Applicable Algebra in Engineering, Communication and Computing (2023) 34:335–358
https://doi.org/10.1007/s00200-021-00508-5

ORIGINAL PAPER

A survey on single server private information retrieval in a coding theory perspective

Gianira N. Alfarano[1] · **Karan Khathuria**[1] · **Violetta Weger**[1]

Received: 12 December 2020 / Accepted: 27 March 2021 / Published online: 12 April 2021

Abstract
In this paper, we present a new perspective of single server private information retrieval (PIR) schemes by using the notion of linear error-correcting codes. Many of the known single server schemes are based on taking linear combinations between database elements and the query elements. Using the theory of linear codes, we develop a generic framework that formalizes all such PIR schemes. This generic framework provides an appropriate setup to analyze the security of such PIR schemes. In fact, we describe some known PIR schemes with respect to this code-based framework, and present the weaknesses of the broken PIR schemes in a unified point of view.

Keywords Computational PIR · Coding theory · Single server PIR

Mathematics Subject Classification 68P20 · 94B99

1 Introduction

Private information retrieval (PIR) was first introduced in [1] to cope with the following problem: retrieving an element from a database, without revealing to the untrusted source managing the database any information about that element. Since its introduction, it has attracted many researchers and several works have addressed their focus on it. There have been proposed two solutions to this problem, namely, the *information theoretical* one and the *computational* one. The first one aims to

✉ Gianira N. Alfarano
gianiranicoletta.alfarano@math.uzh.ch

Karan Khathuria
karan.khathuria@math.uzh.ch

Violetta Weger
violetta.weger@math.uzh.ch

[1] Institute of Mathematics, University of Zurich, Winterthurerstrasse 190, 8057 Zurich, Switzerland

guarantee that the server gets no information about the file that the user wants to retrieve. Solutions for multiple servers were presented in [2–7]. In the case of a single server, the trivial solution, i.e., downloading the whole database, is the only possibility to ensure information theoretical privacy. However, the trivial solution is not satisfactory as it comes with a very large communication cost, which basically implies that it is impractical. On the contrary, in computational PIR, the privacy is guaranteed assuming that the server has limited computational power. Hence, the computational PIR (cPIR) is the only practical solution in case of a single server.

Most of the early cPIR schemes are based on the difficulty of number-theoretical problems, such as integer factorization (see for example [8–11]). The known (non-trivial) single server cPIR constructions require to perform some cryptographic operations on each database element, which increase the computational cost of these schemes in comparison to the information theoretical ones. In [12], Sion and Carbunar showed that the number-theoretical PIR schemes are not practical, and computing a PIR reply is always less efficient than sending the whole database. Moreover, such schemes, based on factoring an integer, will be insecure in the era of quantum computers [13].

Some recent constructions of PIR schemes use a fully homomorphic encryption (FHE) scheme. Yi et al. presented in [14] a generic way to construct a PIR from an FHE. Following this construction many PIR protocols have been proposed using FHE schemes based on problems in lattices and learning with error (LWE) problems [15–18]. Recently, Aguilar-Melchor et al. presented in [16] XPIR, a PIR construction using a Ring-LWE based FHE scheme, that is computationally efficient but comes with a large communication cost. Following [16], Angel et al. in [17] were able to significantly improve its communication cost with only slightly more computations compared to XPIR. Along with the scheme of Angel et al., the recent work of Ali et al. [18] represent the state-of-the-art efficiency for PIR schemes.

Recently, Holzbaur, Hollanti and Wachter-Zeh have proposed in [19] the first single server PIR based on coding theory. However, their proposal was attacked in [20]. The primary idea in [19] is to generate the query by hiding carefully chosen error vectors using codewords from a random linear code. The linear code is kept secret by the user in order to obtain privacy. The same idea was previously used by Aguilar-Melchor and Gaborit in a lattice-based PIR scheme [21], without using the notion of linear codes. The scheme was later attacked by Liu and Bi [22] using lattice reduction algorithms.

Interestingly, the idea of hiding query information using linear codes can be observed, directly or indirectly, in several other PIR schemes. In this paper, we develop a unified framework that describes all such PIR schemes. In particular, this framework characterizes all the single server PIR schemes that generate replies by contracting the database elements and the query elements using linear combinations. The main aim of this paper is to provide a survey on several existing single server PIR schemes in a unified coding theoretic perspective. This allows a deeper theoretical insight on the security of these PIR schemes.

The framework is based on two key elements: a linear code that hides the query information, and a retrieval function that allows the user to retrieve the desired file from a linearly entangled reply. On one hand, the notion of linear codes describes

the common features of several existing PIR schemes, and on the other hand, the retrieval function describes the key differences between the schemes. In terms of the framework, the privacy of a PIR scheme heavily relies on the retrieval function. We observe that several choices of retrieval functions are not safe to use, for example, finite field homomorphisms and vector space homomorphisms. Moreover, we discuss the weaknesses of many broken PIR schemes with respect to this code-based framework.

The paper is organized as follows: in Sect. 2, we introduce the notation that will be used throughout the paper and give the background on single server private information retrieval, and linear codes over finite fields and over rings. In Sect. 3, we present the code-based framework and discuss the security in a general point of view. In Sect. 4, we provide a survey on four different PIR schemes, described in terms of the code-based framework. The first one is a basic scheme that uses a finite field homomorphism as the retrieval function. Whereas the other three are based on the existing PIR schemes [19, 21] and [16], respectively. The latter is the only example of a scheme that we present which is still unbroken. For the former two we will also describe the existing attacks with respect to the proposed code-based framework. Finally, in Sect. 5, we draw some theoretical remarks on the generality of the framework, and on the security of single server PIR schemes.

2 Preliminaries

In this section we introduce the notation that we use in the paper and we recall some background on the theory of single server PIR. Moreover, we introduce the basic notions of error-correcting linear codes.

2.1 Notation

In this paper, we denote by $\mathcal{R}$ a ring and by $\mathcal{R}^\times$ the set of invertible elements in the ring $\mathcal{R}$. Moreover, let q be a prime power, then we denote by $\mathbb{F}_q$ the finite field of size q.

We use bold lower case, respectively bold upper case letters to denote row vectors, respectively matrices. When we consider column vectors, we use the transpose symbol. The identity matrix of size k is denoted by $\mathbf{I}_k$. Given a vector $\mathbf{x}$ of length n and a set $S \subset \{1, \dots, n\}$, we denote by $\mathbf{x}_S$ the projection of $\mathbf{x}$ on the coordinates indexed by S. In the same way, $\mathbf{M}_S$ denotes the projection of the $k \times n$ matrix $\mathbf{M}$ to the columns indexed by S.

For a set S we denote by S^C its complement. The support of a vector $\mathbf{x} \in \mathbb{F}_q^n$ is denoted by $\mathrm{Supp}(\mathbf{x}) = \{1 \leq i \leq n \mid x_i \neq 0\}$.

The i-th entry of a vector $\mathbf{x} \in \mathbb{F}_q^n$ is denoted by $\mathbf{x}[i]$, for $i \in \{1, \dots, n\}$.

Given a set S and a distribution χ on S, $x \leftarrow \chi$ represents a sample x from S following the distribution χ.

2.2 Single server private information retrieval

A single server PIR is a scheme involving two parties, the *user* and the *server*. The server manages a database containing some public information, and the user is interested in retrieving some entries of the database, without revealing which item was queried.

2.2.1 Basic description

A basic description of a single server PIR scheme is as follows. Let the database be denoted by $\mathcal{DB} = \{db_1, \dots, db_N\}$, containing N files, and suppose the user wishes to retrieve the i-th file db_i. The user first constructs a *query* $Q = \{q_1, \dots, q_N\}$, which hides the information about the index i, and sends it to the server. The server computes a *response* by performing certain operations between q_j and db_j for each j, and returns it to the user. The scheme is said to be *correct* if the user can retrieve the desired file db_i from the response.

2.2.2 Communication and computational cost

A simple solution to preserve the privacy is downloading the whole database. However, the communication cost of this operation, measured as the total number of bits exchanged by user and server, in the trivial case is too high, namely $\mathcal{O}(N)$ where N is the size of the database. Modern PIR protocols allow the user to retrieve data from the database, with a communication complexity much smaller than $\mathcal{O}(N)$. Some common methods can be used to improve the communication cost of any PIR scheme. In Sect. 3.2, we discuss such techniques in detail.

Another important aspect of a single server PIR scheme is the computational cost. Since the database has to process each entry of the query, the schemes are computationally expensive.

2.3 Linear codes

2.3.1 Over finite fields

Let $\mathbf{x}$ be a vector in $\mathbb{F}_q^n$. The *Hamming weight* of $\mathbf{x}$ is denoted by $\mathrm{wt}(\mathbf{x})$ and it is defined as the number of its nonzero entries, i.e., it is the size of its support. The *Hamming distance* between two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$ is defined as the number of components in which the two vectors differ, i.e., $d(\mathbf{x}, \mathbf{y}) = |\{i \mid x_i \neq y_i\}|$.

An $[n, k]_q$ *linear code* $\mathcal{C}$ is a k-dimensional subspace of $\mathbb{F}_q^n$ endowed with the Hamming distance and the elements of $\mathcal{C}$ are called *codewords*.

The *minimum distance* d of $\mathcal{C}$ is the quantity

$$d := \min\{d(\mathbf{x}, \mathbf{y}) \mid \mathbf{x}, \mathbf{y} \in \mathcal{C}, \mathbf{x} \neq \mathbf{y}\}.$$

When the minimum distance d of a linear code $\mathcal{C}$ is known, then $\mathcal{C}$ is denoted by $[n, k, d]_q$.

A matrix $\mathbf{G} \in \mathbb{F}_q^{k \times n}$ whose rows form a basis for $\mathcal{C}$ is called *generator matrix* of $\mathcal{C}$. Hence, we can define the code $\mathcal{C}$ as $\{\mathbf{v} \in \mathbb{F}_q^n \mid \mathbf{v} = \mathbf{u}\mathbf{G}^\top, \mathbf{u} \in \mathbb{F}_q^k\}$. Similarly, we can define the code $\mathcal{C}$ as the kernel of a matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, i.e. $\mathcal{C} := \ker(\mathbf{H}) = \{\mathbf{v} \in \mathbb{F}_q^n \mid \mathbf{H}\mathbf{v}^\top = \mathbf{0}\}$. Such a matrix is called *parity-check matrix* for the code $\mathcal{C}$. An information set of an $[n, k, d]_q$ code $\mathcal{C}$ is a set $I \subset \{1, \ldots, n\}$ of size k, such that $\mid \mathcal{C} \mid = \mid \mathcal{C}_I \mid$, where $\mathcal{C}_I$ denotes the restriction of all codewords to the entries indexed by I.

2.3.2 Over rings

Let $\mathcal{R}$ be a commutative ring with identity. A *linear code* $\mathcal{C}$ of length n over $\mathcal{R}$ is an $\mathcal{R}$-module in the space $\mathcal{R}^n$.

A linear code $\mathcal{C}$ of length n over $\mathcal{R}$ is called *cyclic* if $\mathbf{c} = (c_1, \ldots, c_n) \in \mathcal{C}$ implies $(c_n, c_1, \ldots, c_{n-1}) \in \mathcal{C}$. Equivalently, $\mathcal{C}$ is an ideal of the ring $\mathcal{R}[x]/(x^n - 1)$.

A linear code $\mathcal{C}$ of length n over $\mathcal{R}$ is called *negacyclic* if $\mathcal{C}$ is an ideal of the ring $\mathcal{R}[x]/(x^n + 1)$.

3 Code-based framework

In this section, we present a generic framework for single server PIR schemes by using the notion of error-correcting codes. For simplicity, we present the framework using a simple database setup, later we discuss different kinds of database setups that can be used to improve the communication complexity.

3.1 Code-based framework

Before we describe the framework in detail, we highlight some elements that are used in the framework:

- We describe the generic framework over a finite commutative ring $\mathcal{R}$ using a retrieval function $f : \mathcal{R} \to \mathcal{R}$ and three subsets X, Y, Z of $\mathcal{R}$.
- The database files belong to the set X.
- In order to generate queries, we fix a randomly chosen linear code $\mathcal{C}$ over $\mathcal{R}$. Each element of the query is the sum of a randomly chosen codeword in $\mathcal{C}$ and an error vector over $\mathcal{R}$.
- To generate the error vectors corresponding to the non-desired files we use the set Y, whereas for the desired file we use the set Z.

Setup:

We define a retrieval function $f : \mathcal{R} \to \mathcal{R}$, and subsets $X, Y, Z \subseteq \mathcal{R}$ satisfying:

1. f is a non-zero map.

2. $Y \subseteq \ker(f) := \{x \in \mathcal{R} : f(x) = 0\}$ such that any linear combination of elements in Y with scalars in X belongs to $\ker(f)$, i.e., $x_1y_1 + x_2y_2 + \cdots + x_jy_j \in \ker(f)$ whenever $x_1, \ldots, x_j \in X$ and $y_1, \ldots, y_j \in Y$.
3. $Z \subseteq f^{-1}(\mathcal{R}^\times)$ such that $f(y + xz) = xf(z)$ for all $y \in \ker(f), x \in X$ and $z \in Z$.

Note that f does not need to be a ring homomorphism, it can be any kind of function from $\mathcal{R}$ to $\mathcal{R}$ satisfying the above three conditions.

Let $\mathbf{M} = (m_i) \in X^N$ represent the database, i.e., there are N files in the database. Suppose that the user wants to retrieve the b-th file from the database.

Let $\mathcal{C}$ be a random linear code over $\mathcal{R}$ of length n, i.e., $\mathcal{C}$ is an $\mathcal{R}$-submodule of $\mathcal{R}^n$.

Query generation:

Let $\mathbf{g}_1, \ldots, \mathbf{g}_m$ be generators of $\mathcal{C}$ as an $\mathcal{R}$-module, and let $\mathrm{Enc} : \mathcal{R}^m \to \mathcal{R}^n$ be an encoding map of $\mathcal{C}$. Note that Enc is an $\mathcal{R}$-linear map given by $(a_1, \ldots, a_m) \mapsto a_1\mathbf{g}_1 + \cdots + a_m\mathbf{g}_m$.

Let $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_N$ be randomly chosen elements in $\mathcal{R}^m$, and define $\mathbf{c}_i = \mathrm{Enc}(\mathbf{a}_i)$ for all $i \in \{1, \ldots, N\}$.

Now, let v be a randomly chosen fixed element in $\{1, \ldots, n\}$ and we randomly choose error vectors $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_N$ in $\mathcal{R}^n$, such that they satisfy the following conditions that allow the reply extraction:

$$\mathbf{e}_b[v] \in Z \quad \text{and} \quad \mathbf{e}_i[v] \in Y \text{ for all } i \neq b.$$

Let $\mathbf{q}_i := (\mathbf{a}_i, \mathbf{c}_i + \mathbf{e}_i)$ for all $i \in \{1, \ldots, N\}$. The query is then given by

$$Q := \{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_N\}.$$

Reply generation: The response is generated by computing

$$\mathbf{r} = \sum_{i=1}^{N} m_i\mathbf{q}_i = \sum_{i=1}^{N} (m_i\mathbf{a}_i, m_i(\mathbf{c}_i + \mathbf{e}_i)) =: (\mathbf{r}_1, \mathbf{r}_2).$$

Reply extraction: First we perform the decoding by applying the encoding map on $\mathbf{r}_1$, and obtain:

$$\mathbf{r}_2 - \mathrm{Enc}(\mathbf{r}_1) = \sum_{i=1}^{N} m_i\mathbf{e}_i.$$

After that we can use the retrieval function f on the v-th coordinate,

$$\begin{aligned} f\left(\sum_{i=1}^{N} m_i\mathbf{e}_i[v]\right) &= f\left(\sum_{i\neq b} m_i\mathbf{e}_i[v]\right) + f(m_b\mathbf{e}_b[v]) \\ &= m_b f(\mathbf{e}_b[v]). \end{aligned}$$

The above equalities follow from the conditions of the retrieval function. Now, since we know $f(\mathbf{e}_b[v])$ and we have that $f(\mathbf{e}_b[v]) \in f(Z) \subseteq \mathcal{R}^\times$, we can retrieve the desired file m_b.

3.2 Communication complexity and different database setups

With respect to the basic description of the code-based framework, the communication cost is more than the size of the whole database. Indeed, for each file which is an element in $\mathcal{R}$, we are sending a query element in $\mathcal{R}^{n+m}$. Thus the total communication cost is $(N+1)$ times the size of an element in $\mathcal{R}^{m+n}$. We can improve the communication complexity by using a matrix database setup [1] or iterative response techniques:

- *Matrix setup of database*: In order to reduce the communication complexity, one can see the database as an $s \times t$ matrix, where each element of the matrix is a database file. Now, the user generates a *query* $Q = \{\mathbf{q}_1, \ldots, \mathbf{q}_t\}$ containing t elements. For each query, the server replies by sending back the *response* $R = \{\mathbf{r}_1, \ldots, \mathbf{r}_s\}$, which contains s responses corresponding to the s rows of the database matrix. This technique was introduced in [1]. Using this approach and assuming $s = t = \sqrt{N}$, the communication complexity is $2\sqrt{N}$ times the size of an element in $\mathcal{R}^{m+n}$.
- *Iterative reply generation:* In this technique, one splits each file into L parts and repeats the query to retrieve each part of the file. Since the query is generated in order to retrieve only small portions of the desired file, the size of the ambient space reduces accordingly. Hence, relative to the size of the database, the query size reduces by a factor of L, and the response size increases by the same factor.

3.3 Security

The security of a single server computational PIR scheme is based on the difficulty of identifying the index of the desired file by looking at the query. With respect to the code-based framework, we can describe the security using the following distinguishability problem.

Problem 1 (Distinguishability Problem) Consider the notations of the setup and the query generation process of the code-based framework. Given the query vectors $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_N$, determine the index b of the desired file.

The difficulty of solving the distinguishability problem depends highly on the choice of the retrieval function f. In the following, we present two generic strategies that can be used to solve this problem. However, the computational cost of these strategies directly relies on the choice of the retrieval function and the error vectors $\mathbf{e}_1, \ldots, \mathbf{e}_N$.

1. Consider the following matrix consisting of the query vectors

$$\mathbf{A} = \begin{pmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \vdots \\ \mathbf{q}_N \end{pmatrix} = \begin{pmatrix} \mathbf{a}_1 & \mathbf{c}_1 + \mathbf{e}_1 \\ \mathbf{a}_2 & \mathbf{c}_2 + \mathbf{e}_2 \\ \vdots & \\ \mathbf{a}_N & \mathbf{c}_N + \mathbf{e}_N \end{pmatrix} \in \mathcal{R}^{N \times (m+n)}.$$

Observe that the vectors $\big(\mathbf{e}_1[j], \mathbf{e}_2[j], \dots, \mathbf{e}_N[j]\big)$ for all $j \in \{1, \dots, n\}$ belong to the column span of $\mathbf{A}$. We recall that the v-th coordinate of the error vectors are chosen in a special way, i.e., $\mathbf{e}_b[v] \in Z \subseteq f^{-1}(\mathcal{R}^\times)$ and $\mathbf{e}_i[v] \in Y \subseteq \ker(f)$ for all $i \neq b$. Hence, one could solve Problem 1 by finding the vector $\big(\mathbf{e}_1[v], \mathbf{e}_2[v], \dots, \mathbf{e}_N[v]\big)$ in the column span of $\mathbf{A}$.

2. Let $\mathbf{A}$ be the query matrix as defined above. For each $i \in \{1, \dots, N\}$, let $\mathbf{A}_i$ be the submatrix of $\mathbf{A}$ obtained by deleting the i-th row. Clearly, by construction, $\mathbf{A}_b$ has distinct properties compared to $\mathbf{A}_i$ for any $i \neq b$. Thus, if there exists an (algebraic or non-algebraic) invariant that can distinguish $\mathbf{A}_b$ from $\mathbf{A}_i$ for any $i \neq b$, then Problem 1 can be solved by computing this invariant for each $\mathbf{A}_1, \dots, \mathbf{A}_N$.

4 Examples of different PIR's in our framework

In this section, we discuss several examples of single server PIR schemes that are based on different kinds of retrieval function. In each case, we analyze the security with respect to the distinguishability problem. In Table 1, we summarize all the differences among the schemes.

4.1 Basic PIR scheme using finite field isomorphism

In the following we describe the simplest case of the code-based framework, i.e., by considering linear codes over an arbitrary finite field and a field homomorphism for the retrieval function.

4.1.1 Scheme

Setup: Since the identity map is the only non-zero field endomorphism, the retrieval function $f : \mathbb{F}_q \to \mathbb{F}_q$ has to be the identity map. We consider the sets $X = \mathbb{F}_q$, $Y = \ker(f) = \{0\}$ and $Z = f^{-1}(\mathbb{F}_q^\times) = \mathbb{F}_q^\times$. It is easy to see that f satisfies all the conditions of a retrieval function.

Let $\mathbf{M} = (m_i) \in \mathbb{F}_q^N$ represent the database, i.e., there are N files in the database, each file is of size q. Let $\mathcal{C}$ be a random linear $[n, k]$ code over $\mathbb{F}_q$. The code $\mathcal{C}$ is kept secret by the user.

Query generation: Let $\mathbf{G}$ be a generator matrix of $\mathcal{C}$, and let $I \subseteq \{1, \dots, n\}$ be an information set. We use $\mathbf{G}$ to perform the encoding, i.e., the encoding map $\mathrm{Enc} : \mathbb{F}_q^k \to \mathbb{F}_q^n$ is given by $\mathbf{a} \mapsto \mathbf{aG}$.

Let $\mathbf{a}_1, \dots, \mathbf{a}_N$ be randomly chosen vectors in $\mathbb{F}_q^k$, and define the corresponding codewords $\mathbf{c}_i := \mathrm{Enc}(\mathbf{a}_i) = \mathbf{a}_i\mathbf{G}$ for all $i \in \{1, \dots, N\}$.

Table 1 Comparison of different PIR schemes with respect to the code-based framework

PIR scheme	Base ring $\mathcal{R}$	Retrieval function	Set X	Set Y	Set Z
Basic PIR using field homomorphisms	A finite field $\mathbb{F}_q$	$f : \mathbb{F}_q \to \mathbb{F}_q$ $x \mapsto x$	$\mathbb{F}_q$	$\{0\}$	$\mathbb{F}_q^\times$
HHWZ PIR	A finite field extension $\mathbb{F}_{q^m}$	$f : \mathbb{F}_{q^m} \to \mathbb{F}_{q^m}$ $x \mapsto \mathrm{Proj}_{\mathcal{V}}(x)$, where $\mathcal{V}$ is a non-trivial $\mathbb{F}_q$-subspace of $\mathbb{F}_{q^m}$	$\mathbb{F}_q$	$\mathcal{W}$ such that $\mathcal{V} \oplus \mathcal{W} = \mathbb{F}_{q^m}$	$\mathcal{V} \setminus \{0\}$
AMG PIR	A prime field $\mathbb{F}_p$ where p is prime greater than $2^{3\ell}$ for some positive integer ℓ	$f : \mathbb{F}_p \to \mathbb{F}_p$, $x \mapsto x - \mathrm{wt}_{L_t}(x \mod t)$, where $t = 2^{2\ell}$	$\{0, 1, \ldots, 2^\ell - 1\}$	$\{-1, 1\}$	$\{t\}$
LWE-based PIR	$\mathbb{Z}/q\mathbb{Z}$ for some positive integer q	$f : \mathbb{Z}/q\mathbb{Z} \to \mathbb{Z}/q\mathbb{Z}$ $x \mapsto x \pmod t$, for some positive integer $t < q$ with $\gcd(t, q) = 1$	$\{0, \ldots, t-1\}$	$\{ty \mid y \leftarrow \chi\}$	$\{ty + 1 \mid y \leftarrow \chi\}$

Note that since I is an information set, we have $(\mathbf{c}_i)_I = \mathbf{a}_i\mathbf{G}_I$ for all $i \in \{1, \dots, N\}$. Recall that in the code-based framework we send $\mathbf{a}_i$'s in the query to facilitate the decoding in the reply extraction process. However, in this case, this can equivalently be achieved by adding no errors at the coordinates that are indexed by I. In particular, let v be a random element in I^C, and we randomly choose error vectors $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_N$ in $\mathbb{F}_q^n$ such that

1. $\mathrm{Supp}(\mathbf{e}_i) \subseteq I^C$ for all $i \in \{1, \dots, N\}$,
2. $\mathbf{e}_i[v] = 0$ for all $i \neq b$, and $\mathbf{e}_b[v] \neq 0$.

Let $\mathbf{q}_i := \mathbf{c}_i + \mathbf{e}_i$ for all $i \in \{1, \dots, N\}$. The query is then given by

$$Q := \{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_N\}.$$

Reply generation:
The database computes

$$\mathbf{r} = \sum_{i=1}^{N} m_i \mathbf{q}_i \in \mathbb{F}_q^n.$$

Reply extraction: Write $\mathbf{r} = \mathbf{c} + \mathbf{e}$, where $\mathbf{c} := \sum_{i=1}^{N} m_i \mathbf{c}_i$ and $\mathbf{e} := \sum_{i=1}^{N} m_i \mathbf{e}_i$.
Since I is an information set and $\mathrm{Supp}(\mathbf{e}) \subseteq I^C$, we can perform decoding on $\mathbf{r}$ by computing

$$\mathbf{r} - \mathbf{r}_I \mathbf{G}_I^{-1} \mathbf{G} = \mathbf{e} = \sum_{i=1}^{N} m_i \mathbf{e}_i.$$

We now only consider the v-th coordinate of $\mathbf{e}$ and apply the identity retrieval function, which gives $m_b \mathbf{e}_b[v]$, as for all $i \neq b$ we have that $\mathbf{e}_i[v] = 0$. Since $\mathbf{e}_b[v] \neq 0$, we can retrieve m_b.

4.1.2 Security

As we discussed in Sect. 3.3, the security of the presented PIR scheme relies on the hardness of solving the distinguishability problem (see Problem 1). In this case, the distinguishability problem can be solved in polynomial time using the first strategy mentioned in Sect. 3.3.

Let $\mathbf{A}$ be the matrix containing all the query vectors as rows, i.e.,

$$\mathbf{A} = \begin{pmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \vdots \\ \mathbf{q}_N \end{pmatrix} = \begin{pmatrix} \mathbf{c}_1 + \mathbf{e}_1 \\ \mathbf{c}_2 + \mathbf{e}_2 \\ \vdots \\ \mathbf{c}_N + \mathbf{e}_N \end{pmatrix} = \mathbf{C} + \mathbf{E},$$

with $\mathbf{C} = \begin{pmatrix} \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_N \end{pmatrix}$ and $\mathbf{E} = \begin{pmatrix} \mathbf{e}_1 \\ \vdots \\ \mathbf{e}_N \end{pmatrix}$. Since I is an information set, we have

$$\begin{aligned}\mathbf{A}_I &= \mathbf{C}_I + \mathbf{E}_I = \mathbf{C}_I,\\ \mathbf{A}_{I^C} &= \mathbf{C}_{I^C} + \mathbf{E}_{I^C}\\ &= \mathbf{C}_I\mathbf{G}_I^{-1}\mathbf{G}_{I^C} + \mathbf{E}_{I^C}\\ &= \mathbf{A}_I\mathbf{G}_I^{-1}\mathbf{G}_{I^C} + \mathbf{E}_{I^C}.\end{aligned}$$

This implies that

$$\mathbf{E}_{I^C} = \mathbf{A}_{I^C} - \mathbf{A}_I\mathbf{G}_I^{-1}\mathbf{G}_{I^C},$$

and hence the vector $\big(\mathbf{e}_1[v], \mathbf{e}_2[v], \ldots, \mathbf{e}_N[v]\big)$ belongs to the column span of $\mathbf{A}$. We recall that $\mathbf{e}_b[v] \neq 0$ and $\mathbf{e}_i[v] = 0$ for all $i \neq b$. This means that the b-th unitary vector, i.e., the all zero vector having the entry 1 at the b-th position, is in the column span of $\mathbf{A}$.

An attacker can easily find such a vector by simply going through all N unitary vectors and checking their existence in the column span of $\mathbf{A}$. Moreover, existence of another vector of Hamming weight one in the column span of $\mathbf{A}$ is very unlikely. More precisely, given an $N \times (n-1)$ random matrix $\mathbf{A}$ where $N > n$, the probability of having a weight one vector in the column span of $\mathbf{A}$ is $(n-1)q^{(n-N)}$, which is negligible. Despite having a small probability, there exist at most n unit vectors in the column span of $\mathbf{A}$, which leaks information about the index b, since $n < N$.

4.2 HHWZ PIR scheme

Recently, Holzbaur, Hollanti and Wachter-Zeh have proposed the first single server PIR scheme based on coding theory in [19]. In this PIR scheme the authors consider the field extension $\mathbb{F}_{q^m}$ and secretly choose a partition of the basis over $\mathbb{F}_q$. Shortly after, this proposal has been attacked in [20], using that the removal of one row within the query matrix and checking for the dimension of the rest reveals the position of the desired file.

In the following, we describe this PIR scheme presented in [19] with respect to our code-based framework. Later, we also present the attack [20] in terms of solving the distinguishability problem.

Note that the original PIR scheme differs from our description in the database and query setup. In [19], the authors consider the database elements to be $L \times \delta$ matrices over the base field $\mathbb{F}_q$, and the query elements are also $\delta \times n$ matrices over the base field $\mathbb{F}_{q^m}$. Note that the authors have used the technique of iterative reply generation, i.e., by using the same query to retrieve each of the L rows of the database file. In the following description, we consider $L = 1$ and use an equivalent setup where the database files are single elements in $\mathbb{F}_q$ and the query elements are vectors over $\mathbb{F}_{q^m}$.

4.2.1 Scheme

In this case, we work over an extension of the finite field $\mathbb{F}_q$ and the retrieval function is an $\mathbb{F}_q$-linear map.

Setup: Let $\{\beta_1, \dots, \beta_m\}$ be a basis of $\mathbb{F}_{q^m}$ as an $\mathbb{F}_q$-vector space. Further, let $\mathcal{V}$ be the subspace $\mathrm{Span}_{\mathbb{F}_q}(\beta_1, \dots, \beta_s)$ and $\mathcal{W}$ be $\mathrm{Span}_{\mathbb{F}_q}(\beta_{s+1}, \dots, \beta_m)$, where s is some integer in $\{1, \dots, m\}$. The retrieval function is given as

$$\begin{aligned} \mathrm{Proj}_{\mathcal{V}} : \mathbb{F}_{q^m} &\to \mathbb{F}_{q^m}, \\ \sum_{i=1}^{m} \lambda_i \beta_i &\mapsto \sum_{i=1}^{s} \lambda_i \beta_i. \end{aligned}$$

Let X be the set $\mathbb{F}_q$, $Y = \ker(f) = \mathcal{W}$ and $Z = f^{-1}(\mathbb{F}_{q^m}^{\times}) = \mathcal{V} \setminus \{0\}$. It is easy to check that $\mathrm{Proj}_{\mathcal{V}}$ satisfies all the conditions of the retrieval function.

Let $\mathbf{M} = (m_i) \in \mathbb{F}_q^N$ be the database, i.e., there are N files in the database, each file is of size q. Suppose the user wants to retrieve the b-th file from the database. Let $\mathcal{C}$ be a random $[n, k]$ linear code over $\mathbb{F}_{q^m}$.

Query generation: For the encoding and decoding, we follow the same procedure as in Sect. 4.1.

Let $\mathbf{G}$ be a generator matrix of $\mathcal{C}$, and let $I \subseteq \{1, \dots, n\}$ be an information set. We use $\mathbf{G}$ to perform the encoding, i.e., the encoding map is $\mathrm{Enc} : \mathbb{F}_q^k \to \mathbb{F}_q^n$ given by $\mathbf{a} \mapsto \mathbf{aG}$.

Let $\mathbf{a}_1, \dots, \mathbf{a}_N$ be randomly chosen vectors in $\mathbb{F}_{q^m}^k$, and define the corresponding codewords $\mathbf{c}_i := \mathrm{Enc}(\mathbf{a}_i) = \mathbf{a}_i\mathbf{G}$ for all $i \in \{1, \dots, N\}$.

As in Sect. 4.1, we perform the decoding by adding no errors at the coordinates that are indexed by I.

Let v be a fixed element in I^C. Now, we choose error vectors $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_N$ randomly in $\mathbb{F}_{q^m}^n$ such that

1. $\mathrm{Supp}(\mathbf{e}_i) \subseteq I^C$ for all $i \in \{1, \dots, N\}$,
2. $\mathbf{e}_i[v] \in \mathcal{W}$ for all $i \neq b$, and $\mathbf{e}_b[v] \in \mathcal{V} \setminus \{0\}$.

Let $\mathbf{q}_i := \mathbf{c}_i + \mathbf{e}_i$ for $i \in \{1, \dots, N\}$. The query is then given by

$$Q := \{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_N\}.$$

Reply generation: The response is generated by computing

$$\mathbf{r} = \sum_{i=1}^{N} m_i \mathbf{q}_i.$$

Reply extraction:

Write $\mathbf{r} = \mathbf{c} + \mathbf{e}$, where $\mathbf{c} = \sum_{i=1}^{N} m_i \mathbf{c}_i$ and $\mathbf{e} = \sum_{i=1}^{N} m_i \mathbf{e}_i$.

Since I is an information set and $\mathrm{Supp}(\mathbf{e}) \subseteq I^C$, we can perform the decoding on $\mathbf{r}$ by computing

$$\mathbf{r} - \mathbf{r}_I \mathbf{G}_I^{-1} \mathbf{G} = \mathbf{e} = \sum_{i=1}^{N} m_i \mathbf{e}_i.$$

Now we consider the v-th coordinate of $\mathbf{e}$ and apply the retrieval function, which gives

$$\mathrm{Proj}_{\mathcal{V}}\left(\sum_{i=1}^{N} m_i \mathbf{e}_i[v]\right) = m_b \mathbf{e}_b[v].$$

This works because $\mathbf{e}_i[v] \in \mathcal{W}$ for all $i \neq b$, and $\mathbf{e}_b[v] \in \mathcal{V} \setminus \{0\}$. Moreover, since we know $\mathbf{e}_b[v]$, we can retrieve m_b.

4.2.2 Security

The original PIR scheme [19] has been attacked in [20], by solving the distinguishability problem. The attack follows the second strategy mentioned in Sect. 3.3.

Let $\mathbf{A}$ be the matrix containing all the query vectors as rows, i.e.,

$$\mathbf{A} = \begin{pmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \vdots \\ \mathbf{q}_N \end{pmatrix} = \begin{pmatrix} \mathbf{c}_1 + \mathbf{e}_1 \\ \mathbf{c}_2 + \mathbf{e}_2 \\ \vdots \\ \mathbf{c}_N + \mathbf{e}_N \end{pmatrix} = \mathbf{C} + \mathbf{E},$$

with $\mathbf{C} = \begin{pmatrix} \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_N \end{pmatrix}$ and $\mathbf{E} = \begin{pmatrix} \mathbf{e}_1 \\ \vdots \\ \mathbf{e}_N \end{pmatrix}$.

For each $i \in \{1, \ldots, N\}$, let $\mathbf{A}_i$ be the submatrix of $\mathbf{A}$ obtained by deleting the i-th row. Then the $\mathbb{F}_q$-rank of these matrices satisfy the following proposition.

Proposition 1 [20, *Proposition* 3.1] *Let* $\mathbf{A}$ *be given as above. Then*

$$\mathrm{rank}_{\mathbb{F}_q}(\mathbf{A}) = \mathrm{rank}_{\mathbb{F}_q}(\mathbf{C}) + \mathrm{rank}_{\mathbb{F}_q}(\mathbf{E}).$$

Moreover, for all $i \in \{1, \ldots, N\}$

$$\mathrm{rank}_{\mathbb{F}_q}(\mathbf{A}_i) = \mathrm{rank}_{\mathbb{F}_q}(\mathbf{C}_i) + \mathrm{rank}_{\mathbb{F}_q}(\mathbf{E}_i).$$

In the case when $N < mn$, the query size becomes bigger than the size of the database, i.e., the scheme is no better than the trivial PIR protocol of downloading entire database. Hence, we assume $N \geq mn$ and we use the following corollary to distinguish the index b in polynomial time.

Corollary 1 [20, *Corollary* 3.2, *Proposition* 3.3] *Let* $\mathbf{A}_i$ *be given as above. Then, with high probability,*

1. $\mathrm{rank}_{\mathbb{F}_q}(\mathbf{A}_b) = mn - s$,
2. *for* $i \neq b$, *we have that* $\mathrm{rank}_{\mathbb{F}_q}(\mathbf{A}_i) = mn$.

Proof From Proposition 1, we have that $\text{rank}_{\mathbb{F}_q}(\mathbf{A}_i) = \text{rank}_{\mathbb{F}_q}(\mathbf{C}_i) + \text{rank}_{\mathbb{F}_q}(\mathbf{E}_i)$ for all $1 \leq i \leq N$.

In the first case, we have that $\text{rank}_{\mathbb{F}_q}(\mathbf{C}_b) = mk$ and $\text{rank}_{\mathbb{F}_q}(\mathbf{E}_b) = (n-k-1)m + (m-s)$ (with high probability), where the first part comes from the columns not indexed by v, which live in the full space $\mathbb{F}_{q^m} = \mathcal{W} + \mathcal{V}$ and the second part comes from the column indexed by v, which lives in the subspace $\mathcal{W}$. Note that the equation $\text{rank}_{\mathbb{F}_q}(\mathbf{E}_b) = m(n-k) - s$ holds true with high probability due to the randomness of the matrix entries.

In the case of $i \neq b$, we still have that $\text{rank}_{\mathbb{F}_q}(\mathbf{C}_i) = mk$, but now $\text{rank}_{\mathbb{F}_q}(\mathbf{E}_i) = (n-k-1)m + m$ (with high probability), where the first part comes from the columns not indexed by v and the second part comes from the column v (observe that in this case all columns are in the full space $\mathbb{F}_{q^m} = \mathcal{W} + \mathcal{V}$). Note that the equation $\text{rank}_{\mathbb{F}_q}(\mathbf{E}_i) = m(n-k)$ holds true with high probability due to the randomness of the matrix entries. □

4.3 AMG PIR scheme

In the following, we describe the PIR scheme presented in [21] with respect to our code-based framework. Later, we also present the lattice-based attack [22] in terms of solving the distinguishability problem. Note that the original PIR scheme differs from our description in the following way:

- Database setup: in [21], the authors consider the database elements to be vectors over the base field $\mathbb{F}_p$. Moreover, each query element is a matrix over $\mathbb{F}_p$. In the following description, we use an equivalent setup where the database files are single elements in $\mathbb{F}_p$ and query elements are vectors over $\mathbb{F}_p$.
- Noise-scrambling matrix Δ: the authors introduce an invertible diagonal matrix Δ in order to disguise the soft-noise error vectors from the hard-noise error vectors. In our description, we ignore this scrambling matrix Δ, as we will see in the security discussion that Δ has no effect on the column space of the query matrix.
- In [21], the rate k/n of the underlying linear code is fixed $k/n = 0.5$. In our description we use an arbitrary rate.

4.3.1 Scheme

In this scheme, we work over a finite field $\mathbb{F}_p$, where p is a prime number. We will see $\mathbb{F}_p$ as $\{-\lfloor \frac{p}{2} \rfloor, \ldots, \lfloor \frac{p}{2} \rfloor\}$.

Setup: Assume that the database is of the form $\mathbf{M} = (m_i) \in \{0, 1, \ldots, 2^\ell - 1\}^N$ with $\ell = \lceil \log_2(N) \rceil + 1$, i.e., there are N files in the database each of size ℓ bits. Note that if the file size is bigger than ℓ bits, then we split the files in chunks of ℓ bits. Suppose the user wants to retrieve the b-th file from the database.

Let p be a prime number greater than $2^{3\ell}$ and $t = 2^{2\ell}$. The retrieval function is given by the remainder of the Lee weight corresponding to modulo t, i.e.,

$$f : \mathbb{F}_p \to \mathbb{F}_p,$$
$$x \mapsto x - \mathrm{wt}_{L_t}(x \mod t),$$

where wt_{L_t} denotes the Lee weight on $\mathbb{Z}/t\mathbb{Z} = \{0, 1, \ldots, t-1\}$, which is defined as

$$\mathrm{wt}_{L_t}(z) := \min\{z, t-z\}.$$

The set $X = \{0, 1, \ldots, 2^\ell - 1\}$, $Y = \{-1, 1\} \subseteq \ker(f)$ and $Z = \{t\} \subseteq f^{-1}(\mathbb{F}_p^\times)$.

Now observe that a linear combination of elements in Y with scalars from X having arbitrary number of terms does not necessarily belongs to $\ker(f)$. However, the condition is satisfied when we have at most N number of terms in the linear combination: for $x_1, \ldots, x_N \in X$ and $y_1, \ldots, y_N \in Y$ we have that

$$|x_1 y_1 + \cdots + x_N y_N| \leq N 2^\ell < \frac{t}{2},$$

and hence

$$f\left(\sum_{i=1}^{N} x_i y_i\right) = \sum_{i=1}^{N} x_i y_i - \mathrm{wt}_{L_t}\left(\sum_{i=1}^{N} x_i y_i \mod t\right)$$
$$= \sum_{i=1}^{N} x_i y_i - \sum_{i=1}^{N} x_i y_i = 0.$$

Further we have that for $y \in Y, x \in X$ and $z \in Z$

$$\begin{aligned} f(y + xz) &= f(y + xt) \\ &= y + xt - \mathrm{wt}_{L_t}(y + xt \mod t) \\ &= y + xt - \mathrm{wt}_{L_t}(y \mod t) \\ &= y + xt - y \\ &= xt = xf(t), \end{aligned}$$

since $f(z) = f(t) = t - \mathrm{wt}_{L_t}(t \mod t) = t$.

Let $\mathcal{C}$ be a random linear $[n, k]$ code over $\mathbb{F}_p$, which is kept secret by the user.

Query generation: For the encoding and decoding, we follow the same procedure as in Sects. 4.1 and 4.2.

Let $\mathbf{G}$ be a generator matrix of $\mathcal{C}$, and let $I \subseteq \{1, \ldots, n\}$ be an information set. We use $\mathbf{G}$ to perform the encoding, i.e., the encoding map is $\mathrm{Enc} : \mathbb{F}_q^k \to \mathbb{F}_q^n$ given by $\mathbf{a} \mapsto \mathbf{aG}$.

Let $\mathbf{a}_1, \ldots, \mathbf{a}_N$ be randomly chosen vectors in $\mathbb{F}_q^k$, and define the corresponding codewords $\mathbf{c}_i := \mathrm{Enc}(\mathbf{a}_i) = \mathbf{a}_i\mathbf{G}$ for all $i \in \{1, \ldots, N\}$.

As in Sects 4.1 and 4.2, we perform the decoding by adding no errors at the coordinates that are indexed by I.

Let v be a fixed element in I^C. Now, we choose error vectors $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_N$ randomly in $\mathbb{F}_{q^m}^n$ such that

1. $\mathrm{Supp}(\mathbf{e}_i) \subseteq I^C$ for all $i \in \{1, \dots, N\}$,
2. $\mathbf{e}_i[v] \in \{\pm 1\}$ for all $i \neq b$, and $\mathbf{e}_b[v] = t$.

Let $\mathbf{q}_i := \mathbf{c}_i + \mathbf{e}_i$ for all $i \in \{1, \dots, N\}$. The query is then given by

$$Q := \{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_N\}.$$

Reply generation: The response is generated by computing

$$\mathbf{r} = \sum_{i=1}^{N} m_i \mathbf{q}_i.$$

Reply extraction:

Write $\mathbf{r} = \mathbf{c} + \mathbf{e}$, where $\mathbf{c} = \sum_{i=1}^{N} m_i \mathbf{c}_i$ and $\mathbf{e} = \sum_{i=1}^{N} m_i \mathbf{e}_i$.

Since I is an information set and $\mathrm{Supp}(\mathbf{e}) \subseteq I^C$, we can perform the decoding on $\mathbf{r}$ by computing

$$\mathbf{r} - \mathbf{r}_I \mathbf{G}_I^{-1} \mathbf{G} = \mathbf{e} = \sum_{i=1}^{N} m_i \mathbf{e}_i.$$

We will only focus on the v-th coordinate of $\mathbf{e}$ and apply the retrieval function to obtain

$$\begin{aligned} f\left(\sum_{i=1}^{N} m_i \mathbf{e}_i[v]\right) &= \sum_{i=1}^{N} m_i \mathbf{e}_i[v] - \mathrm{wt}_{L_t}\left(\sum_{i=1}^{N} m_i \mathbf{e}_i[v] \mod t\right) \\ &= \left(\sum_{\substack{i=1 \\ i \neq b}}^{N} m_i \mathbf{e}_i[v] - \mathrm{wt}_{L_t}\left(\sum_{\substack{i=1 \\ i \neq b}}^{N} m_i \mathbf{e}_i[v] \mod t\right)\right) + m_b \mathbf{e}_b[v] \\ &= m_b \mathbf{e}_b[v] = m_b t. \end{aligned}$$

This works since

$$\left|\sum_{\substack{i=1 \\ i \neq b}}^{N} m_i \mathbf{e}_i[v]\right| < t/2 \quad \text{and} \quad m_b \mathbf{e}_b[v] \text{is a multiple of} t,$$

and hence

$$\mathrm{wt}_{L_t}\left(\sum_{i=1}^{N} m_i \mathbf{e}_i[v] \mod t\right) = \mathrm{wt}_{L_t}\left(\sum_{\substack{i=1 \\ i\neq b}}^{N} m_i \mathbf{e}_i[v] \mod t\right) = \sum_{\substack{i=1 \\ i\neq b}}^{N} m_i \mathbf{e}_i[v].$$

Now since $\gcd(t, p) = 1$, we can retrieve m_b.

4.3.2 Security

In [22], Liu et al. presented a lattice-based attack on the AMG PIR scheme. The method used in the attack can be described as per the first strategy, mentioned in Sect. 3.3, to solve the distinguishability problem.

Let $\mathbf{A}$ be the matrix containing all the query vectors as rows, i.e.,

$$\mathbf{A} = \begin{pmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \vdots \\ \mathbf{q}_N \end{pmatrix} = \begin{pmatrix} \mathbf{c}_1 + \mathbf{e}_1 \\ \mathbf{c}_2 + \mathbf{e}_2 \\ \vdots \\ \mathbf{c}_N + \mathbf{e}_N \end{pmatrix}.$$

As discussed in Sect. 4.1, the vector $\big(\mathbf{e}_1[v], \mathbf{e}_2[v], \ldots, \mathbf{e}_N[v]\big)$ belongs to the column span of $\mathbf{A}$.

Recall that by construction, the vector $\big(\mathbf{e}_1[v], \mathbf{e}_2[v], \ldots, \mathbf{e}_N[v]\big)$ has $N-1$ entries from $\{-1, +1\}$ and one entry with value equal to t. If we delete the b-th row of $\mathbf{A}$, call it the matrix $\mathbf{A}_b$, then the vector

$$\big(\mathbf{e}_1[v], \ldots, \mathbf{e}_{b-1}[v], \mathbf{e}_{b+1}[v], \ldots, \mathbf{e}_N[v]\big)$$

will be, with a very high probability, the shortest vector in the p-ary lattice generated by the columns of $\mathbf{A}_b$. More precisely, the lattice is generated by the n columns of $[\mathbf{A}_b | p\mathbf{Id}_{N-1}]$. However, it is still infeasible to find this vector due to the large dimension of the lattice.

In [22], the authors construct multiple small dimensional lattices. Let $k \leq s \leq N$, and let $\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(\lceil N/s \rceil)}$ be a row-wise partitioning of the matrix $\mathbf{A}$, i.e., $\mathbf{A}^{(i)}$ is the $s \times n$ matrix given by s rows of $\mathbf{A}$ indexed by $\{(i-1)s+1, \ldots, is\}$. Now, let $\mathcal{L}_i$ be the p-ary lattice generated by the columns of $\mathbf{A}^{(i)}$. Note that the dimension of the lattices $\mathcal{L}_i$ is s, hence the attacker chooses s such that implementing basis reduction algorithms for $\mathcal{L}_i$ is feasible. In order to find the index b, the attacker goes through each of these lattices.

Note that the index b of the desired file corresponds to the lattice $\mathcal{L}_{\lfloor b/s \rfloor}$, which the attacker is able to find, and then the attacker finds the index b by solving the closest vector problem for $\mathcal{L}_{\lfloor b/s \rfloor}$.

More in detail, in the case of $i \neq \lfloor b/s \rfloor$, we observe that the shortest vector in $\mathcal{L}_i$ corresponds to the vector $(\mathbf{e}_{(i-1)s+1}[v], \ldots, \mathbf{e}_{is}[v])$ having entries in $\{-1, +1\}$. This observation does not hold in the case of $i = \lfloor b/s \rfloor$ due to the existence of large t. The

attacker uses the lattice reduction algorithms to find the shortest vector in each $\mathcal{L}_i$, and consequently finds the corresponding lattice $\mathcal{L}_{\lfloor b/s \rfloor}$.

Now, the index b can be located using solving the closest vector problem. Let $j = \lfloor b/s \rfloor$. Then observe that $(\mathbf{e}_{(j-1)s+1}[v], \dots, \mathbf{e}_{js}[v]) \in \mathcal{L}_j$ is the closest lattice vector to $(0, \dots, 0, t, 0, \dots, 0)$ (with t at the b-th position). To find the index b, we can use Kannan's embedding technique [23] to solve (at most) s instances of the closest vector problem with inputs vector of the form $(0, \dots, 0, t, 0, \dots, 0)$.

4.4 Ring-LWE based PIR schemes

In the section, we describe the PIR schemes constructed using the Ring-LWE (RLWE) based homomorphic encryption schemes. In particular, we consider the construction of XPIR scheme [16] that uses the Ring-LWE based homomorphic encryption scheme presented in [24].

The original PIR scheme differs from our description in the error distribution as follows. In [16], the authors use two different distributions χ and χ' to sample errors. The distribution χ is used to generate the public key and the distribution χ', having larger variance, is used for encryption. In the following description, we consider only one distribution, mimicking χ', to sample error vectors in the query generation process.

We would like to remark that in the following description, the database elements and the query elements are polynomials of degree smaller than n with coefficients in $\mathcal{R}$, which can also be represented by vectors in $\mathcal{R}^n$.

4.4.1 Scheme

In this scheme, we work over a finite ring $\mathbb{Z}/q\mathbb{Z}$, where q is a positive integer. Instead of a random linear code over $\mathbb{Z}/q\mathbb{Z}$, we consider a random negacyclic code over $\mathbb{Z}/q\mathbb{Z}$.

Setup: Let q, t be positive integers with $t < q$ and $\gcd(t, q) = 1$. The retrieval function is given by

$$\begin{aligned} f : \mathbb{Z}/q\mathbb{Z} &\to \mathbb{Z}/q\mathbb{Z}, \\ x &\mapsto x \pmod{t}. \end{aligned}$$

Let χ be a discrete Gaussian distribution with standard deviation σ. The parameters q, n, t, σ are chosen such that they satisfy $Nt^2\sigma\sqrt{n} < q/2$, where n is the length of the linear code that will be used in query generation.

Now, we define the subsets

$$\begin{aligned} X &= \{0, \dots, t-1\} \subseteq \mathbb{Z}/q\mathbb{Z}, \\ Y &= \{ty \mid y \text{ is sampled from the distribution } \chi\}, \\ Z &= \{ty + 1 \mid y \text{ is sampled from the distribution } \chi\}. \end{aligned}$$

Observe that for $x_1, x_2, \dots, x_N \in X$ and $ty_1, ty_2, \dots, ty_N \in Y$ we have that

$$f\left(\sum_{i=1}^{N} x_i t y_i\right) = \sum_{i=1}^{N} x_i t y_i \mod t$$
$$= 0.$$

This works since the choice of parameters q, n, t, σ implies that $|\sum_{i=1}^{N} x_i t y_i| < q/2$ with very high probability. And for $x \in X, ty \in Y$ and $tz + 1 \in Z$ we have that

$$f(y + xz) = ty + x(tz + 1) \mod t$$
$$= x \mod t = x = xf(z),$$

since $|ty + x(tz + 1)| < q/2$.

Let n be a power of 2, and let $R_q := (\mathbb{Z}/q\mathbb{Z})[x]/(x^n + 1)$. Let $\mathbf{M} = (m_i) \in (X[x]/(x^n + 1))^N$, i.e., there are N files in the database and each file is an element in R_q with coefficients in X. In particular, each file is of size $\log_2(tn)$ bits. Suppose the user wants to retrieve the b-th file from the database.

Let $\mathcal{C}$ be a negacyclic code of length n over $\mathbb{Z}/q\mathbb{Z}$ generated by some randomly chosen $s \in R_q$, i.e., $\mathcal{C}$ is a ideal in R_q generated by s. The code is kept secret by the user.

Query generation: We use the generating polynomial s to define the encoding map, i.e., $Enc : R_q \to R_q$ is given by $a \mapsto as$.

Let $a_1, a_2, \ldots, a_N$ be randomly chosen elements in R_q, and define N codewords $c_i := a_i s$ for all $i \in \{1, \ldots, N\}$.

Now, we choose the errors $e_1, e_2, \ldots, e_N$ in R_q such that they satisfy the following two conditions that allow the reply extraction:

1. $e_i = ty_i$, with y_i sampled from the distribution χ, for all $i \neq b$,
2. $e_b = ty_b + 1$ with y_b sampled from χ.

Let $\mathbf{q}_i := (a_i, c_i + e_i)$ for all $i \in \{1, \ldots, N\}$. The query is then given by

$$Q := \{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_N\}.$$

Reply generation: The response is generated by computing

$$\mathbf{r} = \sum_{i=1}^{N} m_i \mathbf{q}_i = \sum_{i=1}^{N} (m_i a_i, m_i c_i + m_i e_i) =: (r_1, r_2).$$

Reply extraction:

By applying the encoding map Enc on r_1, we first decode r_2 to obtain the error part, i.e.,

$$r_2 - Enc(r_1) = r_2 - s r_1 = \sum_{i=1}^{N} m_i e_i.$$

After that we can use the retrieval function f,

$$f\left(\sum_{i}^{N} m_i e_i\right) = \sum_{i=1}^{N} m_i t y_i + m_b \pmod{t}$$
$$= m_b.$$

Note that here we apply f on an element of R_q, which is done by applying f on each coefficient.

The last equality follows from the conditions on the parameters n, q, t, σ, since the maximal coefficient of $\sum_{i=1}^{N} m_i e_i$ is, with high probability, upper bounded by $Nt^2\sigma\sqrt{n}$ (see [24, Lemma 1]), which is less than $q/2$.

4.4.2 Security

As mentioned above, the XPIR scheme [16] uses the fully homomorphic encryption scheme presented in [24], whose security is based on the hardness of solving the polynomial learning with error (PLWE) problem, which is a simplified version of the ring LWE problem.

Let $R_q = \mathbb{Z}/q\mathbb{Z}[x]/(x^n + 1)$, and let χ be a narrow discrete Gaussian distribution on R_q. Then the PLWE assumption states that it is computationally hard to distinguish a polynomial number of samples of the form $(a_i, a_i s + e_i)$ and the same number of samples of the form (a_i, u_i), where s, a_i's and u_i's are sampled uniformly from R_q and the e_i's are sampled from χ.

Moreover, [24, Proposition 1] states that if the samples are of the form $(a_i, a_i s + te_i)$, where a_i, s, e_i are as above and $t \in (\mathbb{Z}/q\mathbb{Z})^\times$, then distinguishing such samples from the uniform samples is equivalent to the PLWE assumption.

Let $\mathbf{A}$ be the query matrix and $\mathbf{A}_i$ be the submatrix of $\mathbf{A}$ obtained by deleting the i-th row. Translating the above mentioned approach [24, Proposition 1] to our generic framework means that distinguishing $\mathbf{A}_b$ from an uniformly sampled matrix is equivalent to the PLWE problem.

However, the second strategy mentioned in Sect. 3.3 aims in a different direction: that is to distinguish between A_i for $i \neq b$ and A_b. Thus, this might lead to new security analyses of such PIR schemes.

5 Theoretical remarks

5.1 Generic PIR scheme vs code-based framework

A natural question would be to ask whether any single server PIR scheme can be described in terms of the code-based framework. The answer is no, as the number theoretic PIR scheme by Kushilevitz and Ostrovsky [25] does not fit the framework. However, if we restrict to the class of PIR schemes that generates replies by contracting the database elements and the query elements using linear combinations (which will be denoted from now on as additive PIR schemes), then the answer is yes. In the following, we discuss the requirements of an arbitrary additive PIR

scheme and argue the necessity of the elements in the code-based framework to fulfil those requirements:

1. *Ambient space:* An additive PIR scheme needs two operations: multiplication ($*$) between database and query elements, and addition ($+$) of those products. Hence, the canonical choice of the ambient space is rings. For practical reasons, the rings should be finite.
2. *Retrieval:* Let the database be denoted by $\mathcal{DB} = \{db_1, \ldots, db_N\}$, and the corresponding query be given by $Q = \{q_1, \ldots, q_N\}$. Suppose that the user wants to retrieve the b-th file. In an additive PIR scheme, the reply is $\sum_{i=1}^{N} db_i * q_i$ and user wants to retrieve db_b from the reply. The operation $\sum_{i=1}^{N} db_i * q_i \mapsto db_b$, denoted by g, is an analogue to the retrieval function used in the code-based framework. First we note that g annihilates $\sum_{i \neq b} db_i * q_i$ in such a way that we are only left with $g(db_b * q_b)$. And then db_b is recovered from $g(db_b * q_b)$. These two properties imply that db_i's and q_i's live in special subsets of the ambient space R. Let X denote the space of database elements, Y denote the space of query elements that are not associated with the desired file and Z denote the space of query element associated with the desired file. The requirements on g imply that: (1) a linear combination of elements in Y with scalars in X belongs to the kernel of g, and (2) $g(x * z) = x * g(z)$ and $g(z)$ is an invertible element, for any $x \in X$ and $z \in Z$. These two conditions are the basis of the conditions of the retrieval function used in the code-based framework.
3. *Privacy:* Another important aspect of a PIR scheme is privacy, i.e., given a query Q, it should be computationally infeasible to determine the index b of the desired file. Let us look at the scenario where we directly use elements in Y and Z to generate query elements. Then the privacy relies on the hardness of the following decisional problem: given $q \in Y \cup Z$, decide whether $q \in Y$ or $q \in Z$. In general this may not be a hard problem, as one can apply the retrieval function to distinguish the elements between Y and Z. Therefore, to ensure privacy we must add some randomness to the query elements. Moreover, the user should be able to remove this randomness even after receiving the reply that contains their linear combinations. This is exactly the rationale of linear error-correcting codes. We treat the elements of Y and Z as errors, and the added randomness belongs to a random linear code.

5.2 On security of PIR schemes

In terms of the code-based framework, the security of a PIR scheme relies on the type of the underlying retrieval function. As we have noticed from the examples in Sect. 4, the following type of retrieval functions are not safe to use.

1. *Field homomorphisms:* In the case where the retrieval function is a non-trivial field homomorphism, the PIR scheme is then equivalent to the one described in Sect. 4.1. The kernel of the retrieval function must be $\{0\}$, as $\{0\}$ is the only proper ideal in any field. As a consequence, determining the index of the desired

file becomes an easy task of finding a unitary vector in the column space of the query matrix, thus it suffers from the first attack strategy discussed in Sect. 3.3.

2. *Vector space homomorphisms:* In this case, the resulting PIR scheme is equivalent to HHWZ PIR scheme [19], described in Sect. 4.2. The kernel of a non-trivial linear map is a proper subspace of the parent vector space. This results in an exceptionally low rank of the matrix that is obtained from the query matrix by deleting the row that corresponds to the desired file, thus it suffers from the second attack strategy discussed in Sect. 3.3.

We can generalize these two cases to more types of retrieval functions. Clearly, the weakness of vector space homomorphisms can also be observed in the case of free module homomorphims, because of the existence of the notion of rank and dimension for free modules. On the other hand, the weakness of field homomorphisms can be seen in the case of local ring homomorphims. Let R be a finite local ring with maximal ideal M, then the kernel of the retrieval function is a subideal of M. Note that there exists an integer ℓ such that $M^\ell = \{0\}$ and $M^{\ell-1} \neq \{0\}$. Let $a \in M^{\ell-1} \setminus \{0\}$, then note that $ar = 0$ for all $r \in M$. This implies that the special column vector $(\mathbf{e}_1[v], \dots, \mathbf{e}_N[v])$, when multiplied by a, results in a unit vector. Hence, similar to the field homomorphism case, we observe the existence of a unit vector in the column space of the query matrix.

The other two schemes, presented in Sects. 4.3 and 4.4 respectively, do not use additive retrieval functions. Both the schemes work on the idea of using small modulus errors in a large modulus ambient space. Due to which the security eventually relies on finding short vectors in a high dimensional lattice, which is a computationally hard problem. However, in the case of AMG PIR scheme, the problem breaks down over multiple small dimensional lattices and hence the attack becomes feasible. Whereas in the case of LWE-based PIR schemes, this new perspective may have a potential in introducing new approaches for their security analysis.

In order to construct an additive PIR scheme, one may investigate the cases of structured morphisms like ring homomorphisms and module homomorphisms, or the cases of unstructured morphisms like the functions used in AMG scheme and LWE-based schemes.

Furthermore, if one constructs an additive PIR scheme independently, then it would be worth translating the scheme in terms of the code-based framework to check for possible security issues.

Acknowledgements The authors would like to thank Lukas Holzbaur, Antonia Wachter-Zeh and Camilla Hollanti for useful discussions and Razane Tajeddine for bringing this interesting topic to their knowledge. This work was partially supported by Swiss National Science Foundation grant no. 188430, Grant No. 195290 and Forschungskredit of the University of Zurich grant no. FK-19-080.

Funding Open Access funding provided by Universität Zürich.

References

1. Chor, B., Goldreich, O., Kushilevitz, E., Sudan, M.: Private information retrieval. In *Proceedings of IEEE 36th annual foundations of computer science*, pp. 41–50. IEEE (1995)
2. Dvir, Z., Gopi, S.: 2-server PIR with subpolynomial communication. J. ACM (JACM) **63**(4), 1–15 (2016)
3. Beimel, A., Ishai, Y., Kushilevitz, E., Raymond, J-F.: Breaking the $O(n^{1/(2k-1)})$ barrier for information-theoretic private information retrieval. In *Proceedings of the 43rd annual IEEE symposium on foundations of computer science, 2002*, pp. 261–270. IEEE (2002)
4. Sun, H., Jafar, S.A.: The capacity of symmetric private information retrieval. IEEE Trans. Inform. Theory **65**(1), 322–329 (2018)
5. Sun, H., Jafar, S.A.: The capacity of robust private information retrieval with colluding databases. IEEE Trans. Inform. Theory **64**(4), 2361–2370 (2017)
6. Banawan, K., Ulukus, S.: The capacity of private information retrieval from coded databases. IEEE Trans. Inform. Theory **64**(3), 1945–1956 (2018)
7. Freij-Hollanti, R., Gnilke, O.W., Hollanti, C., Karpuk, D.A.: Private information retrieval from coded databases with colluding servers. SIAM J. Appl. Algebra Geom **1**(1), 647–664 (2017)
8. Dong, C., Chen, L.: A fast single server private information retrieval protocol with low communication cost. In *European symposium on research in computer security*, pp. 380–399. Springer (2014)
9. Kushilevitz, E., Ostrovsky, R.: Replication is not needed: single database, computationally—private information retrieval. In *Proceedings 38th annual symposium on foundations of computer science*, pp. 364–373. IEEE (1997)
10. Lipmaa, H., Pavlyk, K.: A simpler rate-optimal CPIR protocol. In *International conference on financial cryptography and data security*, pp. 621–638. Springer (2017)
11. Stern, JP.: A new and efficient all-or-nothing disclosure of secrets protocol. In *International conference on the theory and application of cryptology and information security*, pp. 357–371. Springer (1998)
12. Sion, R., Carbunar, B.: On the computational practicality of private information retrieval. In *Proceedings of the network and distributed systems security symposium*, pp. 2006–06. Internet Society (2007)
13. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Rev **41**(2), 303–332 (1999)
14. Yi, X., Kaosar, M.G., Paulet, R., Bertino, E.: Single-database private information retrieval from fully homomorphic encryption. IEEE Trans. Knowl. Data Eng. **25**(5), 1125–1134 (2012)
15. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. SIAM J. Comput. **43**(2), 831–871 (2014)
16. Aguilar-Melchor, C., Barrier, J., Fousse, L., Killijian, M.-O.: XPIR: private information retrieval for everyone. Proc. Priv. Enhanc. Technol. **2016**(2), 155–174 (2016)
17. Angel, S., Chen, H., Laine, K., Setty, S.: PIR with compressed queries and amortized query processing. In *2018 IEEE symposium on security and privacy (SP)*. pp. 962–979 (2018)
18. Ali, A., Lepoint, T., Patel, S., Raykova, M., Schoppmann, P., Seth, K., Yeo, K.: Communication–computation trade-offs in PIR. IACR Cryptol. ePrint Arch. (2019)
19. Holzbaur, L., Hollanti, C., Wachter-Zeh, A.: Computational code-based single-server private information retrieval. In *2020 IEEE international symposium on information theory (ISIT)*, pp. 1065–1070. IEEE (2020)
20. Bordage, S., Lavauzelle, J.: On the privacy of a code-based single-server computational PIR scheme. Cryptogr, Commun (2021)
21. Melchor, CA, Gaborit, .: A fast private information retrieval protocol. In *2008 IEEE international symposium on information theory*, pp. 1848–1852 (2008)
22. Liu, J., Bi, J.: Cryptanalysis of a fast private information retrieval protocol. In *Proceedings of the 3rd ACM international workshop on ASIA public-key cryptography*, pp. 56–60 (2016)
23. Kannan, R.: Minkowski's convex body theorem and integer programming. Math. Oper. Res. **12**(3), 415–440 (1987)

24. Brakerski, Z., Vaikuntanathan, V.: Fully homomorphic encryption from ring-LWE and security for key dependent messages. In *Annual cryptology conference*, pp. 505–524. Springer (2011)
25. Kushilevitz, E., Ostrovsky, R.: Replication is not needed: single database, computationally-private information retrieval. In *Proceedings 38th annual symposium on foundations of computer science*, pp. 364–373. IEEE (1997)