Multi-Server PIR with Full Error Detection and Limited Error Correction

Reo Eriguchi

Graduate School of Information Science and Technology, The University of Tokyo, Japan
National Institute of Advanced Industrial Science and Technology, Tokyo, Japan

Kaoru Kurosawa

Research and Development Initiative, Chuo University, Tokyo, Japan
National Institute of Advanced Industrial Science and Technology, Tokyo, Japan

Koji Nuida

Institute of Mathematics for Industry, Kyushu University, Japan
National Institute of Advanced Industrial Science and Technology, Tokyo, Japan

Abstract

An ℓ-server Private Information Retrieval (PIR) scheme allows a client to retrieve the τ-th element a_τ from a database $\boldsymbol{a} = (a_1, \dots, a_n)$ which is replicated among ℓ servers. It is called t-private if any coalition of t servers learns no information on τ, and b-error correcting if a client can correctly compute a_τ from ℓ answers containing b errors. This paper concerns the following problems: Is there a t-private ℓ-server PIR scheme with communication complexity $o(n)$ such that a client can detect errors with probability $1 - \epsilon$ even if $\ell - 1$ servers return false answers? Is it possible to add error correction capability to it? We first formalize a notion of $(1 - \epsilon)$-fully error detecting PIR in such a way that an answer returned by any malicious server depends on at most t queries, which reflects t-privacy. We then prove an impossibility result that there exists no 1-fully error detecting (i.e., $\epsilon = 0$) PIR scheme with $o(n)$ communication. Next, for $\epsilon > 0$, we construct 1-private $(1 - \epsilon)$-fully error detecting and $(\ell/2 - O(1))$-error correcting PIR schemes which have $n^{o(1)}$ communication, and a t-private one which has $O(n^c)$ communication for any $t \geq 2$ and some constant $c < 1$. Technically, we show generic transformation methods to add error correction capability to a basic fully error detecting PIR scheme. We also construct such basic schemes by modifying certain existing PIR schemes which have no error detection capability.

2012 ACM Subject Classification Security and privacy → Information-theoretic techniques

Keywords and phrases Private Information Retrieval, Error Detection, Error Correction

Digital Object Identifier 10.4230/LIPIcs.ITC.2022.1

Related Version *Full Version*: `https://eprint.iacr.org/2022/500`

Funding This research was partially supported by JSPS KAKENHI Grant Numbers JP20J20797 and 19H01109, Japan and JST CREST Grant Numbers JPMJCR19F6 and JPMJCR2113, Japan.

1 Introduction

Private Information Retrieval (PIR) was introduced by Chor, Goldreich, Kushilevitz, and Sudan [7]. In an ℓ-server PIR scheme, a client can retrieve the τ-th element a_τ of a database $\boldsymbol{a} = (a_1, \dots, a_n)$ replicated among ℓ servers without revealing any information on the index τ to the servers. A trivial solution is that servers send the entire database to the client. However, it results in communication complexity $O(n)$, which is shown to be optimal in the information-theoretic setting when $\ell = 1$ [7]. To get around this, Chor et al. [7] considered ℓ-server PIR schemes for $\ell \geq 2$ in which servers do not collude. More generally, a PIR scheme is called t-private if any coalition of t servers learns no information on τ.

3rd Conference on Information-Theoretic Cryptography (ITC 2022).
Editor: Dana Dachman-Soled; Article No. 1; pp. 1:1–1:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Since then, many ℓ-server PIR schemes have been developed to improve communication cost [1, 3, 4, 6, 7, 10, 11, 14, 22]. Currently, the most communication-efficient schemes are 1-private $2^{O(r)}$-server PIR schemes with sub-polynomial (in n) communication complexity $\mathcal{L}_n[1/r, O_r(1)]$ [6, 10], where $\mathcal{L}_n[s, c]$ denotes a function $\exp(c(\log n)^s(\log\log n)^{1-s})$ and the notation $O_r(\cdot)$ hides constants that depend on r only.[1] To achieve t-privacy for $t \geq 2$, Woodruff and Yekhanin [20] proposed a t-private ℓ-server PIR scheme with communication complexity $n^{\lfloor(2k-1)/t\rfloor^{-1}}\ell^{O(1)}$ for any $1 \leq k \leq \ell$.

As more servers are involved, there is a higher possibility that servers are malicious or fault, or that the databases are not updated simultaneously. It is then important to enable a client to detect or even correct errors when part of servers return false answers. Beimel and Stahl [5] introduced b-error correcting PIR, which enables a client to retrieve a correct value a_τ even if b (or less) servers return false answers. They showed that a b-error correcting PIR scheme can be generically obtained from any k-server PIR scheme if $b \leq (\ell - k)/2$ while the time complexity of error correction is proportional to $\binom{\ell}{k}$. Kurosawa [15] proposed a more time-efficient error correction algorithm specialized for the t-private PIR scheme in [20] and as a result, it performs $\lfloor(\ell - k)/2\rfloor$-error correction in polynomial time in ℓ for any $1 \leq k \leq \ell$.

However, as pointed out in [5], b-error correcting PIR is possible only if $b < \ell/2$. It is therefore important to consider a weaker notion of error detecting to tolerate more malicious servers. Specifically, we define $(1 - \epsilon)$-fully error detecting PIR as the one which enables a client to detect errors with probability $1 - \epsilon$ even if $\ell - 1$ out of ℓ answers are false. To the best of our knowledge, there are no fully error detecting PIR schemes in the literature except for the trivial scheme or the one implicitly used in [21] both of which have communication cost $O(n)$. This paper concerns the following problem:

> *Is there a t-private $(1 - \epsilon)$-fully error detecting ℓ-server PIR scheme with communication complexity $o(n)$? Is it possible to add error correction capability to it?*

1.1 Our Results

We first formalize the notion of $(1 - \epsilon)$-fully error detecting PIR. We then prove an impossibility result that there exists no 1-fully error detecting (i.e., $\epsilon = 0$) PIR scheme with $o(n)$ communication. Next, for $\epsilon > 0$, we construct 1-private $(1 - \epsilon)$-fully error detecting and $(\ell/2 - O(1))$-error correcting PIR schemes which has $n^{o(1)}$ communication. For $t \geq 2$, we also propose a t-private one which has $O(n^c)$ communication for some constant $c < 1$. Here, we ignore a factor of $\log \epsilon^{-1}$ in communication cost. Our constructions are based on the following technical contributions:

- We prove that the transformations [5], which add error correction capability to PIR schemes, preserve full error detection capability and even reduces the probability of failure.
- We construct $(1 - \epsilon)$-fully error detecting PIR schemes by modifying certain existing schemes.

In what follows we briefly discuss each of these contributions.

Formalization of Fully Error Detecting PIR. Let Π be a t-private ℓ-server PIR scheme. In our model, a set of at most $\ell - 1$ malicious servers T is partitioned into pairwise disjoint subsets $T = T_1 \cup \cdots \cup T_m$ such that $|T_h| \leq t$ for any h, and servers in each T_h can collude to

[1] If $c = O(1)$, $\mathcal{L}_n[1, c]$ is polynomial in n and $\mathcal{L}_n[0, c]$ is polylogarithmic in n. For $0 < s < 1$, $\mathcal{L}_n[s, c]$ is sub-polynomial in n.

generate their false answers. Our model is natural since due to the t-privacy, no malicious server is allowed to see more than t queries and hence its false answer should not depend on more than t queries. We say that Π is $(1-\epsilon)$-fully error detecting if a client can detect errors with probability $1-\epsilon$ for any $T = T_1 \cup \cdots \cup T_m$ satisfying the above condition. We prove that there exists no 1-fully error detecting (i.e., $\epsilon = 0$) PIR scheme with $o(n)$ communication (Theorem 13). This implies that it is necessary to consider $(1-\epsilon)$-fully error detecting PIR with $\epsilon > 0$.

Transformation to Increase Robustness of Fully Error Detecting PIR. To transform a k-server PIR scheme Π to an $\lfloor(\ell-k)/2\rfloor$-error correcting ℓ-server PIR scheme Π', Beimel and Stahl [5] presented a naive method, which executes an independent instance of Π for each group of k servers, and a more refined method, which uses perfect hash families.[2] We prove that the two transformation methods preserve full error detection capability and even reduces the probability of failure. Therefore, they can be used to add $\lfloor(\ell-k)/2\rfloor$-error correction capability to a fully error detecting PIR scheme.More specifically, the method using a perfect hash family transforms a 1-private $(1-\epsilon)$-fully error detecting k-server PIR scheme Π to a 1-private $(1-\epsilon)$-fully error detecting ℓ-server PIR scheme Π' (Theorem 14). The overhead in communication cost is $2^{O(k)}\ell\log\ell$. The naive method can be used to transform a t-private $(1-\epsilon)$-fully error detecting k-server PIR scheme Π to a t-private $(1-\epsilon^M)$-fully error detecting ℓ-server PIR scheme Π', where $M = \lceil(\ell-k+1)/(k+t-2)\rceil$ (Theorem 15). The communication cost of Π' is $\binom{\ell}{k}$ times larger than Π. Although the method in Theorem 14 is more communication-efficient for large k, the naive transformation in Theorem 15 has the following advantages:

- From any 1-private 2-server $(1-\epsilon)$-fully error detecting PIR scheme, we can obtain a 1-private ℓ-server $(1-\epsilon)$-fully error detecting one which has lower communication cost by a factor of $O(\log\ell)$ than if Theorem 14 is applied.
- It works for any $t \geq 1$, where t is the number of servers who can collude.

Constructions of Fully Error Detecting PIR Schemes.

1-Private two-server PIR scheme. Dvir and Gopi [10] showed a 1-private 2-server PIR scheme with communication complexity $\mathcal{L}_n[1/2, O(1)]$ by using a matching vector family and a kind of polynomial interpolation. Based on their scheme, we construct a 1-private $(1-\epsilon)$-fully error detecting 2-server PIR scheme with communication complexity $\mathcal{L}_n[1/2, O(1)] \cdot \log\epsilon^{-1}$ (Theorem 16). Our technical novelty is modifying the scheme [10] in such a way that a client chooses interpolation points at random and carefully analyzing its error detection capability. By applying the naive transformation in Theorem 15, we obtain a 1-private ℓ-server $(1-\epsilon)$-fully error detecting and $\lfloor(\ell-2)/2\rfloor$-error correcting PIR scheme with communication complexity $\mathcal{L}_n[1/2, O(1)] \cdot \ell\log\epsilon^{-1}$ (Corollary 17).

1-Private ℓ-server PIR scheme for larger ℓ. We show that the communication complexity of fully error detecting PIR can be further reduced by increasing the number of servers. We invoke a basic PIR scheme based on a matching vector family shown in [9], which uses Lagrange interpolation to retrieve a_τ. We carefully choose parameters for the matching vector family and let a client choose interpolation points at random. As a result, for any fixed $r \geq 2$, we obtain a 1-private $(1-\epsilon)$-fully error detecting k_r-server PIR scheme with communication complexity $\mathcal{L}_n[1/r, O_r(1)] \cdot \log\epsilon^{-1}$, where k_r is a constant

[2] We note that their method shown in [5, Section 3.1] is a special case of the latter based on perfect hash families.

depending on r (Corollary 19). By applying the transformation in Theorem 14, we obtain a 1-private $(1-\epsilon)$-fully error detecting and $\lfloor(\ell-k_r)/2\rfloor$-error correcting ℓ-server PIR scheme with communication complexity $\mathcal{L}_n[1/r, O_r(1)] \cdot 2^{O(k_r)}\ell \log \ell \log \epsilon^{-1}$ for any $\ell \geq k_r$ (Corollary 20). By setting $r = 3$, we obtain a $(1-\epsilon)$-fully error detecting ℓ-server PIR scheme with communication cost $\mathcal{L}_n[1/3, O(1)] \cdot \ell \log \ell \log \epsilon^{-1}$ for $\ell \geq 2^{17}$.

t-Private ℓ-server PIR scheme for $t \geq 2$ and $\ell \geq 2$. Our construction for $t \geq 2$ is based on the best known t-private $\lfloor(\ell-k)/2\rfloor$-error correcting ℓ-server PIR scheme [20] with communication complexity $O(dn^d \ell \log \ell)$, where $1 \leq k \leq \ell$ and $d = \lfloor(2k-1)/t\rfloor$. Their scheme uses Hermite interpolation [17] to retrieve a_τ. By choosing interpolation points randomly, we obtain a t-private $(1-\epsilon)$-fully error detecting and $\lfloor(\ell-k)/2\rfloor$-error correcting ℓ-server PIR scheme with communication complexity $O(dn^{1/d}\ell \log \ell \log \epsilon^{-1})$ (Theorem 21). We note that the polynomial-time error correction algorithm [15], which was originally proposed for the scheme [20] with no error detection, is applicable to our fully error detecting scheme. Hence, this scheme achieves error correction without the transformations in Theorems 14 and 15.

1.2 Related Work

Beimel and Stahl [5] introduced (k,ℓ)-robust PIR, which allows a client to retrieve a correct value from answers of any k out of ℓ honest servers. They presented generic transformations from any k-server PIR scheme to (k,ℓ)-robust PIR scheme. They also showed that any (k,ℓ)-robust PIR scheme achieves b-error correction for $b \leq (\ell-k)/2$ while the time complexity of error correction is proportional to $\binom{\ell}{k}$. Any (k,ℓ)-robust PIR scheme implies an ℓ-server PIR scheme that detects errors if at most $\ell-k$ servers are malicious, by letting the client recover data from answers of every k servers and check the consistency. However, it cannot be better than the trivial scheme if there are $\ell-1$ malicious servers.

Yang, Xu, and Bennett [21] proposed a PIR scheme which achieves b-error correction by performing error detection for all subsets of servers of size $b+1$ for $b = \lfloor(\ell-1)/2\rfloor$. Their scheme satisfies our definition of fully error detecting PIR. However, the communication complexity is $O(n)$ and it is not better than the trivial scheme downloading the whole database. Although it can be reduced to $O(\sqrt{n})$ by the balancing technique of [7], the communication complexities of our schemes are still lower than theirs.

Goldberg [12] proposed a list decodable ℓ-server PIR scheme with communication complexity $O(\sqrt{n})$, in which a client outputs a list including a correct value instead of just one. However, the scheme tolerates at most $\ell - \lfloor\sqrt{\ell}\rfloor$ malicious servers and hence it cannot detect errors in the presence of $\ell-1$ malicious servers. Devet, Goldberg, and Heninger [8] considered a different scenario where a client performs multiple queries and runs a decoding algorithm on multiple answers simultaneously. In this setting, they proposed a list decodable ℓ-server PIR scheme for $\ell - O(1)$ malicious servers with communication complexity $O(\sqrt{n})$.

Sun and Jafar [18, 19] and Banawan and Ulukus [2] considered error correction in the setting where the size of each block of a database is very large, and hence only the download cost is of interest.

2 Preliminaries

Notations. For $m \in \mathbb{N}$, define $[m] = \{1,\dots,m\}$. For a vector $\boldsymbol{x}$, let x_i denote the i-th entry of $\boldsymbol{x}$. Let $f \in \mathbb{F}_q[X_1,\dots,X_m]$ be an m-variate polynomial over a finite field $\mathbb{F}_q$ of size q. We say that f is a degree-d polynomial if its total degree is at most d. Define the partial derivative of f with respect to X_j as

$$\partial_{X_j} f = \sum_{\boldsymbol{e}=(e_i)_{i\in[m]}\in I} c_{\boldsymbol{e}} e_j X_j^{e_j-1} \prod_{i\in[m]\setminus\{j\}} X_i^{e_i}$$

if $f = \sum_{\boldsymbol{e}\in I} c_{\boldsymbol{e}} \prod_{i\in[m]} X_i^{e_i}$, where $c_{\boldsymbol{e}} \in \mathbb{F}_q$ and I is a finite set of m-tuples of non-negative integers. For a univariate polynomial f, we denote by ∂f the derivative of f with respect to its unique variable. We write $u \leftarrow_{\$} \mathcal{U}$ if u is randomly chosen from a set $\mathcal{U}$. For two vectors $\boldsymbol{x} = (x_i)_{i\in[m]}, \boldsymbol{y} = (y_i)_{i\in[m]}$ over a ring $\mathcal{U}$, we define $\langle \boldsymbol{u}, \boldsymbol{v}\rangle = \sum_{i\in[h]} u_i v_i$ and $\mathsf{wt}(\boldsymbol{u}) = |\{i \in [m] : u_i \neq 0\}|$. Let $\mathcal{L}_n[s, c]$ denote the function of n defined as

$$\mathcal{L}_n[s, c] = \exp(c(\log n)^s(\log n)^{1-s}),$$

where $0 \leq s \leq 1$ and $c > 0$. Note that if $c = O(1)$, $\mathcal{L}_n[1, c]$ is polynomial in n and $\mathcal{L}_n[0, c]$ is polylogarithmic in n. For $0 < s < 1$, $\mathcal{L}_n[s, c]$ is sub-polynomial in n.

2.1 Lagrange and Hermite Interpolation

Lagrange interpolation recovers a polynomial using its values on given points. Let $\ell \in \mathbb{N}$ and $\mathbb{F}_p$ be a prime field such that $p \geq \ell + 1$. Let $\alpha_1, \dots, \alpha_\ell$ be ℓ pairwise distinct non-zero elements of $\mathbb{F}_p$ and let $y_j \in \mathbb{F}_p$ for each $j \in [\ell]$. Then, there exists an explicit formula for finding a unique polynomial $g \in \mathbb{F}_p[X]$ such that $\deg g \leq \ell - 1$ and $g(\alpha_j) = y_j$ for all $j \in [\ell]$.

Hermite interpolation is a generalization of Lagrange interpolation, which recovers a polynomial using its derivatives and values on given points. Let $y_{j,w} \in \mathbb{F}_p$ for each $j \in [\ell]$ and $w \in \{0, 1\}$. Then, there exists an explicit formula for finding a unique polynomial $g \in \mathbb{F}_p[X]$ such that $\deg g \leq 2\ell - 1$ and $g(\alpha_j) = y_{j,0}$ and $\partial g(\alpha_j) = y_{j,1}$ for all $j \in [\ell]$ [17].

3 Private Information Retrieval (PIR)

3.1 Definitions

In an ℓ-server PIR scheme, each server has a copy of a database $\boldsymbol{a} = (a_1, \dots, a_n) \in \{0, 1\}^n$. A client can obtain a_τ by interacting with ℓ servers without revealing any information on τ to the servers.

▶ **Definition 1** (Syntax). *An ℓ-server PIR scheme Π consists of three algorithms $\Pi = (\mathcal{Q}, \mathcal{A}, \mathcal{R})$, where $\mathcal{Q}$ is probabilistic while $\mathcal{A}$ and $\mathcal{R}$ are deterministic.*

- *A query algorithm $\mathcal{Q}$ takes $\tau \in [n]$ as input and outputs ℓ queries $\mathsf{que}_1, \dots, \mathsf{que}_\ell$ together with auxiliary information aux. A client computes*

  $$\mathcal{Q}(\tau; r) \to (\mathsf{que}_1, \dots, \mathsf{que}_\ell; \mathsf{aux})$$

  and then sends que_i to the i-th server for $i \in [\ell]$, where r is a random string.
- *An answer algorithm $\mathcal{A}$ takes as input an index $i \in [\ell]$, a query que_i and a database $\boldsymbol{a} = (a_1, \dots, a_n) \in \{0, 1\}^n$, and outputs an answer ans_i. The i-th server computes*

  $$\mathcal{A}(i, \mathsf{que}_i, \boldsymbol{a}) \to \mathsf{ans}_i$$

  and then returns ans_i to the client.

- *A reconstruction algorithm $\mathcal{R}$ takes as input ℓ answers $\mathsf{ans}_1, \dots, \mathsf{ans}_\ell$ and auxiliary information aux, and outputs $\widetilde{a} \in \{0,1\}$. The client computes*

  $$\mathcal{R}(\mathsf{ans}_1, \dots, \mathsf{ans}_\ell; \mathsf{aux}) \to \widetilde{a}$$

  and outputs $\widetilde{a}$.

We say that Π is correct if for any database $\boldsymbol{a} = (a_1, \dots, a_n) \in \{0,1\}^n$ and any $\tau \in [n]$, it holds that $\mathcal{R}(\mathsf{ans}_1, \dots, \mathsf{ans}_\ell; \mathsf{aux}) = a_\tau$, where $(\mathsf{que}_1, \dots, \mathsf{que}_\ell; \mathsf{aux}) \leftarrow \mathcal{Q}(\tau)$ and $\mathsf{ans}_i \leftarrow \mathcal{A}(i, \mathsf{que}_i, \boldsymbol{a})$ for $i \in [\ell]$. The (total) communication complexity of Π is given by $\sum_{i=1}^{\ell} |\mathsf{que}_i| + \sum_{i=1}^{\ell} |\mathsf{ans}_i|$, where $|\mathsf{que}_i|$ and $|\mathsf{ans}_i|$ are the bit lengths of que_i and ans_i, respectively.

We say that an ℓ-server PIR scheme is t-private if any t servers learn no information on the client's secret index τ even if they collude. Formally,

▶ **Definition 2** (t-Privacy). *An ℓ-server PIR scheme $\Pi = (\mathcal{Q}, \mathcal{A}, \mathcal{R})$ is said to be t-private if for any t indices $i_1, \dots, i_t \in [\ell]$ and any $\tau, \tau' \in [n]$, the joint distributions of $(\mathsf{que}_{i_1}, \dots, \mathsf{que}_{i_t})$ and $(\mathsf{que}'_{i_1}, \dots, \mathsf{que}'_{i_t})$ are perfectly identical, where $(\mathsf{que}_1, \dots, \mathsf{que}_\ell; \mathsf{aux}) \leftarrow \mathcal{Q}(\tau)$ and $(\mathsf{que}'_1, \dots, \mathsf{que}'_\ell; \mathsf{aux}') \leftarrow \mathcal{Q}(\tau')$.*

Beimel and Stahl [5] introduced the notion of robust and error correcting PIR.

▶ **Definition 3** (Robust PIR). *An ℓ-server PIR scheme Π is said to be (k, ℓ)-robust if for any $K = \{i_1, \dots, i_k\} \subseteq [\ell]$, there exists an algorithm $\mathcal{R}_K$ that correctly computes a_τ from k answers $\mathsf{ans}_{i_1}, \dots, \mathsf{ans}_{i_k}$.*

▶ **Definition 4** (Error correcting PIR). *An ℓ-server PIR scheme $\Pi = (\mathcal{Q}, \mathcal{A}, \mathcal{R})$ is said to be b-error correcting if $\mathcal{R}$ can correctly compute a_τ even if b (or less) answers among $(\mathsf{ans}_1, \dots, \mathsf{ans}_\ell)$ are false.*

In [5, Theorem 6.2], it is shown that a (k, ℓ)-robust PIR scheme is $\lfloor (\ell - k)/2 \rfloor$-error correcting while the time complexity of error correction is proportional to $\binom{\ell}{k}$ since $\mathcal{R}$ needs to perform $\mathcal{R}_K$ for all subsets K of size k.

3.2 Known Transformation from k-Server PIR to (k, ℓ)-Robust PIR

Beimel and Stahl [5] showed a generic transformation from any k-server PIR scheme Π into a (k, ℓ)-robust PIR scheme Π' for any $\ell > k$. Their transformation is based on a minimal perfect hash family.

▶ **Definition 5.** *Let $\ell \geq k$. An (ℓ, k)-minimal perfect hash family $\mathcal{H} = \{h_1, \dots, h_w\}$ is a family of functions of the form $h_j : [\ell] \to [k]$ such that for each $A \subseteq [\ell]$ of size k, there exists an index j such that $h_j(A) = [k]$.*

Let $\Pi = (\mathcal{Q}, \mathcal{A}, \mathcal{R})$ be any k-server PIR scheme and $\mathcal{H} = \{h_1, \dots, h_w\}$ be an (ℓ, k)-minimal perfect hash family. Beimel and Stahl [5] construct a (k, ℓ)-robust PIR scheme $\Pi' = (\mathcal{Q}', \mathcal{A}', \mathcal{R}'_K)$ where $\mathcal{R}'_K$ is a reconstruction algorithm for a set K of k servers, as follows:

$\boldsymbol{\mathcal{Q}'(\tau)}$. To obtain a_τ, a client executes w times Π independently and generates w query vectors $\mathsf{que}^{(j)} = (\mathsf{que}_1^{(j)}, \dots, \mathsf{que}_k^{(j)})$, $j \in [w]$ along with auxiliary information $\mathsf{aux}^{(j)}$. For each $i \in [\ell]$, the client sends $\mathsf{que}_i = (\mathsf{que}_{h_1(i)}^{(1)}, \dots, \mathsf{que}_{h_w(i)}^{(w)})$ to the i-th server.

$\boldsymbol{\mathcal{A}'(i, \mathsf{que}_i, a)}$. The i-th server replies to each query $q = \mathsf{que}_{h_j(i)}^{(j)}$ for $j \in [w]$ as the $h_j(i)$-th server would reply to q in the original k-server PIR scheme Π. The i-th server returns $\mathsf{ans}_i = (\mathcal{A}(h_j(i), \mathsf{que}_{h_j(i)}^{(j)}, \boldsymbol{a}))_{j \in [w]}$ to the client.

$\mathcal{R}'_K(\mathsf{ans}_{i_1}, \dots, \mathsf{ans}_{i_k}; \mathsf{aux})$. If the client receives answers from a set of k servers $K = \{i_1, \dots, i_k\} \subseteq [\ell]$, let $h_j \in \mathcal{H}$ be a function such that $h_j(K) = [k]$. Due to the correctness of Π, the client can obtain a_τ from $\{\mathcal{A}(h_j(i), \mathsf{que}^{(j)}_{h_j(i)}, \boldsymbol{a})\}_{i \in K}$ and $\mathsf{aux}^{(j)}$.

A construction of $\mathcal{H}$ with $w = 2^{O(k)} \log \ell$ is also given in [5]. Therefore, the communication complexity of Π' is $c \cdot 2^{O(k)} \ell \log \ell$ if Π has communication complexity c per server. Since each execution of Π is independent, if Π is t-private, so is Π'.

4 Matching Vector Family

4.1 Definitions and Constructions

▶ **Definition 6.** *Let $m \in \mathbb{Z}$ and $S \subseteq \mathbb{Z}_m \setminus \{0\}$. We say that $\mathcal{U} = (\boldsymbol{u}_1, \dots, \boldsymbol{u}_n)$ and $\mathcal{V} = (\boldsymbol{v}_1, \dots, \boldsymbol{v}_n)$, where $\boldsymbol{u}_i, \boldsymbol{v}_i \in \mathbb{Z}_m^h$, form an S-matching vector family if the following condition is satisfied:*

- *$\langle \boldsymbol{u}_i, \boldsymbol{v}_i \rangle = 0$ for every $i \in [n]$;*
- *$\langle \boldsymbol{u}_i, \boldsymbol{v}_j \rangle \in S$ for every $i \neq j$.*

We say that an S-matching vector family is d-bounded if $s \leq d$ for all $s \in S$ in terms of the usual order on $\mathbb{Z}$.

There exists an explicit construction of a matching vector family.

▶ **Proposition 7** ([13]). *Let $p < q$ be two primes and set $m = pq$. For any integer $n > 1$, there exist a constant θ_m depending on m only and an S-matching vector family $\mathcal{U} = \mathcal{V} = (\boldsymbol{u}_1, \dots, \boldsymbol{u}_n)$ over $\mathbb{Z}_m^h$ such that $h = \mathcal{L}_n[1/2, \theta_m]$ and $S = \{p, q, p+q\}$.*

There also exists an explicit construction of a bounded matching vector family.

▶ **Proposition 8** ([9]). *Let $p_1, \dots, p_r$ be $r \geq 2$ pairwise distinct primes and set $m = p_1 \cdots p_r$. Let u, w be positive integers such that $u \geq w$. For each $i \in [r]$, let e_i be the smallest integer such that $p_i^{e_i} > w^{1/r}$. Set $c = \max_{i \in [r]} p_i^{e_i}$ and $d = m \sum_{i \in [r]} p_i^{-1}$. Then, there exists a d-bounded matching vector family of size n over $\mathbb{Z}_m^h$ such that $n = \binom{u}{w}$ and $h = \binom{u}{\leq c} := \sum_{i=0}^{c} \binom{u}{i}$.*

4.2 Basic PIR Based on a Matching Vector Family

Following [9], we can construct a $(d+1)$-server PIR scheme $\Pi = (\mathcal{Q}, \mathcal{A}, \mathcal{R})$ based on a d-bounded S-matching vector family $\mathcal{U} = (\boldsymbol{u}_1, \dots, \boldsymbol{u}_n)$ and $\mathcal{V} = (\boldsymbol{v}_1, \dots, \boldsymbol{v}_n)$ over $\mathbb{Z}_m^h$ as follows. Let q be a prime such that $q = 1 \bmod m$. Let γ be an m-th root of unity of $\mathbb{F}_q$. Let $\alpha_1, \dots, \alpha_{d+1}$ be distinct elements of $\mathbb{Z}_m$. Let $\boldsymbol{a} = (a_1, \dots, a_n) \in \{0,1\}^n$ be a database.

$\mathcal{Q}(\tau)$. To obtain a_τ, the client chooses $\boldsymbol{w} \in \mathbb{Z}_m^h$ randomly. He then computes $\boldsymbol{\rho}_i = (\rho_{i1}, \dots, \rho_{ih}) = \boldsymbol{w} + \alpha_i \boldsymbol{u}_\tau$, sends $\mathsf{que}_i = (\gamma^{\rho_{i1}}, \dots, \gamma^{\rho_{ih}})$ to the i-th server for $i \in [d+1]$, and stores $\mathsf{aux} = \boldsymbol{w}$.

$\mathcal{A}(i, \mathsf{que}_i, a)$. The i-th server returns

$$\mathsf{ans}_i = \xi_i = \sum_{\sigma \in [n]} a_\sigma (\gamma^{\rho_{i1}})^{v_{\sigma 1}} \cdots (\gamma^{\rho_{ih}})^{v_{\sigma h}} = \sum_{\sigma \in [n]} a_\sigma \gamma^{\langle \boldsymbol{\rho}_i, \boldsymbol{v}_\sigma \rangle}$$

to the client for $i \in [d+1]$, where $v_{\sigma j}$ is the j-th coordinate of $\boldsymbol{v}_\sigma$.

$\mathcal{R}(\mathsf{ans}_1, \ldots, \mathsf{ans}_{d+1}; \mathsf{aux})$. The client computes a_τ from $\xi_1, \ldots, \xi_{d+1}$ as follows. Note that

$$\begin{aligned}\xi_i &= \sum_{\sigma \in [n]} a_\sigma \gamma^{\langle \boldsymbol{\rho}_i, \boldsymbol{v}_\sigma \rangle} \\ &= \sum_{\sigma \in [n]} a_\sigma \gamma^{\langle \boldsymbol{w} + \alpha_i \boldsymbol{u}_\tau, \boldsymbol{v}_\sigma \rangle} \\ &= a_\tau \gamma^{\langle \boldsymbol{w}, \boldsymbol{v}_\tau \rangle} + \sum_{\sigma \neq \tau} a_\sigma \gamma^{\langle \boldsymbol{w}, \boldsymbol{v}_\sigma \rangle} \gamma^{\alpha_i \langle \boldsymbol{u}_\tau, \boldsymbol{v}_\sigma \rangle} \\ &= c_0 + \sum_{s \in S} c_s (\gamma^{\alpha_i})^s\end{aligned}$$

where $c_s = \sum_{\sigma \in [n] : \langle \boldsymbol{u}_\tau, \boldsymbol{v}_\sigma \rangle = s} a_\sigma \gamma^{\langle \boldsymbol{w}, \boldsymbol{v}_i \rangle}$ for each $s \in S$ and $c_0 = a_\tau \gamma^{\langle \boldsymbol{w}, \boldsymbol{u}_\tau \rangle}$. Let $f(x) = c_0 + \sum_{s \in S} c_s x^s$. The degree of f is at most d and $\xi_i = f(\gamma^{\alpha_i})$ for $i \in [d+1]$. By using Lagrange interpolation, the client can compute $f(0) = c_0 = a_\tau \gamma^{\langle \boldsymbol{w}, \boldsymbol{u}_\tau \rangle}$ from $\xi_1, \ldots, \xi_{d+1}$ and obtain a_τ.

5 Formalization of Fully Error Detecting PIR

5.1 Definitions

We formally define error detecting PIR. In a t-private PIR scheme, any t servers learn no information on τ even if they collude, where τ is the secret index of the client. In a t-private error detecting ℓ-server PIR scheme, we require that the client can detect errors even if $\ell - 1$ servers return false answers. We allow only t servers to collude when computing their false answers, which is the same condition as that for t-privacy. Namely a set of malicious servers T is given by a union of pairwise disjoint subsets $T = T_1 \cup \cdots \cup T_m$ in such a way that $|T| \leq \ell - 1$, $|T_h| \leq t$ for $h \in [m]$ and the servers in each T_h can collude. We formalize such malicious servers by using a tampering function f such that

$$f(\mathsf{que}_1, \ldots, \mathsf{que}_\ell, \boldsymbol{a}) = (\widetilde{\mathsf{ans}}_1, \ldots, \widetilde{\mathsf{ans}}_\ell), \tag{1}$$

where que_i is a query sent to the i-th server and $\boldsymbol{a} = (a_1, \ldots, a_n) \in \{0,1\}^n$ is a database.

▶ **Definition 9** (Tampering function). *Let $T_1, \ldots, T_m \subseteq [\ell]$ be pairwise disjoint subsets. We say that a function f given by Eq. (1) is a tampering function for an ℓ-server PIR scheme $\Pi = (\mathcal{Q}, \mathcal{A}, \mathcal{R})$ with respect to $(T_1, \ldots, T_m)$ if for each $i \in [\ell]$, it holds that*

$$\widetilde{\mathsf{ans}}_i = \begin{cases} \mathcal{A}(i, \mathsf{que}_i, \boldsymbol{a}), & \text{if } i \notin T_1 \cup \cdots \cup T_m, \\ f_i(\{\mathsf{que}_{i'}\}_{i' \in T_j}, \boldsymbol{a}), & \text{if } i \in T_j \text{ for some } j \in [m], \end{cases} \tag{2}$$

for some function f_i. We denote the family of all such tampering functions by $\mathcal{F}^{\Pi}_{T_1, \ldots, T_m}$.

▶ **Definition 10** (Error detecting PIR). *We say that an ℓ-server PIR scheme $\Pi = (\mathcal{Q}, \mathcal{A}, \mathcal{R})$ is $(1-\epsilon)$-error detecting with respect to $(T_1, \ldots, T_m)$ if Π is correct and*

$$\Pr[\mathsf{ED}_\Pi(\boldsymbol{a}, \tau, f) = 1] \geq 1 - \epsilon$$

for any database $\boldsymbol{a} = (a_1, \ldots, a_n) \in \{0,1\}^n$, any $\tau \in [n]$ and any $f \in \mathcal{F}^{\Pi}_{T_1, \ldots, T_m}$, where the experiment $\mathsf{ED}_\Pi(\boldsymbol{a}, \tau, f)$ is defined as follows:

1. *Let $(\mathsf{que}_1, \ldots, \mathsf{que}_\ell; \mathsf{aux}) \leftarrow \mathcal{Q}(\tau)$;*
2. *Let $f(\mathsf{que}_1, \ldots, \mathsf{que}_\ell, \boldsymbol{a}) = (\widetilde{\mathsf{ans}}_1, \ldots, \widetilde{\mathsf{ans}}_\ell)$;*
3. *Return 1 if $\mathcal{R}(\widetilde{\mathsf{ans}}_1, \ldots, \widetilde{\mathsf{ans}}_\ell; \mathsf{aux}) \in \{a_\tau, \perp\}$ and return 0 otherwise.*

We say that a t-private ℓ-server PIR scheme Π is $(1-\epsilon)$-fully error detecting if it is $(1-\epsilon)$-error detecting with respect to any tuple of pairwise disjoint subsets $(T_1, \ldots, T_m)$ such that $|T_1 \cup \cdots \cup T_m| \leq \ell - 1$ and $|T_i| \leq t$ for $i \in [m]$.

▶ Remark 11. Although tampering functions are supposed to be deterministic, it can be seen that they capture randomized behavior of malicious servers. This is because the success probability is considered over a random string of $\mathcal{Q}$, which is independent of servers' randomness, and also because servers are assumed to be computationally unbounded.

▶ Remark 12. In a t-private fully error detecting PIR scheme, incorrect answers are allowed to depend on at most t queries. In particular, if $t = 1$, this means that the incorrect answers are generated independently. We note that this somewhat restricted adversarial model is still practically important. For example, consider a situation where a database $\boldsymbol{a}$ is updated frequently. If an honest server i has an old database $\boldsymbol{b} \neq \boldsymbol{a}$, then it returns an incorrect answer $\mathcal{A}(i, \mathsf{que}_i, \boldsymbol{b})$. Such errors can be detected by a 1-private fully error detecting PIR scheme.

5.2 Impossibility of 1-Fully Error Detecting PIR

The trivial scheme clearly achieves 1-full error detection, i.e., $\epsilon = 0$. Theorem 13 shows that we cannot do better than the trivial scheme in the case of $\epsilon = 0$.

▶ **Theorem 13.** *Let $\Pi = (\mathcal{Q}, \mathcal{A}, \mathcal{R})$ be a 1-private 1-fully error detecting ℓ-server PIR scheme for a universe of databases $\{0,1\}^n$. Then, the bit length of an answer of any server is at least n.*

Proof. Suppose that $\mathcal{A}$ outputs a c-bit string and the set of random strings for $\mathcal{Q}$ is $\{0,1\}^\rho$. We show that $c \geq n$. Let $r^{(1)} \in \{0,1\}^\rho$ be any random string for $\mathcal{Q}$ and let $(q_1^{(1)}, \ldots, q_\ell^{(1)}; \mathsf{aux}^{(1)}) = \mathcal{Q}(1; r^{(1)})$. Since Π is 1-private, for any $\tau \in [n] \setminus \{1\}$, there exists $r^{(\tau)} \in \{0,1\}^\rho$ such that $q_1^{(\tau)} = q_1^{(1)}$, where $(q_1^{(\tau)}, \ldots, q_\ell^{(\tau)}; \mathsf{aux}^{(\tau)}) = \mathcal{Q}(\tau; r^{(\tau)})$. We define $q_1 := q_1^{(1)} = q_1^{(2)} = \cdots = q_1^{(n)}$.

We define a function $\phi : \{0,1\}^n \to \{0,1\}^c$ as $\phi(\boldsymbol{a}) = \mathcal{A}(1, q_1, \boldsymbol{a})$. It is sufficient to show that ϕ is injective. Assume that $\mathcal{A}(1, q_1, \boldsymbol{a}) = \mathcal{A}(1, q_1, \boldsymbol{b})$ for some $\boldsymbol{a} \neq \boldsymbol{b} \in \{0,1\}^n$. Let $\tau \in [n]$ be such that $a_\tau \neq b_\tau$. Then, we have that $\mathcal{A}(1, q_1^{(\tau)}, \boldsymbol{a}) = \mathcal{A}(1, q_1^{(\tau)}, \boldsymbol{b})$. Set $T = [\ell] \setminus \{1\}$. Let $f \in \mathcal{F}^{\Pi}_{\{2\},\ldots,\{\ell\}}$ be any tampering function such that

$$f(q_1^{(\tau)}, q_2^{(\tau)}, \ldots, q_\ell^{(\tau)}, \boldsymbol{a}) = (\mathcal{A}(1, q_1^{(\tau)}, \boldsymbol{a}), (\mathcal{A}(i, q_i^{(\tau)}, \boldsymbol{b}))_{i \in T}).$$

Consider the experiment $\mathsf{ED}_\Pi(\boldsymbol{a}, \tau, f)$ in Definition 10. If $r^{(\tau)}$ is chosen, we have that $\mathcal{Q}(\tau; r^{(\tau)}) = (q_1^{(\tau)}, \ldots, q_\ell^{(\tau)}; \mathsf{aux}^{(\tau)})$ at Step 1. At Step 2, it holds that $\widetilde{\mathsf{ans}}_1 = \mathcal{A}(1, q_1^{(\tau)}, \boldsymbol{a})$ and $\widetilde{\mathsf{ans}}_i = \mathcal{A}(i, q_i^{(\tau)}, \boldsymbol{b})$ for $i \in T$. Then, $\mathsf{ED}_\Pi(\boldsymbol{a}, \tau, f)$ returns 0 since

$$\begin{aligned}\mathcal{R}(\widetilde{\mathsf{ans}}_1, (\widetilde{\mathsf{ans}}_i)_{i \in T}; \mathsf{aux}^{(\tau)}) &= \mathcal{R}(\mathcal{A}(1, q_1^{(\tau)}, \boldsymbol{a}), (\mathcal{A}(i, q_i^{(\tau)}, \boldsymbol{b}))_{i \in T}; \mathsf{aux}^{(\tau)}) \\ &= \mathcal{R}(\mathcal{A}(1, q_1^{(\tau)}, \boldsymbol{b}), (\mathcal{A}(i, q_i^{(\tau)}, \boldsymbol{b}))_{i \in T}; \mathsf{aux}^{(\tau)}) \\ &= b_\tau \notin \{a_\tau, \bot\}.\end{aligned}$$

Hence $\Pr[\mathsf{ED}_\Pi(\boldsymbol{a}, \tau, f) = 0] \geq \Pr\left[r^{(\tau)} \leftarrow_{\$} \{0,1\}^\rho\right] > 0$, which contradicts the 1-full error detection of Π. ◀

In view of Theorem 13, we will consider $(1-\epsilon)$-fully error detecting PIR with $\epsilon > 0$ in the following sections.

6 Transformation to Increase Robustness of Fully Error Detecting PIR

Beimel ans Stahl [5] presented two generic transformations from a k-server PIR scheme Π to (k,ℓ)-robust (and hence $\lfloor(\ell-k)/2\rfloor$-error correcting) PIR scheme Π'. One is based on a perfect hash family and the other simply executes Π for all groups of k servers. We prove that these methods preserve full error detection capability and can be used to add error correction capability to fully error detecting PIR schemes. Specifically, let Π be a $(1-\epsilon)$-fully error detecting k-server PIR scheme and Π' be an $\lfloor(\ell-k)/2\rfloor$-error correcting ℓ-server PIR scheme obtained by applying one of the transformations [5] to Π. We prove that Π' is $(1-\epsilon')$-fully error detecting for a certain $\epsilon' \leq \epsilon$. Although the method based on a perfect hash family is more communication-efficient for large k, the naive method has the following advantages:

- From any 1-private 2-server $(1-\epsilon)$-fully error detecting PIR scheme, we can obtain a 1-private ℓ-server $(1-\epsilon)$-fully error detecting one which has lower communication cost by a factor of $O(\log \ell)$ than if Theorem 14 is applied;
- It works for any $t \geq 1$, where t is the number of servers who can collude.

First, we consider the transformation based on a perfect hash family $\mathcal{H} = \{h_i : [\ell] \to [k] : i \in [w]\}$ (see Definition 5). In Π', a client executes w independent instances $\Pi_1, \dots, \Pi_w$ of Π and sends to Server $i \in [\ell]$ a query sent to Server $h_j(i) \in [k]$ in Π_j for all $j \in [w]$. We show that if Server i is honest, for any subset $S \subseteq [\ell]$ of size k containing i, the $(1-\epsilon)$-full error detection of Π' follows from that of Π_j, where j is an index such that $h_j(S) = [k]$. We note that this transformation does not provide a t-private $(1-\epsilon)$-fully error detecting ℓ-server PIR scheme for $t \geq 2$, that is, Π' is not necessarily t-private $(1-\epsilon)$-fully error detecting even if Π is. Roughly speaking, this is because the answer of a malicious server may depend on the query which is sent to an honest server. In summary the following theorem holds. See the full version for the proof.

▶ **Theorem 14.** *Suppose that there exists a 1-private $(1-\epsilon)$-fully error detecting k-server PIR scheme $\Pi = (\mathcal{Q}, \mathcal{A}, \mathcal{R})$ such that the communication complexity is c per server. Then, for any $\ell \geq k$, there exists a 1-private $(1-\epsilon)$-fully error detecting ℓ-server PIR scheme Π' with communication complexity $c \cdot 2^{O(k)}\ell \log \ell$. Furthermore, Π' is (k,ℓ)-robust and hence $\lfloor(\ell-k)/2\rfloor$-error correcting.*

Second, the naive transformation executes $p = \binom{\ell}{k}$ independent instances of Π for all groups $S_1, \dots, S_p$ of k servers. Let T be a set of $\ell - 1$ malicious servers. Suppose that T is partitioned into pairwise disjoint subsets $T = T_1 \cup \cdots \cup T_m$ such that $|T_h| \leq t$ for any h and servers in each T_h can collude. For $i \in [p]$, we show that if $|S_i \cap T| \leq k-1$, the $(1-\epsilon)$-full error detection of Π' follows from that of the instance of Π corresponding to S_i. More generally, based on the fact that a client's randomness for S_i, S_j $(i \neq j)$ are independent, we show that Π' is even $(1-\epsilon^M)$-fully error detecting if there are M subsets $S_{i_1}, \dots, S_{i_M}$ such that for every pair S_i, S_j, servers in S_i and in S_j do not receive the same query. We formalize that condition in a combinatorial way and show that the maximum number of M is at least $(\ell-k+1)/(k+t-2)$. As a result, Π' is t-private $(1-\epsilon')$-fully error detecting for $\epsilon' = \epsilon^M$. See Appendix A for the proof.

▶ **Theorem 15.** *Suppose that there exists a t-private $(1-\epsilon)$-fully error detecting k-server PIR scheme $\Pi = (\mathcal{Q}, \mathcal{A}, \mathcal{R})$ with communication complexity c. Let $\ell \geq k$. Set*

$$M' = \left\lceil \frac{\ell - k + 1}{k + t - 2} \right\rceil$$

and $\epsilon' = \epsilon^{M'}$. Then, there exists a t-private $(1-\epsilon')$-fully error detecting ℓ-server PIR scheme Π' with communication complexity $c \cdot \binom{\ell}{k}$. Furthermore, Π' is (k,ℓ)-robust and hence $\lfloor(\ell-k)/2\rfloor$-error correcting.

7 1-Private Fully Error Detecting PIR with Sub-polynomial Communication

In this section, we show 1-private $(1-\epsilon)$-fully error detecting ℓ-server PIR schemes Π_1' and Π_2' such that:

- For any $\ell \geq 2$, Π_1' is $\lfloor(\ell-2)/2\rfloor$-error correcting and has the communication complexity $\mathcal{L}_n[1/2, O(1)] \cdot \ell \log \epsilon^{-1}$.
- For any $r \geq 2$ and any $\ell \geq k := r^r 2^{r^2+2r-3} + 1$, Π_2' is $\lfloor(\ell-k)/2\rfloor$-error correcting and has the communication complexity $\mathcal{L}_n[1/r, 2^{O(r)}] \cdot \ell \log \ell \log \epsilon^{-1}$.

7.1 How to Construct Π_1'

In this subsection, we show a 1-private $(1-\epsilon)$-fully error detecting 2-server PIR scheme Π_1 with communication complexity $\mathcal{L}_n[1/2, O(1)] \cdot \log \epsilon^{-1}$. We can obtain Π_1' by applying Theorem 15 to Π_1. The scheme Π_1 is a variant of the 1-private 2-server PIR scheme of Dvir and Gopi [10] with communication complexity $\mathcal{L}_n[1/2, O(1)]$. Their scheme uses a matching vector family given by Proposition 7 with $p = 2$ and $q = 3$, and does a sort of polynomial interpolation with fixed points $\beta_1 = \gamma^0$ and $\beta_2 = \gamma^1$. On the other hand, Π_1 uses a matching vector family with $p \geq 3$ and $q = 1 \bmod p$, and does polynomial interpolation with random points $\beta_1 = \gamma^{\alpha_1}$ and $\beta_2 = \gamma^{\alpha_2}$ where α_1, α_2 are randomly chosen from $\{0, 1, \ldots, p-1\}$. A more formal description of Π_1 is shown in Figure 1. We obtain the following theorem. See Appendix B for the proof.

▶ **Theorem 16.** *For any $\epsilon > 0$, Π_1 is a 1-private $(1-\epsilon)$-fully error detecting 2-server PIR scheme with communication complexity $\mathcal{L}_n[1/2, O(1)] \cdot \log \epsilon^{-1}$.*

By applying Theorem 15 to the $(1-\epsilon)$-fully error detecting 2-server scheme Π_1, we obtain a $(1-\epsilon^{\Theta(\ell)})$-fully error detecting ℓ-server scheme Π_1', which means that the overhead in communication cost is only $O(\ell)$. Note that if we apply Theorem 14 to Π_1, then the overhead is $O(\ell \log \ell)$.

▶ **Corollary 17.** *Let $\epsilon > 0$. For any $\ell \geq 2$, there exists a 1-private $(1-\epsilon)$-fully error detecting and $\lfloor(\ell-2)/2\rfloor$-error correcting ℓ-server PIR scheme Π_1' for a universe of databases $\{0,1\}^n$ such that the communication complexity is*

$$\mathcal{L}_n[1/2, O(1)] \cdot \ell \log \epsilon^{-1} \tag{3}$$

and the time complexity of its reconstruction algorithm is polynomial in ℓ, n and $\log \epsilon^{-1}$.

7.2 How to Construct Π_2'

In this subsection, for $r \geq 2$ and $k = r^r 2^{r^2+2r-3} + 1$, we show a 1-private $(1-\epsilon)$-fully error detecting k-server PIR scheme Π_2 such that the communication complexity is $\mathcal{L}_n[1/r, 2^{O(r)}] \cdot \log \epsilon^{-1}$. We can obtain Π_2' by applying Theorem 14 to Π_2.

To construct Π_2, we first consider a variant of the 1-private k-server PIR scheme of Section 4.2 such that $\alpha_1, \ldots, \alpha_k$ are chosen randomly (Figure 2). The following theorem holds. See the full version for the proof.

Notations.

- A positive integer λ
- Two primes $p < q$ such that $q \equiv 1 \bmod p$
- $m = pq$ and a $\{p, q, p{+}q\}$-matching vector family $\mathcal{U}$, $\mathcal{V}$ over $\mathbb{Z}_m^h$ given by Proposition 7
- A primitive root $\delta \in \mathbb{F}_q^*$ and $\gamma = \delta^{(q-1)/p}$
- The ring homomorphism $\phi : \mathbb{Z}_m \to \mathbb{F}_q$ defined as $\phi(x) = x \bmod q$
- Polynomials $F_{\boldsymbol{a}} \in \mathbb{F}_q[z_1, \ldots, z_h]$ and $G_{\boldsymbol{a}} \in (\mathbb{F}_q^h)[z_1, \ldots, z_h]$ associated with $\boldsymbol{a} = (a_1, \ldots, a_n) \in \{0,1\}^n$ defined as $F_{\boldsymbol{a}}(z_1, \ldots, z_h) = \sum_{\tau \in [n]} a_\tau z_1^{v_{\tau 1}} \cdots z_h^{v_{\tau h}}$, and $G_{\boldsymbol{a}}(z_1, \ldots, z_h) = \sum_{\tau \in [n]} a_\tau \phi(\boldsymbol{v}_\tau) z_1^{v_{\tau 1}} \cdots z_h^{v_{\tau h}}$, where we assume $\boldsymbol{a} \in \mathbb{F}_q^n$, $v_{\tau j}$ is the j-th coordinate of $\boldsymbol{v}_\tau \in \mathbb{Z}_m^h$, and ϕ is applied on vectors entry-wise.

$\mathcal{Q}(\tau)$. Given an input $\tau \in [n]$:

1. For each $j \in [\lambda]$:
   - a. Choose two distinct elements $\alpha_1^{(j)}, \alpha_2^{(j)} \in \{0, 1, 2, \ldots, p-1\}$ randomly.
   - b. Choose $\boldsymbol{w}^{(j)} \leftarrow_\$ \mathbb{Z}_m^h$.
   - c. Let $(\rho_{i1}^{(j)}, \ldots, \rho_{ih}^{(j)}) = \boldsymbol{w}^{(j)} + \alpha_i^{(j)} \boldsymbol{u}_\tau \in \mathbb{Z}_m^h$ for $i \in \{1, 2\}$.
2. Output $\mathsf{que}_i = (\gamma^{\rho_{i1}^{(j)}}, \ldots, \gamma^{\rho_{ih}^{(j)}})_{j \in [\lambda]}$ for $i \in \{1, 2\}$ together with $\mathsf{aux} = ((\alpha_1^{(j)}, \alpha_2^{(j)})_{j \in [\lambda]}, (\boldsymbol{w}^{(j)})_{j \in [\lambda]}, \boldsymbol{u}_\tau, \boldsymbol{v}_\tau)$.

$\mathcal{A}(i, \mathsf{que}_i, a)$. Given $i \in \{1, 2\}$, a query que_i, and a database $\boldsymbol{a} \in \{0,1\}^n$:

1. Parse $\mathsf{que}_i = (\gamma^{\rho_{i1}^{(j)}}, \ldots, \gamma^{\rho_{ih}^{(j)}})_{j \in [\lambda]}$.
2. For each $j \in [\lambda]$, let $\xi_i^{(j)} = F_{\boldsymbol{a}}(\gamma^{\rho_{i1}^{(j)}}, \ldots, \gamma^{\rho_{ih}^{(j)}})$ and $\boldsymbol{\zeta}_i^{(j)} = G_{\boldsymbol{a}}(\gamma^{\rho_{i1}^{(j)}}, \ldots, \gamma^{\rho_{ih}^{(j)}})$.
3. Output $\mathsf{ans}_i = (\xi_i^{(j)}, \boldsymbol{\zeta}_i^{(j)})_{j \in [\lambda]}$.

$\mathcal{R}(\widetilde{\mathsf{ans}}_1, \widetilde{\mathsf{ans}}_2; \mathsf{aux})$. Given two answers $\widetilde{\mathsf{ans}}_i = (\widetilde{\xi}_i^{(j)}, \widetilde{\boldsymbol{\zeta}}_i^{(j)})_{j \in [\lambda]} \in (\mathbb{F}_q^{h+1})^\lambda$ and auxiliary information $\mathsf{aux} = ((\alpha_1^{(j)}, \alpha_2^{(j)})_{j \in [\lambda]}, (\boldsymbol{w}^{(j)})_{j \in [\lambda]}, \boldsymbol{u}_\tau, \boldsymbol{v}_\tau)$:

1. Let $\mathcal{L} = \emptyset$.
2. For each $j \in [\lambda]$:
   - a. Let $\widetilde{\eta}_i^{(j)} = \langle \phi(\boldsymbol{u}_\tau), \widetilde{\boldsymbol{\zeta}}_i^{(j)} \rangle$ and $\beta_i^{(j)} = \gamma^{\alpha_i^{(j)}}$ for $i \in \{1, 2\}$.
   - b. Define an invertible matrix $M^{(j)}$ as
   $$M^{(j)} = \begin{pmatrix} 1 & 1 & \beta_1^{(j)} & \beta_1^{(j)} \\ 0 & p & 0 & p\beta_1^{(j)} \\ 1 & 1 & \beta_2^{(j)} & \beta_2^{(j)} \\ 0 & p & 0 & p\beta_2^{(j)} \end{pmatrix} \in \mathbb{F}_q^{4 \times 4}.$$
   - c. Find $\widetilde{c}_0^{(j)}, \widetilde{c}_p^{(j)}, \widetilde{c}_q^{(j)}, \widetilde{c}_{p+q}^{(j)} \in \mathbb{F}_q$ such that
   $$M^{(j)} \begin{pmatrix} \widetilde{c}_0^{(j)} & \widetilde{c}_p^{(j)} & \widetilde{c}_q^{(j)} & \widetilde{c}_{p+q}^{(j)} \end{pmatrix}^\top = \begin{pmatrix} \widetilde{\xi}_1^{(j)} & \widetilde{\eta}_1^{(j)} & \widetilde{\xi}_2^{(j)} & \widetilde{\eta}_2^{(j)} \end{pmatrix}^\top.$$
   - d. Add $\widetilde{c}_0^{(j)} \gamma^{-\langle \boldsymbol{w}^{(j)}, \boldsymbol{v}_\tau \rangle}$ to $\mathcal{L}$ if $\widetilde{c}_0^{(j)} \gamma^{-\langle \boldsymbol{w}^{(j)}, \boldsymbol{v}_\tau \rangle} \in \{0, 1\}$. Otherwise, output $\perp$.
3. If $\mathcal{L} = \{s\}$ for some $s \in \{0, 1\}$, output s. Otherwise, output $\perp$.

Figure 1 A 1-private $(1 - \epsilon)$-fully error detecting 2-server PIR scheme Π_1.

▶ **Theorem 18.** *Given a d-bounded matching vector family $\mathcal{U}, \mathcal{V}$ of size n over $\mathbb{Z}_m^h$, there exists a 1-private $(1-\epsilon)$-fully error detecting k-server PIR scheme with communication complexity $O(kh\lambda \log m)$ for $k \geq d+1$ and*

$$\epsilon := \left(\frac{k-1}{m-k+1}\right)^{\lambda}.$$

In the full version, we present a matching vector family that is suitable for the scheme in Theorem 18. The following corollary can be obtained by combining Theorem 18 and that matching vector family. See the full version for the details.

▶ **Corollary 19.** *Let $r \geq 2$ and $\epsilon > 0$. Set $k = r^r 2^{r^2+2r-3} + 1$. Then, there exists a function $n_0 = n_0(r) = \exp(O(2^r r))$ such that the following holds: For any $n \geq n_0$, there exists a 1-private $(1-\epsilon)$-fully error detecting k-server PIR scheme Π_2 for a universe of databases $\{0,1\}^n$ such that the communication complexity is $\mathcal{L}_n[1/r, 2^{O(r)}] \cdot k \log \epsilon^{-1}$ and the time complexity of its reconstruction algorithm is polynomial in k, n and $\log \epsilon^{-1}$.*

By applying Theorem 14 to the k-server PIR scheme Π_2, we obtain a $(1-\epsilon)$-fully error detecting ℓ-server PIR scheme Π_2' while the overhead in communication cost is $2^{O(k)}\ell \log \ell$.

▶ **Corollary 20.** *Let $r \geq 2$ and $\epsilon > 0$. Set $k = r^r 2^{r^2+2r-3} + 1$. For a sufficiently large n (depending on r only) and any $\ell \geq k$, there exists a 1-private $(1-\epsilon)$-fully error detecting and $\lfloor (\ell-k)/2 \rfloor$-error correcting ℓ-server PIR scheme Π_2' for a universe of databases $\{0,1\}^n$ such that the communication complexity is*

$$\mathcal{L}_n[1/r, 2^{O(r)}] \cdot 2^{O(k)} \ell \log \ell \log \epsilon^{-1}$$

and the time complexity of its reconstruction algorithm is $\binom{\ell}{k} \cdot \mathsf{poly}\left(k, n, \log \epsilon^{-1}\right)$.

If we set $r = 3$, for any $\ell \geq 2^{17}$, there is a 1-private $(1-\epsilon)$-fully error detecting ℓ-server PIR scheme such that the communication complexity is $\mathcal{L}_n[1/3, O(1)] \cdot \ell \log \ell \log \epsilon^{-1}$, which is lower than the communication complexity (3) of Corollary 17 as functions of n.

8 t-Private Fully Error Detecting PIR with Polynomial Communication

In this section, we show a t-private $(1-\epsilon)$-fully error detecting and $\lfloor (\ell-k)/2 \rfloor$-error correcting PIR scheme Π with polynomial (in n) communication. Our scheme Π is the same as the t-private ℓ-server PIR scheme [20] except that it uses Hermite interpolation with random points α_i to achieve error detection (Figure 3). Note that Π has in common with the scheme [20] that a client can compute $\{(g(\alpha_i), \partial g(\alpha_i)) : i \in B\}$ for the unique polynomial g with $g(0) = a_\tau$, where B is a set of honest servers. This property implies that the polynomial-time error correction algorithm of [15] is applicable to Π. We obtain the following theorem. See the full version for the proof.

▶ **Theorem 21.** *Let $\epsilon > 0$ and $\ell \geq k \geq t \geq 1$. Set $d = \lfloor (2k-1)/t \rfloor$. Then, there exists a t-private $(1-\epsilon)$-fully error detecting and $\lfloor (\ell-k)/2 \rfloor$-error correcting ℓ-server PIR scheme for a universe of databases $\{0,1\}^n$ such that the communication complexity is $O(dn^{1/d}\ell \log \ell \log \epsilon^{-1})$ and the time complexity of its reconstruction algorithm is polynomial in ℓ, n and $\log \epsilon^{-1}$.*

Notations.

- Positive integers k, d, λ
- $m \geq k$ and a d-bounded S-matching vector family $\mathcal{U} = (\boldsymbol{u}_1, \dots, \boldsymbol{u}_n)$, $\mathcal{V} = (\boldsymbol{v}_1, \dots, \boldsymbol{v}_n)$ over $\mathbb{Z}_m^h$
- A prime field $\mathbb{F}_q$ such that $q \equiv 1 \bmod m$
- A primitive root $\delta \in \mathbb{F}_q^*$ and $\gamma = \delta^{(q-1)/m}$
- A polynomial $F_{\boldsymbol{a}}$ associated with $\boldsymbol{a} = (a_1, \dots, a_n) \in \{0,1\}^n$ defined as

$$F_{\boldsymbol{a}}(z_1, \dots, z_h) = \sum_{\tau \in [n]} a_\tau z_1^{v_{\tau 1}} \cdots z_h^{v_{\tau h}} \in \mathbb{F}_q[z_1, \dots, z_h],$$

  where we assume $\boldsymbol{a} \in \mathbb{F}_q^n$ and $v_{\tau j}$ is the j-th coordinate of $\boldsymbol{v}_\tau \in \mathbb{Z}_m^h$

$\mathcal{Q}(\tau)$. Given an input $\tau \in [n]$:

1. For each $j \in [\lambda]$:
   - a. Choose k pairwise distinct random elements $\alpha_1^{(j)}, \dots, \alpha_k^{(j)} \in \mathbb{Z}_m$.
   - b. Choose $\boldsymbol{w}^{(j)} \leftarrow_{\$} \mathbb{Z}_m^h$.
   - c. Let $(\rho_{i1}^{(j)}, \dots, \rho_{ih}^{(j)}) = \boldsymbol{w}^{(j)} + \alpha_i^{(j)} \boldsymbol{u}_\tau \in \mathbb{Z}_m^h$ for $i \in [k]$.
2. Output $\mathsf{que}_i = (\gamma^{\rho_{i1}^{(j)}}, \dots, \gamma^{\rho_{ih}^{(j)}})_{j \in [\lambda]}$ for $i \in [k]$ together with $\mathsf{aux} = ((\alpha_1^{(j)}, \dots, \alpha_k^{(j)})_{j \in [\lambda]}, (\boldsymbol{w}^{(j)})_{j \in [\lambda]}, \boldsymbol{v}_\tau)$.

$\mathcal{A}(i, \mathsf{que}_i, a)$. Given $i \in [k]$, a query que_i, and a database $\boldsymbol{a} \in \{0,1\}^n$:

1. Parse $\mathsf{que}_i = (\gamma^{\rho_{i1}^{(j)}}, \dots, \gamma^{\rho_{ih}^{(j)}})_{j \in [\lambda]}$.
2. For each $j \in [\lambda]$, let $\zeta_i^{(j)} = F_{\boldsymbol{a}}(\gamma^{\rho_{i1}^{(j)}}, \dots, \gamma^{\rho_{ih}^{(j)}})$.
3. Output $\mathsf{ans}_i = (\zeta_i^{(j)})_{j \in [\lambda]}$.

$\mathcal{R}(\widetilde{\mathsf{ans}}_1, \dots, \widetilde{\mathsf{ans}}_k; \mathsf{aux})$. Given k answers $\widetilde{\mathsf{ans}}_i = (\widetilde{\zeta}_i^{(j)})_{j \in [\lambda]} \in \mathbb{F}_q^\lambda$ and auxiliary information $\mathsf{aux} = ((\alpha_1^{(j)}, \dots, \alpha_k^{(j)})_{j \in [\lambda]}, (\boldsymbol{w}^{(j)})_{j \in [\lambda]}, \boldsymbol{v}_\tau)$:

1. Let $\mathcal{L} = \emptyset$.
2. Choose any subset $A \subseteq [k]$ of size $d + 1$.
3. For each $j \in [\lambda]$:
   - a. Compute a degree-d polynomial $\widetilde{g}^{(j)}(x) \in \mathbb{F}_q[x]$ such that $\widetilde{g}^{(j)}(\gamma^{\alpha_i^{(j)}}) = \widetilde{\zeta}_i^{(j)}$ for all $i \in A$, using Lagrange interpolation.
   - b. Add $\widetilde{g}^{(j)}(0)\gamma^{-\langle \boldsymbol{w}^{(j)}, \boldsymbol{v}_\tau \rangle}$ to $\mathcal{L}$ if $\widetilde{\zeta}_i^{(j)} = \widetilde{g}^{(j)}(\gamma^{\alpha_i^{(j)}})$ for all $i \notin A$ and $\widetilde{g}^{(j)}(0)\gamma^{-\langle \boldsymbol{w}^{(j)}, \boldsymbol{v}_\tau \rangle} \in \{0,1\}$. Otherwise, output $\perp$.
4. If $\mathcal{L} = \{s\}$ for some $s \in \{0,1\}$, output s. Otherwise, output $\perp$.

Figure 2 A 1-private $(1-\epsilon)$-fully error detecting k-server PIR scheme Π.

Notations.

- A prime field $\mathbb{F}_p$ such that $p \geq \ell + 1$.
- A positive integer k
- $d = \lfloor (2k-1)/t \rfloor$ and $m \in \mathbb{N}$ such that $\binom{m}{d} \geq n$
- An injection $E : [n] \to \{0,1\}^m$ such that $\mathsf{wt}(E(\tau)) = d$
- $\boldsymbol{u}^{E(\tau)} = \prod_{j:E(\tau)_j=1} u_j$ for $\boldsymbol{u} = (u_j)_{j\in[m]}$, where $E(\tau) = (E(\tau)_j)_{j\in[m]}$
- A polynomial $F_{\boldsymbol{a}}$ associated with $\boldsymbol{a} = (a_1, \ldots, a_n) \in \{0,1\}^n$ defined as

  $$F_{\boldsymbol{a}}(z_1, \ldots, z_m) = \sum_{\tau\in[n]} a_\tau \boldsymbol{z}^{E(\tau)} \in \mathbb{F}_p[z_1, \ldots, z_m],$$

  where $\boldsymbol{z} = (z_1, \ldots, z_m)$ and we assume $\boldsymbol{a} \in \mathbb{F}_p^n$

$\mathcal{Q}(\tau)$**.** Given an input $\tau \in [n]$:

1. Choose ℓ pairwise distinct random non-zero elements $\alpha_1, \ldots, \alpha_\ell \in \mathbb{F}_p$.
2. Choose $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_t \leftarrow_\$ \mathbb{F}_p^m$.
3. Let $\boldsymbol{q}_i = E(\tau) + \alpha_i \boldsymbol{v}_1 + \cdots + \alpha_i^t \boldsymbol{v}_t$ for $i \in [\ell]$.
4. Set $\boldsymbol{w}_i = \boldsymbol{v}_1 + 2\alpha_i \boldsymbol{v}_2 + \cdots + t\alpha^{t-1}\boldsymbol{v}_t$.
5. Output $\mathsf{que}_i = \boldsymbol{q}_i$ for $i \in [\ell]$ together with auxiliary information $\mathsf{aux} = ((\alpha_i)_{i\in[\ell]}, (\boldsymbol{w}_i)_{i\in[\ell]})$.

$\mathcal{A}(i, \mathsf{que}_i, a)$**.** Given $i \in [\ell]$, a query $\mathsf{que}_i = \boldsymbol{q}_i$, and a database $\boldsymbol{a} \in \{0,1\}^n$:

1. Let $\zeta_i^{(0)} = F_{\boldsymbol{a}}(\boldsymbol{q}_i)$ and $\zeta_{ij}^{(1)} = \partial_{z_j} F_{\boldsymbol{a}}(\boldsymbol{q}_i)$ for $j \in [m]$.
2. Output $\mathsf{ans}_i = (\zeta_i^{(0)}, (\zeta_{ij}^{(1)})_{j\in[m]})$.

$\mathcal{R}(\widetilde{\mathsf{ans}}_1, \ldots, \widetilde{\mathsf{ans}}_\ell; \mathsf{aux})$**.** Given ℓ answers $\widetilde{\mathsf{ans}}_i = (\widetilde{\zeta}_i^{(0)}, (\widetilde{\zeta}_{ij}^{(1)})_{j\in[m]}) \in \mathbb{F}_p^{m+1}$ and auxiliary information $\mathsf{aux} = ((\alpha_i)_{i\in[\ell]}, (\boldsymbol{w}_i)_{i\in[\ell]})$:

1. Let $\widetilde{\boldsymbol{\zeta}}_i^{(1)} = (\widetilde{\zeta}_{i1}^{(1)}, \ldots, \widetilde{\zeta}_{im}^{(1)})$ for all $i \in [\ell]$.
2. Let $\widetilde{\xi}_i^{(0)} = \widetilde{\zeta}_i^{(0)}$ and $\widetilde{\xi}_i^{(1)} = \langle \widetilde{\boldsymbol{\zeta}}_i^{(1)}, \boldsymbol{w}_i \rangle$ for all $i \in [\ell]$.
3. Choose any subset $A \subseteq [\ell]$ of size at least $(td+1)/2$.
4. Compute a polynomial $\widetilde{g}(x) \in \mathbb{F}_p[x]$ of degree at most td such that $\widetilde{\xi}_i^{(0)} = \widetilde{g}(\alpha_i)$ and $\widetilde{\xi}_i^{(1)} = \partial\widetilde{g}(\alpha_i)$ for all $i \in A$, using Hermite interpolation.
5. Output $\widetilde{g}(0)$ if $\widetilde{\xi}_i^{(0)} = \widetilde{g}(\alpha_i)$ and $\widetilde{\xi}_i^{(1)} = \partial\widetilde{g}(\alpha_i)$ for all $i \notin A$ and $\widetilde{g}(0) \in \{0,1\}$. Otherwise, output $\perp$.

Figure 3 A t-private $(1-\epsilon)$-fully error detecting ℓ-server PIR scheme.

References

1 A. Ambainis. Upper bound on the communication complexity of private information retrieval. In *Automata, Languages and Programming*, pages 401–407, 1997.

2 K. Banawan and S. Ulukus. The capacity of private information retrieval from byzantine and colluding databases. *IEEE Transactions on Information Theory*, 65(2):1206–1219, 2019.

3 A. Beimel and Y. Ishai. Information-theoretic private information retrieval: A unified construction. In *Automata, Languages and Programming*, pages 912–926, 2001.

4 A. Beimel, Y. Ishai, E. Kushilevitz, and J.-F. Raymond. Breaking the o(n/sup 1/(2k-1)/) barrier for information-theoretic private information retrieval. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pages 261–270, 2002.

5 A. Beimel and Y. Stahl. Robust information-theoretic private information retrieval. *Journal of Cryptology*, 20(3):295–321, 2007.

6 Y.M. Chee, T. Feng, S. Ling, H. Wang, and L.F. Zhang. Query-efficient locally decodable codes of subexponential length. *computational complexity*, 22(1):159–189, 2013.

7 B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. *Journal of the ACM*, 45(6):965–982, 1998.

8 C. Devet, I. Goldberg, and N. Heninger. Optimally robust private information retrieval. In *21st USENIX Security Symposium (USENIX Security 12)*, pages 269–283, 2012.

9 Z. Dvir, P. Gopalan, and S. Yekhanin. Matching vector codes. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 705–714, 2010.

10 Z. Dvir and S. Gopi. 2-server pir with subpolynomial communication. *Journal of the ACM*, 63(4):1–15, 2016.

11 K. Efremenko. 3-query locally decodable codes of subexponential length. *SIAM Journal on Computing*, 41(6):1694–1703, 2012.

12 I. Goldberg. Improving the robustness of private information retrieval. In *2007 IEEE Symposium on Security and Privacy (SP'07)*, pages 131–148, 2007.

13 Vince Grolmusz. Superpolynomial size set-systems with restricted intersections mod 6 and explicit ramsey graphs. *Combinatorica*, 20(1):71–86, 2000.

14 T. Itoh and Y. Suzuki. Improved constructions for query-efficient locally decodable codes of subexponential length. *IEICE Transactions on Information and Systems*, E93.D(2):263–270, 2010.

15 K. Kurosawa. How to correct errors in multi-server pir. In *Advances in Cryptology – ASIACRYPT 2019*, pages 564–574, 2019.

16 UV Linnik. On the least prime in an arithmetic progression. II. The Deuring–Heilbronn phenomenon. *Rec. Math. [Mat. Sbornik] N.S.*, 15(3):347–368, 1944.

17 A. Spitzbart. A generalization of Hermite's interpolation formula. *The American Mathematical Monthly*, 67(1):42–46, 1960.

18 H. Sun and S.A. Jafar. The capacity of private information retrieval. *IEEE Transactions on Information Theory*, 63(7):4075–4088, 2017.

19 H. Sun and S.A. Jafar. The capacity of robust private information retrieval with colluding databases. *IEEE Transactions on Information Theory*, 64(4):2361–2370, 2018.

20 D. Woodruff and S. Yekhanin. A geometric approach to information-theoretic private information retrieval. In *20th Annual IEEE Conference on Computational Complexity (CCC'05)*, pages 275–284, 2005.

21 E.Y. Yang, Jie Xu, and K.H. Bennett. Private information retrieval in the presence of malicious failures. In *Proceedings 26th Annual International Computer Software and Applications*, pages 805–810, 2002.

22 S. Yekhanin. Towards 3-query locally decodable codes of subexponential length. *Journal of the ACM (JACM)*, 55(1):1–16, 2008.

A Proof of Theorem 15

We first show a lemma used in the proof of Theorem 15.

▶ **Lemma 22.** *Let $T_1, \dots, T_m$ be m pairwise disjoint subsets of $[\ell]$ such that $T_1 \cup \dots \cup T_m = [\ell] \setminus \{1\}$ and $|T_i| \le t$ for all $i \in [m]$. Consider the following conditions on a family $\mathcal{F} = \{S_1, \dots, S_N\}$ of subsets of $[\ell]$:*

1. *For any $i \in [\ell]$, $|S_i| = k$ and $1 \in S_i$;*
2. *For any $h \in [m]$, there exists at most one $i \in [N]$ such that $T_h \cap S_i \neq \emptyset$.*

Define M be the maximum size of $\mathcal{F}$ satisfying the above conditions. Then,

$$M \ge M' = \left\lceil \frac{\ell - k + 1}{k + t - 2} \right\rceil .$$

Proof of Lemma 22. It is sufficient to construct a family $\mathcal{F}$ with $|\mathcal{F}| \ge M'$ satisfying the conditions 1 and 2. We consider the following algorithm to generate a sequence $u_1, u_2, \dots$ in $[m]$:

1. Set $i = 1$ and $u_0 = 0$;
2. Repeat the following:
   a. If $\sum_{u_{i-1}+1 \le h \le m} |T_h| \ge k - 1$, let u_i be the smallest element such that $|T_{u_{i-1}+1}| + |T_{u_{i-1}+2}| + \dots + |T_{u_i}| \ge k - 1$. Otherwise, go to Step 3.
   b. Set $i \leftarrow i + 1$.
3. Output $u_1, \dots, u_{i-1}$.

Let $u_0 = 0, u_1, \dots, u_N$ be an output of the above algorithm. For each $i \in [N]$, choose any subset S'_i of size $k - 1$ such that $S'_i \subseteq T_{u_{i-1}+1} \cup \dots \cup T_{u_i}$. We define $\mathcal{F} = \{S'_i \cup \{1\} : i \in [N]\}$. It easily follows from the definition that $\mathcal{F}$ satisfies the conditions 1 and 2. Since $\sum_{u_{i-1}+1 \le h \le u_i - 1} |T_h| \le k - 2$ and $|T_{u_i}| \le t$, it holds that $\sum_{u_{i-1}+1 \le h \le u_i} |T_h| \le k + t - 2$ for any $i \in [N]$. By adding them up, we have that

$$\ell - 1 - \sum_{u_N+1 \le h \le m} |T_h| = \sum_{1 \le h \le u_N} |T_h| \le N(k + t - 2)$$

and hence

$$|\mathcal{F}| = N \ge \frac{\ell - k + 1}{k + t - 2}$$

since $\sum_{u_N+1 \le h \le m} |T_h| \le k - 2$. ◀

We consider a PIR scheme $\Pi' = (\mathcal{Q}', \mathcal{A}', \mathcal{R}')$ where $(\mathcal{Q}', \mathcal{A}')$ runs $\binom{\ell}{k}$ independent instances of $(\mathcal{Q}, \mathcal{A})$ between a client and every subset of k servers. The communication complexity of Π' is $c \cdot \binom{\ell}{k}$. Since each execution of $\mathcal{Q}(\tau)$ is done independently, Π' is also t-private. Furthermore for each execution of $(\mathcal{Q}, \mathcal{A})$, $\mathcal{R}'$ runs $\mathcal{R}$ on the corresponding input. If $\mathcal{R}$ outputs the same value a for every execution, then $\mathcal{R}'$ outputs a. Otherwise $\mathcal{R}'$ outputs $\bot$. Then it is clear that Π' is correct. It also follows that it is (k, ℓ)-robust and hence $\lfloor (\ell - k)/2 \rfloor$-error correcting.

We prove that Π' is $(1 - \epsilon')$-fully error detecting. Without loss of generality, suppose that the first server is honest and all the other servers are malicious. Consider pairwise disjoint subsets $T_1, \dots, T_m$ such that $T_1 \cup \ldots \cup T_m = [\ell] \setminus \{1\}$ and $|T_h| \le t$ for $h \in [m]$. We assume that all the servers in each T_h can collude.

Let $p = \binom{\ell}{k}$ and let $S_1, \dots, S_p$ be all k-sized subsets of $[\ell]$. Let $\mathcal{F}$ be a family of subsets of $[\ell]$ with $|\mathcal{F}| = M \ge M'$ given by Lemma 22. By rearranging the order, we may assume that $\mathcal{F} = \{S_1, \dots, S_M\}$. Let Π_j denote the instance of Π executed by the client and servers in S_j.

During the execution of Π_j, the client generates $\mathcal{Q}(\tau; r_j) \to (\{\mathsf{que}_i^{(j)} : i \in S_j\}; \mathsf{aux}^{(j)})$, where r_j is a random string and $\mathsf{que}_i^{(j)}$ is sent to the i-th server for $i \in S_j$. The i-th server then receives $\mathsf{que}'_i = \{\mathsf{que}_i^{(j)} : j \in [p]$ with $i \in S_j\}$. In each Π_j, the i-th server with $i \in S_j$ returns

$$\widetilde{\mathsf{ans}}_i^{(j)} = \begin{cases} \mathsf{ans}_1^{(j)} = \mathcal{A}(1, \mathsf{que}_1^{(j)}, \boldsymbol{a}), & \text{if } i = 1, \\ f_i^{(j)}(\{\mathsf{que}'_{i'}\}_{i' \in T_h}, \boldsymbol{a}), & \text{if } i \in T_h \end{cases}$$

for some function $f_i^{(j)}$, where $\boldsymbol{a} = (a_1, \dots, a_n)$ is a database. It then follows from our definition of $\mathcal{R}'$ that

$$\begin{aligned} & \Pr[\mathcal{R}' \text{ outputs some } a' \notin \{a_\tau, \bot\}] \\ \leq\ & \Pr\left[\forall j \in [M] : \mathcal{R}(\mathsf{ans}_1^{(j)}, \{\widetilde{\mathsf{ans}}_i^{(j)}\}_{i \in S_j \setminus \{1\}}; \mathsf{aux}^{(j)}) \notin \{a_\tau, \bot\}\right]. \end{aligned}$$

Since $\epsilon^M \leq \epsilon'$, it is enough to show that

$$p_0 := \Pr\left[\forall j \in [M] : \mathcal{R}(\mathsf{ans}_1^{(j)}, \{\widetilde{\mathsf{ans}}_i^{(j)}\}_{i \in S_j \setminus \{1\}}; \mathsf{aux}^{(j)}) \notin \{a_\tau, \bot\}\right] \leq \epsilon^M.$$

Now fix $r' = (r_{M+1}, \dots, r_p)$ arbitrarily. Then $\mathsf{que}_i^{(j)}$ is a fixed constant for any $j \in \{M+1, \dots, p\}$ and $i \in S_j$. Let $j \in [M]$, $i \in S_j \setminus \{1\}$ and let $h \in [m]$ be the unique index such that $i \in T_h$. Since $\mathcal{F}$ satisfies the condition 2 in Lemma 22, we have that $T_h \cap S_{j'} = \emptyset$ for any $j' \in [M] \setminus \{j\}$. Therefore, we can write

$$\widetilde{\mathsf{ans}}_i^{(j)} = f_i^{(j)}(\{\mathsf{que}'_{i'}\}_{i' \in T_h}, \boldsymbol{a}) = g_{i,r'}(\{\mathsf{que}_{i'}^{(j)}\}_{i' \in T_h}, \boldsymbol{a})$$

using some function $g_{i,r'}$. In particular, $\{\widetilde{\mathsf{ans}}_i^{(j)}\}_{i \in S_j \setminus \{1\}}$ and $\{\widetilde{\mathsf{ans}}_i^{(j')}\}_{i \in S_{j'} \setminus \{1\}}$ are independent if $j \neq j' \in [M]$. Let $\mathcal{X}$ denote the random variable which represents $r' = (r_{M+1}, \dots, r_p)$. Then, for any fixed $r' = (r_{M+1}, \dots, r_p)$, we have that

$$\begin{aligned} & \Pr_{r_1, \dots, r_M}\left[\forall j \in [M] : \mathcal{R}(\mathsf{ans}_1^{(j)}, \{\widetilde{\mathsf{ans}}_i^{(j)}\}_{i \in S_j \setminus \{1\}}; \mathsf{aux}^{(j)}) \notin \{a_\tau, \bot\} \mid \mathcal{X} = r'\right] \\ \leq\ & \prod_{j \in [M]} \Pr_{r_j}\left[\mathcal{R}(\mathsf{ans}_1^{(j)}, \{\widetilde{\mathsf{ans}}_i^{(j)}\}_{i \in S_j \setminus \{1\}}; \mathsf{aux}^{(j)}) \notin \{a_\tau, \bot\} \mid \mathcal{X} = r'\right] \\ \leq\ & \epsilon^M \end{aligned}$$

since Π is $(1-\epsilon)$-fully error detecting. Therefore, it holds that

$$\begin{aligned} p_0 &= \sum_{r'} \Pr\left[\mathcal{X} = r'\right] \Pr_{r_1, \dots, r_M}\left[\forall j \in [M] : \mathcal{R}(\mathsf{ans}_1^{(j)}, \{\widetilde{\mathsf{ans}}_i^{(j)}\}_{i \in S_j}; \mathsf{aux}^{(j)}) \notin \{a_\tau, \bot\} \;\middle|\; \mathcal{X} = r'\right] \\ &\leq \sum_{r'} \Pr\left[\mathcal{X} = r'\right] \times \epsilon^M \\ &\leq \epsilon^M. \end{aligned}$$

B Proof of Theorem 16

Using the notations in Figure 1, the 1-privacy of Π_1 follows from the fact that a random vector $\boldsymbol{w}^{(j)}$ masks $\boldsymbol{u}_\tau$ for each $j \in [\lambda]$.

First we show the correctness of Π_1. Fix a database $\boldsymbol{a} = (a_1, \dots, a_n) \in \{0,1\}^n$ and a client's index τ. Note that $\xi_i^{(j)}$ at Step 2 of $\mathcal{A}$ in Figure 1 is computed as

$$\xi_i^{(j)} = \sum_{\sigma \in [n]} a_\sigma \gamma^{\langle \boldsymbol{w}^{(j)}, \boldsymbol{v}_\sigma \rangle + \alpha_i^{(j)} \langle \boldsymbol{u}_\tau, \boldsymbol{v}_\sigma \rangle} = c_0^{(j)} + \sum_{s \in \{p, q, p+q\}} c_s^{(j)} (\gamma^{\alpha_i^{(j)}})^s$$

for each $j \in [\lambda]$, where $c_0^{(j)} = a_\tau \gamma^{\langle \boldsymbol{w}^{(j)}, \boldsymbol{v}_\tau \rangle}$ and $c_s^{(j)} = \sum_{\sigma \in [n] : \langle \boldsymbol{u}_\tau, \boldsymbol{v}_\sigma \rangle = s} a_\sigma \gamma^{\langle \boldsymbol{w}^{(j)}, \boldsymbol{v}_\sigma \rangle}$. Similarly, $\boldsymbol{\zeta}_i^{(j)}$ at Step 2 of $\mathcal{A}$ is written as

$$\boldsymbol{\zeta}_i^{(j)} = \sum_{\sigma \in [n]} a_\sigma \phi(\boldsymbol{v}_\sigma) \gamma^{\langle \boldsymbol{w}^{(j)}, \boldsymbol{v}_\sigma \rangle + \alpha_i^{(j)} \langle \boldsymbol{u}_\tau, \boldsymbol{v}_\sigma \rangle}.$$

Hence, $\eta_i^{(j)}$ at Step 2(a) of $\mathcal{R}$ is computed as follows:

$$\begin{aligned}
\eta_i^{(j)} &:= \langle \phi(\boldsymbol{u}_\tau), \boldsymbol{\zeta}_i^{(j)} \rangle \\
&= \sum_{\sigma \in [n]} a_\sigma \phi(\langle \boldsymbol{u}_\tau, \boldsymbol{v}_\sigma \rangle) \gamma^{\langle \boldsymbol{w}^{(j)}, \boldsymbol{v}_\sigma \rangle} (\gamma^{\alpha_i^{(j)}})^{\langle \boldsymbol{u}_\tau, \boldsymbol{v}_\sigma \rangle} \\
&= a_\tau \phi(0) \gamma^{\langle \boldsymbol{w}^{(j)}, \boldsymbol{v}_\tau \rangle} + \sum_{s \in \{p, q, p+q\}} c_s^{(j)} \phi(s) (\gamma^{\alpha_i^{(j)}})^s \\
&= \sum_{s \in \{p, q, p+q\}} c_s^{(j)} \phi(s) (\gamma^{\alpha_i^{(j)}})^s.
\end{aligned}$$

On the other hand, the matrix $M^{(j)}$ computed by $\mathcal{R}$ is written as

$$M^{(j)} = \begin{pmatrix} 1 & (\beta_1^{(j)})^p & (\beta_1^{(j)})^q & (\beta_1^{(j)})^{p+q} \\ 0 & \phi(p)(\beta_1^{(j)})^p & \phi(q)(\beta_1^{(j)})^q & \phi(p+q)(\beta_1^{(j)})^{p+q} \\ 1 & (\beta_2^{(j)})^p & (\beta_2^{(j)})^q & (\beta_2^{(j)})^{p+q} \\ 0 & \phi(p)(\beta_2^{(j)})^p & \phi(q)(\beta_2^{(j)})^q & \phi(p+q)(\beta_2^{(j)})^{p+q} \end{pmatrix}$$

since $(\beta_i^{(j)})^p = (\gamma^p)^{\alpha_i^{(j)}} = 1$, $(\beta_i^{(j)})^q = \beta_i^{(j)}$ in $\mathbb{F}_q$, and $\phi(p) = p, \phi(q) = 0, \phi(p+q) = p$ due to the definition of ϕ. We therefore have that

$$M^{(j)} \begin{pmatrix} c_0^{(j)} \\ c_p^{(j)} \\ c_q^{(j)} \\ c_{p+q}^{(j)} \end{pmatrix} = \begin{pmatrix} \xi_1^{(j)} \\ \eta_1^{(j)} \\ \xi_2^{(j)} \\ \eta_2^{(j)} \end{pmatrix}. \tag{4}$$

Furthermore, it holds that $\det M^{(j)} = p^2(\beta_1^{(j)} - \beta_2^{(j)})^2 \neq 0$ since $\alpha_1^{(j)} \neq \alpha_2^{(j)}$. Therefore $\mathcal{R}$ correctly recovers $c_0^{(j)}$, $c_p^{(j)}$, $c_q^{(j)}$, $c_{p+q}^{(j)}$ for each $j \in [\lambda]$. Hence $\mathcal{L} = \{a_\tau\}$ and $\mathcal{R}$ outputs a_τ. Thus the correctness of Π_1 holds.

We next show that Π_1 is $(1 - \epsilon)$-fully error detecting. The reconstruction algorithm $\mathcal{R}$ computes $\widetilde{c}_0, \widetilde{c}_p, \widetilde{c}_q, \widetilde{c}_{p+q}$ such that

$$M^{(j)} \begin{pmatrix} \widetilde{c}_0^{(j)} \\ \widetilde{c}_p^{(j)} \\ \widetilde{c}_q^{(j)} \\ \widetilde{c}_{p+q}^{(j)} \end{pmatrix} = \begin{pmatrix} \widetilde{\xi}_1^{(j)} \\ \widetilde{\eta}_1^{(j)} \\ \widetilde{\xi}_2^{(j)} \\ \widetilde{\eta}_2^{(j)} \end{pmatrix} \tag{5}$$

for all $j \in [\lambda]$. Without loss of generality, we may assume that $(\widetilde{\xi}_1^{(j)}, \widetilde{\boldsymbol{\eta}}_1^{(j)}) = (\xi_1^{(j)}, \boldsymbol{\eta}_1^{(j)})$ and $(\widetilde{\xi}_2^{(j)}, \widetilde{\boldsymbol{\eta}}_2^{(j)}) \neq (\xi_2^{(j)}, \boldsymbol{\eta}_2^{(j)})$. Namely the first server is honest and the second server is malicious. From Eqs. (4) and (5), it holds that

$$M^{(j)} \begin{pmatrix} \widetilde{c}_0^{(j)} - c_0^{(j)} \\ \widetilde{c}_p^{(j)} - c_p^{(j)} \\ \widetilde{c}_q^{(j)} - c_q^{(j)} \\ \widetilde{c}_{p+q}^{(j)} - c_{p+q}^{(j)} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \mu_2^{(j)} \\ \nu_2^{(j)} \end{pmatrix} \quad (\forall j \in [\lambda])$$

where $\mu_2^{(j)} = \widetilde{\xi}_2^{(j)} - \xi_2^{(j)}$ and $\nu_2^{(j)} = \widetilde{\eta}_2^{(j)} - \eta_2^{(j)}$. By multiplying the adjugate matrix $\mathrm{adj}(M^{(j)})$ from the left, we obtain the first entry as

$$(\det M^{(j)})(\widetilde{c}_0^{(j)} - c_0^{(j)}) = M_{13}^{(j)}\mu_2^{(j)} + M_{14}^{(j)}\nu_2^{(j)} \ (\forall j \in [\lambda]),$$

where $M_{k\ell}^{(j)}$ is the (k,ℓ)-entry of $\mathrm{adj}(M^{(j)})$. Let $\widetilde{c}_0^{(j)} = a'\gamma^{\langle \boldsymbol{w}^{(j)}, \boldsymbol{v}_\tau \rangle}$, and define $\Delta := a' - a_\tau$ and $\delta^{(j)} := \gamma^{\langle \boldsymbol{w}^{(j)}, \boldsymbol{v}_\tau \rangle}$. By calculating the adjacent matrix $\mathrm{adj}(M^{(j)})$ directly, we have

$$p^2(\beta_1^{(j)} - \beta_2^{(j)})^2 \Delta\delta^{(j)} = (p\mu_2^{(j)} - \nu_2^{(j)})p(\beta_1^{(j)} - \beta_2^{(j)})\beta_1^{(j)} \tag{6}$$

for any $j \in [\lambda]$. Let $x_1 = \beta_1^{(j)} = \gamma^{\alpha_1^{(j)}}$. Then we have

$$p^2(x_1 - \beta_2^{(j)})^2 \Delta\delta^{(j)} - (p\mu_2^{(j)} - \nu_2^{(j)})p(x_1 - \beta_2^{(j)})x_1 = 0$$

and hence

$$p(x_1 - \beta_2^{(j)})\left((p\Delta\delta^{(j)} - p\mu_2^{(j)} + \nu_2^{(j)})x_1 - p\Delta\delta^{(j)}\beta_2^{(j)}\right) = 0.$$

Since $x_1 = \beta_1^{(j)} \neq \beta_2^{(j)}$, it must hold that

$$(p\Delta\delta^{(j)} - p\mu_2^{(j)} + \nu_2^{(j)})x_1 - p\Delta\delta^{(j)}\beta_2^{(j)} = 0. \tag{7}$$

Now suppose that $\Delta = a' - a_\tau \neq 0$. Then it must hold that $p\Delta\delta^{(j)} - p\mu_2^{(j)} + \nu_2^{(j)} \neq 0$ since $p\Delta\delta^{(j)}\beta_2^{(j)} \neq 0$. Furthermore $x_1 \neq \beta_2^{(j)}$ is randomly chosen independently from the other values in Eq. (7). Therefore Eq. (7) holds for all $j \in [\lambda]$ with probability at most $(p-1)^{-\lambda}$. This means that $\mathcal{R}$ adds $a' \neq a_\tau$ to $\mathcal{L}$ at Step 2(d) with probability at most $\epsilon = (p-1)^{-\lambda}$ since Δ takes at most one value. Therefore our PIR scheme is $(1-\epsilon)$-fully error detecting.

Finally, a prime q satisfying $q \equiv 1 \bmod p$ can be chosen as $q = p^{O(1)}$ from Linnik's theorem [16]. It then holds that $\lambda \log q = O(\lambda \log p) = O(\log \epsilon^{-1})$ and the communication complexity is given by $O(h\lambda \log q) = \mathcal{L}_n[1/2, \theta_m] \cdot \log \epsilon^{-1}$ from Proposition 7, where θ_m is a constant depending on $m = pq$ only. Since $m = pq$ can be chosen as a constant, we conclude that the communication complexity is $\mathcal{L}_n[1/2, O(1)] \cdot \log \epsilon^{-1}$.