} \pi_{(b_{ir}+x_i,\mathbf{x})}(X_i).$$

  The server S_r returns the string $c_{(r,\mathbf{x})}$, for all $\mathbf{x} = (x_1, x_2, \ldots, x_k) \in \mathbb{Z}_n^k$ such that $\mathbf{x} + q_r \neq \mathbf{0}$. So S_r returns $n^k - 1$ strings.

- *To recover the block of X_ℓ indexed by a pair $(j, \mathbf{x})$, the user finds the integers r and r' such that $b_{\ell r} + x_\ell = 0$ and $b_{\ell r'} + x_\ell = j$. The user then computes $c_{(r,\mathbf{x})} \oplus c_{(r',\mathbf{x})}$.*

Theorem 3.6. *Construction 6 is an n-server PIR scheme with download complexity $(1 - 1/n^k)\frac{n}{n-1}R$. The scheme has upload complexity $nk \log n$ and total storage is nkR.*

Proof. We begin with the correctness of the scheme. Exactly as in the proof of Theorem 3.3, we note that r and r' exist since $b_{\ell r} + x_\ell \in \{0, 1, 2 \ldots, n-1\}$ takes on each possible value once as $r \in \{0, 1, \ldots, n\}$ varies. Moreover, we note that the string $c_{(r,\mathbf{x})}$ is all zero if $\mathbf{x} + q_r = 0$ (and similarly the string $c_{(r',\mathbf{x})}$ is all zero if $\mathbf{x} + q_{r'} = 0$) and so the user always receives enough information to calculate $c_{(r,\mathbf{x})} \oplus c_{(r',\mathbf{x})}$.

Let $\mathbf{x} = (x_1, x_2, \ldots, x_k)$. When $i \neq \ell$,

$$\begin{aligned} &\pi_{(b_{ir}+x_i,\mathbf{x})}(X_i) \oplus \pi_{(b_{ir'}+x_i,\mathbf{x})}(X_i) \\ &\qquad = \pi_{(a_i+x_i,\mathbf{x})}(X_i) \oplus \pi_{(a_i+x_i,\mathbf{x})}(X_i) \\ &\qquad = 0^{R/(n-1)}. \end{aligned}$$

When $i = \ell$,

$$\begin{aligned} &\pi_{(b_{ir}+x_i,\mathbf{x})}(X_i) \oplus \pi_{(b_{ir'}+x_i,\mathbf{x})}(X_i) \\ &\qquad = \pi_{(0,\mathbf{x})}(X_i) \oplus \pi_{(j,\mathbf{x})}(X_i) \\ &\qquad = \pi_{(j,\mathbf{x})}(X_i) = \pi_{(j,\mathbf{x})}(X_\ell). \end{aligned}$$

Hence

$$\begin{aligned} c_{(r,\mathbf{x})} \oplus c_{(r',\mathbf{x})} &= \bigoplus_{i=1}^{k} (\pi_{(b_{ir}+x_i,\mathbf{x})}(X_i) \oplus \pi_{(b_{ir'}+x_i,\mathbf{x})}(X_i)) \\ &= \pi_{(j,\mathbf{x})}(X_\ell). \end{aligned}$$

So the user recovers the block of X_ℓ indexed by $(j, \mathbf{x})$ correctly.

Privacy follows from the privacy of Construction 3, as the method for generating queries is identical.

The total storage is nkR, since each of n servers stores the entire kR-bit database. Each query q_r is $k\lceil \log n \rceil$ bits long, since an element of $\mathbb{Z}_n$ may be specified using $\log n$ bits. Hence the upload complexity is $nk \log n$. Since there are n servers, and each server returns $n^k - 1$ strings of length $R/(n^k(n-1))$, the download complexity is $(1 - 1/n^k)\frac{n}{n-1}R$. □

4 Conclusions and future work

In this paper, we have used classical PIR techniques to prove bounds on the download complexity of PIR schemes in modern models, and we have presented various constructions for PIR schemes which are either simpler or perform better than previously known schemes. The characteristics of the six constructions in this paper are summarised in Fig 2, and parameters for the schemes in [38] and [39] are included for comparison.

Various interesting problems remain in this area. We first consider schemes with optimal download complexity:

Question 1. *Are there PIR schemes with fewer than $R + 1$ bits of download complexity?*

Our paper, like the rest of the literature, only considers PIR schemes over binary channels, and in this model the answer is 'no'. But the proofs of this fact in this paper and in Shah at el. [38] both use the fact that we are working over binary channels: more than R bits of download implies that at least $R + 1$ bits are downloaded. So this problem is still open if we extend the model to schemes that do not necessarily use binary channels.

We now return to the standard binary channel model.

Question 2. *Are there PIR schemes with download complexity $R+1$ and total storage linear in R?*

This result was claimed in Shah at el. [38], but we believe that a proof of this is still not known. A proof of this result might depend on a more detailed structural analysis of PIR schemes with $R+1$ bits of download. As a first step, we believe the following to be of interest:

Question 3. *Theorem 2.7 bounds the probability that only R bits are downloaded in a PIR scheme with (worst case) download complexity $R + 1$. Is this bound tight?*

We conjecture that the bound could be significantly improved in some cases.

We now consider families of schemes that have good asymptotic complexity as $R \to \infty$.

Question 4. *Does there exist a family of schemes with download complexity $(1 + o(R))R$ and linear total storage?*

Note that an affirmative solution to Question 2 will imply an affirmative solution to this question.

Question 5. *Are there practical PIR schemes that approach asymptotic capacity as R grows?*

The schemes by Sun and Jafar [39] and the related schemes presented in this paper have the strong restriction that n^k must divide R.

Question 6. *Is there a combinatorial proof that provides a tight upper bound on the asymptotic capacity as $R \to \infty$?*

We comment that the proof in Sun and Jafar [39] uses information theoretic techniques. A

	Download	Upload	Restrictions	Comments
[38]	$R+1$	$R(R+1)$	$n=(R-1)^k$	Algorithm 1 and 2 in [38]
[38]	$R+1$	$k(R+1)\log(R+1)$	$n=R+1$	End of [38, Sec. IV]
[38]	$(2\Delta/(\Delta-(k-1)))R$	$(\Delta^2/(\Delta-(k-1)))R$	$n\geq 2\Delta$, $\Delta\geq 2k$	Algorithm 3 in [38]; linear storage
[39]	$(1-1/n^k)(n/(n-1)R$	$k^2n^k\log n$	$n^k\|R$	Optimal asymptotic download; recursive
1	$R+1$	$kR(R+1)$	$n=R+1$	Generalisation of [14]
2	$R+1$	$k(R+1)\log(R+1)$	$n=R+1$	Similar to [38, Sect. IV]; improved expected download
3	$\frac{n}{n-1}R$	$nk\log n$	$(n-1)\|R$	Optimal download for n servers
4	$\frac{t}{t-1}R$	$nk\log t$	$s\|R$, $(t-1)\|s$, $n=tR/s$	Each server stores only ks bits
5	$(1-1/n^k)(n/(n-1)R$	$k^2n^k\log n$	$n^k\|R$	Optimal asymptotic download; non-recursive
6	$(1-1/n^k)(n/(n-1)R$	$nk\log n$	$n^k(n-1)\|R$	Optimal asymptotic download; improved upload

Figure 2: Summary of the six constructions in this paper and those in [38, 39]

combinatorial proof might give extra structural information for schemes meeting the bound, and might improve the bound in non-asymptotic cases.

Finally, we turn to larger questions. It is clearly very important to construct schemes with practical parameter sizes, which can work in real-life distributed storage settings. In particular, the following problems are key.

Question 7. *Can we find better constructions for PIR schemes?*

Schemes are of interest if they improve per server storage, total storage, upload or download complexity, if the number of servers needed was reduced, or if the divisibility conditions for parameters such as R are weakened.

Question 8. *Can the techniques from this paper be applied to establish bounds or give constructions in other models, such as those discussed in Subsection 1.3?*

In particular, can these constructions be adapted to work when the database is coded (in order to provide robustness against server failure, for example)?

Acknowledgement The authors would like to thank Doug Stinson for comments on an earlier draft.

References

[1] H. Asi and E. Yaakobi, *Nearly optimal constructions of PIR and batch codes, Proc. of IEEE Int. Symp. on Inform. Theory (ISIT)*, pp. 151–155, Aachen, Germany, June 2017.

[2] H. Asi and E. Yaakobi, *Nearly optimal constructions of PIR and batch codes, IEEE Trans. Information Theory*, 65 (2019), 947–964.

[3] D. Augot, F. Levy-Dit-Vahel, and A. Shikfa, *A storage-efficient and robust private information retrieval scheme allowing few servers,* in *Cryptology and Network Security,* pp. 222–239, Springer 2014.

[4] K. Banawan and S. Ulukus, *Multi-message private information retrieval: capacity results and near-optimal schemes, Proc. of IEEE Int. Symp. on Inform. Theory (ISIT)*, pp. 1898–1902, Aachen, Germany, June 2017.

[5] K. Banawan and S. Ulukus, *The capacity of private information retrieval from coded databases, IEEE Trans. Information Theory*, 64 (2018), 1945–1956.

[6] K. Banawan and S. Ulukus, *Multi-message private information retrieval: capacity results and near-optimal schemes, IEEE Trans. Information Theory*, 64 (2018), 6842–6862.

[7] K. Banawan and S. Ulukus, *The capacity of private information retrieval from byzantine and colluding databases, IEEE Trans. Information Theory*, 65 (2019), 1206–1219.

[8] Z. Bar-Yossef, Y. Birk, T. S. Jayram, and T. Kol, *Index coding with side information, IEEE Trans. Information Theory*, 57 (2011), 1479–1494.

[9] S. R. Blackburn and T. Etzion, *PIR array codes with optimal virtual server rate, IEEE Trans. Information Theory*, to appear.

[10] S. R. Blackburn and T. Etzion, *PIR array codes with optimal PIR rate, Proc. of IEEE Int. Symp. on Information Theory (ISIT)*, pp. 2658–2662, Aachen, Germany, June 2017.

[11] S. R. Blackburn, T. Etzion, and M.B. Paterson, *PIR schemes with small download complexity and low storage requirements, Proc. of IEEE Int. Symp. on Information Theory (ISIT)*, pp. 146–150, Aachen, Germany, June 2017.

[12] T. H. Chan, S. Ho, and H. Yamamoto, *Private information retrieval for coded storage, arxiv.org/abs/1410.5489*, October 2014.

[13] T. H. Chan, S. Ho, and H. Yamamoto, *Private information retrieval for coded storage, Proc. of IEEE Int. Symp. on Inform. Theory (ISIT)*, pp. 2842–2846, Hong Kong, June 2015.

[14] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, *Private information retrieval, Journal ACM*, 45 (1998), 965–981.

[15] A. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, *Network coding for distributed storage systems, IEEE Trans. Information Theory*, 56 (2010), 4539–4551.

[16] A. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, *A survey on network codes for distributed storage, Proc. of the IEEE*, 56 (2011), 476–489.

[17] G. Fanti and K. Ramchandran, *Multi-server private information retrieval over unsynchronized databases, Fifty-second Annual Allerton Conference*, pp. 437–444, Illinois, October 2014.

[18] G. Fanti and K. Ramchandran, *Efficient private information retrieval over unsynchronized databases, IEEE J. on Selected Topics in Signal Processing*, 9 (2015), 1229–1239.

[19] A. Fazeli, A. Vardy, and E. Yaakobi, *Coded for distributed PIR with low storage overhead, Proc. of IEEE Int. Symp. on Inform. Theory (ISIT)*, pp. 2852–2856, Hong Kong, June 2015.

[20] A. Fazeli, A. Vardy, and E. Yaakobi, *Private information retrieval without storage overhead: coding instead of replication, arxiv.org/abs/1505.06241*, May 2015.

[21] S. L. Frank-Fischer, V. Guruswami, and M. Wootters, *Locality via partially lifted codes, arxiv.org/abs/1704.08627*, April 2017.

[22] R. Freij-Hollanti, O. Gnilke, C. Hollanti, and D. Karpuk, *Private information retrieval from coded databases with colluding servers, SIAM J. Appl. Algebra Geometry*, 1 (2017), 647–664.

[23] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, *On the locality of codeword symbol, IEEE Trans. Information Theory*, 58 (2012), 6925–6934.

[24] H. T. Gupta, N. Crooks, W. Mulhern, S. Setty, L. Alvisi, and M. Walfish, *Scalable and private media consumption with Popcorn, Proc. 13th USENIX Symposium on Networked Systems Design and*

Implementation, Santa Clara, CA, March 2016.

[25] P. Huang, E. Yaakobi, B. Uchikawa, and P. H. Siegel, *Linear locally repairable codes with availability*, *Proc. of IEEE Int. Symp. on Inform. Theory (ISIT)*, pp. 1871–1875, Hong Kong, June 2015.

[26] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai, *Batch codes and their applications*, *Proc. 36-th ACM Symp. Theory Computing*, pp. 262–271, Chicago, IL, June 2004.

[27] S. Kadhe, B. Garcia, A. Heidarzadeh, S. El Rouayheb, and A. Sprintson, *Private information retrieval with side information*, *arxiv.org/abs/1709.00112*, September 2017.

[28] M. Karmoose, L. Song, M. Cardone, and C. Fragouli, *Private broadcasting: an index coding approach*, *Proc. of IEEE Int. Symp. on Information Theory (ISIT)*, pp. 2543–2547, Aachen, Germany, June 2017. Full version: *arxiv.org/abs/1701.04958*, January 2017.

[29] S. Kumar, E. Rosnes, and A. G. Amat, *Private information retrieval in distributed storage systems using an arbitrary linear code*, *Proc. of IEEE Int. Symp. on Information Theory (ISIT)*, pp. 1421–1425, Aachen, Germany, June 2017.

[30] E. Kushilevitz and R. Ostrovsky, *Replication is not needed: Single database, computationally-private information retrieval*, *Proc. 38-th IEEE Symp. Foundations Computer Science*, pp. 364–373, Miami Beach, FL, October 1997.

[31] H.-Y. Lin and E. Rosnes, *Lengthening and extending binary private information retrieval codes*, *International Zurich Seminar on Information and Communication (IZS)*, pp. 113–117, Zurich, Switzerland, February 2018. Full version: *arxiv.org/abs/arXiv:1707.03495*, July 2017.

[32] S. Rao and A. Vardy, *Lower bound on the redundancy of PIR codes*, *arxiv.org/abs/1605.01869*, May 2016.

[33] A. S. Rawat, D. S. Papailiopoulos, A. G. Dimakis, and S. Vishwanath, *Locality and availability in distributed storage*, *Proc. of IEEE Int. Symp. on Inform. Theory (ISIT)*, pp. 631–635, Honolulu, HI, June 2014.

[34] A. Rawat, D. S. Papailiopoulos, A. G. Dimakis, and S. Vishwanath, *Locality and availability in distributed storage*, *IEEE Trans. Information Theory*, 62 (2016), 4481–4493.

[35] A. S. Rawat, Z. Song, A. G. Dimakis, and A. Gaal, *Batch codes through dense graphs without short cycles*, *IEEE Trans. Information Theory*, 62 (2016), 1592–1604.

[36] N. Silberstein, T.Etzion, and M. Schwartz, *Locality and availability of array codes constructed from subspaces*, *Proc. of IEEE Int. Symp. on Information Theory (ISIT)*, pp. 829–833, Aachen, Germany, June 2017.

[37] N. Silberstein, T.Etzion, and M. Schwartz, *Locality and availability of array codes constructed from subspaces*, *IEEE Trans. Information Theory*, to appear

[38] N.B. Shah, K.V. Rashmi and K. Ramchandran, *One extra bit of download ensures perfectly private information retrieval*, in *Proc. of IEEE Int. Symp. on Inform. Theory (ISIT)*, pp. 856–860, Honolulu, June 2014.

[39] H. Sun and A. Jafar, *The capacity of private information retrieval*, *IEEE Trans. Information Theory*, 63 (2017), 4075–4088.

[40] H. Sun and A. Jafar, *Private information retrieval from MDS coded data with colluding servers: settling a conjecture by Freij-Hollanti et al., Proc. of IEEE Int. Symp. on Information Theory (ISIT)*, pp. 1893–1897, Aachen, Germany, June 2017.

[41] H. Sun and A. Jafar, *Optimal download cost of private information retrieval for arbitrary message length, IEEE Trans. Information Forensics and Security*, 12 (2017), 2920–2932.

[42] H. Sun and A. Jafar, *Private information retrieval from MDS coded data with colluding servers: settling a conjecture by Freij-Hollanti et al., IEEE Trans. Information Theory*, 64 (2018), 1000–1022.

[43] H. Sun and A. Jafar, *The capacity of robust private information retrieval with colluding databases, IEEE Trans. Information Theory*, 64 (2018), 2361–2370.

[44] H. Sun and A. Jafar, *Multiround private information retrieval: capacity and storage overhead, IEEE Trans. Information Theory*, 64 (2018), 5743–5754.

[45] H. Sun and A. Jafar, *The capacity of symmetric private information retrieval, IEEE Trans. Information Theory*, 65 (2019), 322–329.

[46] R. Tajeddine and S. El Rouayheb, *Private information retrieval from MDS coded data in distributed storage systems, Proc. of IEEE Int. Symp. on Information Theory (ISIT)*, pp. 1411–1415, Barcelona, Spain, July 2016.

[47] R. Tajeddine and S. El Rouayheb, *Robust private information retrieval on coded data, Proc. of IEEE Int. Symp. on Information Theory (ISIT)*, pp. 1903–1907, Aachen, Germany, June 2017.

[48] R. Tajeddine, O. W. Gnilke, D. Karpuk, R. Freij-Hollanti, C. Hollanti, and S. El Rouayheb, *Private information retrieval schemes for coded data with arbitrary collusion patterns, Proc. of IEEE Int. Symp. on Information Theory (ISIT)*, pp. 1908–1912, Aachen, Germany, June 2017.

[49] R. Tandon, *The capacity of cache aided private information retrieval, Fifty-fifth Annual Allerton Conference*, pp. 1078–1082, Illinois, October 2017. Full version: *arxiv.org/abs/arXiv:1706.07035*, June 2017.

[50] M. Vajha, V. Ramkumar, and P. V. Kumar, *Binary, shortened projective Reed Muller codes for coded private information retrieval, Proc. of IEEE Int. Symp. on Information Theory (ISIT)*, pp. 2648–2652, Aachen, Germany, June 2017. Full version: *arxiv.org/abs/arXiv:1702.05074*, February 2017.

[51] Q. Wang and M. Skoglund, *Symmetric private information retrieval for MDS coded distributed storage, IEEE International Conference on Communications (ICC)*, Paris, France, May 2017.

[52] Q. Wang and M. Skoglund, *Secure symmetric private information retrieval from colluding databases with adversaries, Fifty-fifth Annual Allerton Conference*, pp. 1083–1090, Illinois, October 2017.

[53] Q. Wang and M. Skoglund, *Linear symmetric private information retrieval for MDS coded distributed storage with colluding servers, IEEE Information Theory Workshop (ITW)*, pp. 71–75, Kaohsiung, Taiwan, November 2018. *arxiv.org/abs/1708.05673*, August 2017.

[54] F. Wang, C. Yun, S. Goldwasser, V. Vaikuntanathan, and M. Zaharia,

Splinter: practical private queries on public data, Proc. 13th USENIX Symposium on Networked Systems Design and Implementation, Boston, MA, March 2017.

[55] S. Yekhanin, *Private information retrieval, Comm. ACM*, 53 (2010), 68–73.

[56] Y. Zhang and G. Ge, *A general private information retrieval from MDS coded databases with colluding servers, Designs, Codes and Cryptography*, to appear.

[57] Y. Zhang and G. Ge, *Multi-file private information retrieval scheme for MDS coded databases with colluding servers, arxiv.org/abs/1704.06785*, May 2017.

[58] Y. Zhang, X. Wang, N. Wei, and G. Ge, *On private information retrieval array codes, IEEE Trans. Information Theory*, 65 (2019), 5565–5573.

[59] Z. Zhang and J. Xu, *Private multi-file retrieval from distributed databases, arxiv.org/abs/1704.00250*, April 2017.