PAPER *Special Section on Cryptography and Information Security*

# Efficient Private Information Retrieval

**Toshiya ITOH**†, ***Member***

**SUMMARY** Informally, private information retrieval for $k \geqq 1$ databases ($k$-PIR) is an interactive scheme that enables a user to make access to (separated) $k$ replicated copies of a database and privately retrieve any single bit out of the $n$ bits of data stored in the database. In this model, "privacy" implies that the user retrieves the bit he is interested in but releases to each database nothing about which bit he really tries to get. Chor et. al. proposed 2-PIR with communication complexity $12n^{1/3} + 2$ that is based on the covering codes. Then Ambainis recursively extended the scheme by Chor et. al. and showed that for each $k \geqq 2$, there exists $k$-PIR with communication complexity at most $c_k \cdot n^{1/(2k-1)}$ some constant $c_k > 0$. In this paper, we relax the condition for the covering codes and present time-efficient 2-PIR with communication complexity $12n^{1/3}$. In addition, we generally formulate the recursive scheme by Ambainis and show that for each $k \geqq 4$, there exists $k$-PIR with communication complexity at most $c'_k \cdot n^{1/(2k-1)}$ for some constant $c'_k \ll c_k$.

***key words:*** *information retrieval, privacy, communication complexity, time complexity, covering codes*

## 1. Introduction

### 1.1 Background

Private information retrieval for $k \geqq 1$ databases ($k$-PIR) is initiated by Chor et. al. [4] as a useful way for a user to privately get information through networks. It would be quite natural to ask for the privacy of the user. For example, an investor (or a speculator) that makes access to the stock-market database to get the value of a certain stock may wish to keep private which stock he is interested in. Informally, $k$-PIR is a scheme that enables a user to make access to (separated) $k$ replicated copies of a database and privately retrieve a single bit of data stored in the database. In this framework, "private" implies that the user is able to retrieve his desired bit but releases to each database nothing about which bit he tries to get in the information-theoretic sense. In the practical point of view, the communication complexity between the user and the databases (and the time complexity of the user and the databases) seems to be one of the most important resources for constructing efficient and practical $k$-PIR.

When the user makes access to only a single database, he may ask for a copy of the whole database to privately retrieve the bit that he is interested in. Obviously, this requires $O(n)$ communication complexity but is proved to be essentially the best he can do. Indeed, Chor et al. [4] showed that any 1-PIR requires $\Omega(n)$ communication complexity. To reduce communication complexity of $k$-PIR, Chor et. al. [4] applied the covering codes [10] and proposed 2-PIR with communication complexity $12n^{1/3} + 2$ and $k$-PIR with communication complexity $O(n^{1/k})$, where $n$ is the length of total data stored in the database. It is conjectured by Chor et. al. [4] that $O(n^{1/3})$ might be the lower bound for the communication complexity of any 2-PIR. Then Ambainis [1] recursively extended the scheme by Chor et. al. [4] to construct $k$-PIR with less communication complexity and showed that for each $k \geqq 2$, there exists $k$-PIR with communication complexity $O(n^{1/(2k-1)})$.

By applying cryptographic (secure) primitives, Chor and Gilboa [3] extended $k$-PIR in a natural way to define *computationally* private information retrieval for $k \geqq 1$ databases ($k$-CPIR). This is a scheme that is similar to the original $k$-PIR but the user releases (in the computational sense) to each database nothing about which bit he tries to retrieve. In this framework, Chor and Gilboa [3] showed that for any $\varepsilon > 0$, there exists 2-CPIR with communication complexity $O(n^{\varepsilon})$ under the general assumption that pseudo-random generators exist [6], [7]. Intuitively, 1-CPIR with communication complexity $o(n)$ would be impossible, however, Kushilevitz and Ostrovsky [8] recently showed that for any $\varepsilon > 0$, there exists 1-CPIR with communication complexity $O(n^{\varepsilon})$ under the (stronger but reasonable) assumption that quadratic residuosity [5] is hard.

### 1.2 Motivation

As for the communication complexity, the 2-CPIR [3] and the 1-CPIR [8] achieve much better than $k$-PIR for any reasonable $k \geqq 2$, however, there exist trade-offs between the communication complexity and the time complexity. To achieve $O(n^{\varepsilon})$ communication complexity for any $\varepsilon > 0$, the 2-CPIR [3] and the 1-CPIR [8] are recursively constructed and in each recursive step, each database (in the 2-CPIR [3] and the 1-CPIR [8]) needs to execute large amount of computations. This may eventually lead to a large delay until the user receives the response from the database. In addition to this, the privacy of the user in $k$-CPIR is asymptotically guaran-

Manuscript received March 18, 1998.
Manuscript revised July 21, 1998.

†The author is with Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology, Yokohama-shi, 226–8502 Japan.

teed with respect to $n$ (the length of total data stored in the database), i.e., the privacy of the user in $k$-CPIR is protected for sufficiently large $n$. In general, this implies that the privacy of the user in $k$-CPIR are not necessarily guaranteed for reasonable size of data stored in the database.

Thus $k$-PIR is more advantageous than $k$-CPIR from the practical-oriented privacy point of view, because $k$-PIR protects the privacy of the user for any size of data stored in the database. The goal of this paper is to design time-efficient (or communication-efficient) $k$-PIR.

### 1.3 Main Results

In this paper, we relax the condition for the covering codes to design time-efficient 2-PIR and present 2-PIR $\mathsf{PIR}_{\mathsf{BD}}$ with communication complexity $12n^{1/3}$. In addition, we generally formulate the recursive $k$-PIR by Ambainis [1] and show that for each $k \geqq 4$, there exist $k$-PIR $\mathsf{PIR}^k_{\mathsf{BREC}}$ of which communication complexity is much less than that of $k$-PIR by Ambainis [1].

## 2. Preliminaries

In this paper, $x = x_1x_2\ldots x_n \in \{0,1\}^n$ denotes the contents of the database and let $n = |x|$. For $m > 0$, let $[m] = \{1, 2, \ldots, m\}$ and $d(a, b)$ be the *Hamming distance* of $a$ and $b$.

### 2.1 The Model

A model for $k$-PIR is similar to that of multi-prover interactive proofs [2]. We use $\mathcal{U}$ to denote a user that is a probabilistic polynomial time interactive Turing machine and $\mathcal{DB}_1, \mathcal{DB}_2, \ldots, \mathcal{DB}_k$ denote deterministic polynomial time interactive Turing machines that are allowed to communicate with $\mathcal{U}$ but are not allowed to communicate with each other. We also use $q_j^\ell$ to denote the $\ell$th question made by $\mathcal{U}$ to $\mathcal{DB}_j$ and $a_j^\ell$ to denote the $\ell$th answer returned by $\mathcal{DB}_j$ to $\mathcal{U}$. For each $j \in [k]$, let $Q_j^\ell = (q_j^1, q_j^2, \ldots, q_j^\ell)$ and $A_j^\ell = (a_j^1, a_j^2, \ldots, a_j^\ell)$.

**Definition 2.1** [4]**:** We say that $(\mathcal{U}; \mathcal{DB}_1, \mathcal{DB}_2, \ldots, \mathcal{DB}_k)$ is $m$-round information retrieval for $k \geqq 1$ databases ($k$-IR) if for any $n > 0$, any $x \in \{0,1\}^n$, and any $i \in [n]$, it satisfies the following: Let $x_i \in \{0,1\}$ be the bit that $\mathcal{U}$ wishes to retrieve from $x$. For each round $\ell \in [m]$ and each $j \in [k]$, $\mathcal{U}$ sends $q_j^\ell(i) = \mathcal{U}(i, j, n; A_1^{\ell-1}, A_2^{\ell-1}, \ldots, A_k^{\ell-1})$ to $\mathcal{DB}_j$ as the $\ell$th question and $\mathcal{DB}_j$ responds $\mathcal{U}$ with $a_j^\ell = \mathcal{DB}_j(x, Q_j^\ell(i))$ as the $\ell$th answer. At the end of round $m$, $\mathcal{U}$ can always retrieve the bit $x_i$ from $A_1^m, A_2^m, \ldots, A_k^m$, i.e., $x_i = \mathcal{U}(i, n; A_1^m, A_2^m, \ldots, A_k^m)$.

Since $\mathcal{U}$ is a probabilistic polynomial time machine, $Q_j^m(i)$ is a random variable. Note that $i \in [n]$ is not a random variable, because it is the bit position that $\mathcal{U}$ wishes to retrieve.

**Definition 2.2** (Privacy [4])**:** We say that $(\mathcal{U}; \mathcal{DB}_1, \mathcal{DB}_2, \ldots, \mathcal{DB}_k)$ is $m$-round private information retrieval for $k \geqq 1$ databases ($k$-PIR) if it is $k$-IR and satisfies the following: For each $j \in [k]$, $Q_j^m(i_1)$ and $Q_j^m(i_2)$ are identically distributed for every $i_1, i_2 \in [n]$.

**Definition 2.3** (Communication Complexity [4])**:** Let $\Pi = (\mathcal{U}; \mathcal{DB}_1, \mathcal{DB}_2, \ldots, \mathcal{DB}_k)$ be $k$-PIR. Then communication complexity of $\Pi$ is the sum of the total amount of bits exchanged between $\mathcal{U}$ and each $\mathcal{DB}_j$ $(1 \leqq j \leqq k)$, i,e., $\sum_{j\in[k]}\{|Q_j^m(i)| + |A_j^m|\}$.

**Definition 2.4** (Time Complexity)**:** Let $\Pi = (\mathcal{U}; \mathcal{DB}_1, \mathcal{DB}_2, \ldots, \mathcal{DB}_k)$ be $k$-PIR. Then time complexity of $\Pi$ is the sum of the running time of $\mathcal{U}$ and the maximum running time of $\mathcal{DB}_j$.

Obviously, communication complexity corresponds to the network cost and time complexity to the delay to retrieve information. For $k$-PIR $\Pi$, we use $\mathrm{COM}_\Pi(n)$ to denote communication complexity of $\mathsf{PIR}_\Pi$, and $\mathrm{TIME}_\Pi(n)$ to denote time complexity of $\mathsf{PIR}_\Pi$.

### 2.2 Useful Properties

For any $S \subseteqq [m]$, $\vec{S} = (s_1, s_2, \ldots, s_m) \in \{0,1\}^m$ is defined to be $s_i = 1$ if $i \in S$ and $s_i = 0$ if $i \notin S$. For any $S \subseteqq [m]$ and $i \in [m]$, define $S \oplus i$ to be $S \setminus \{i\}$ if $i \in S$ and $S \cup \{i\}$ if $i \notin S$, and for $a = (a_1, a_2, \ldots, a_m) \in \{0,1\}^m$ and $b = (b_1, b_2, \ldots, b_m) \in \{0,1\}^m$, let $(a, b)$ denote the inner product of $a$ and $b$ modulo 2, i.e., $(a, b) = \sum_{i\in[m]} a_ib_i \pmod 2 = \bigoplus_{i\in[m]} a_ib_i$.

**Proposition 2.5** [4, §3.1]**:** For any $S \subseteqq [n]$ and $i \in [n]$, let $S_1 = S$ and $S_2 = S \oplus i$ and also let $A_1 = (x, \vec{S_1}) = \bigoplus_{i\in S_1} x_i$ and $A_2 = (x, \vec{S_2}) = \bigoplus_{i\in S_2} x_i$. Then $A_1 \oplus A_2 = x_i$.

For each $d \geqq 2$, we assume without loss of generality that $n = \ell^d$, i.e., $\ell = n^{1/d}$. If it is not the case, we expand $x$ by padding appropriate number of zeros to satisfy the condition. We embed $x$ in a $d$-dimensional cube by associating each position $i \in [n]$ with a $d$-tuple $(i_1, i_2, \ldots, i_d)$ in a natural manner. The following is essential to design communication-efficient $k$-PIR.

**Proposition 2.6** [4, §3.2]**:** For each $t \in [n]$, let $(t_1, t_2, \ldots, t_d)$ be the associated $d$-tuple of $t \in [n]$. For each $j \in [d]$, we also let $S_j^1 \subseteqq [\ell]$ and $S_j^2 = S_j^1 \oplus t_j$, and for each $\sigma = \sigma_1\sigma_2\cdots\sigma_d \in \{1,2\}^d$, we define $A_{\sigma_1\sigma_2\cdots\sigma_d} = \bigoplus_{i_1\in S_1^{\sigma_1}} \bigoplus_{i_2\in S_2^{\sigma_2}} \cdots \bigoplus_{i_d\in S_d^{\sigma_d}} x_{i_1,i_2,\ldots,i_d}$. Then $x_{t_1,t_2,\ldots,t_d} = \bigoplus_{\sigma_1\sigma_2\cdots\sigma_d\in\{0,1\}^d} A_{\sigma_1\sigma_2\cdots\sigma_d}$.

For any integer $s \geqq 1$, let $m = \lceil n/s \rceil$ and $X = (X_1, X_2, \ldots, X_m) = x \in \{0,1\}^n$, where $X_i \in \{0,1\}^s$ for

each block $i \in [m]$, and we also assume without loss of generality that $m = \ell^d$, i.e., $\ell = m^{1/d}$. In a way similar to the above, we embed $X$ in a $d$-dimensional cube by associating each block $i \in [m]$ with a $d$-tuple $(i_1, i_2, \ldots, i_d)$ in a natural manner.

Note that Proposition 2.6 holds for the case that $s = 1$ but can be generalized to the case that $s > 1$ in an obvious way. The lemma below will play a central role in Sect. 3 to design time-efficient (and communication-efficient) $k$-PIR by an appropriate choice of $s > 1$.

**Lemma 2.7:** For each $t \in [m]$, let $(t_1, t_2, \ldots, t_d)$ be the associated $d$-tuple of $t$. For each $j \in [d]$, let $S_j^1 \subseteqq [\ell]$ and $S_j^2 = S_j^1 \oplus t_j$, and for each $\sigma_1\sigma_2\cdots\sigma_d \in \{1,2\}^d$, define $A_{\sigma_1\sigma_2\cdots\sigma_d} = \bigoplus_{i_1\in S_1^{\sigma_1}} \bigoplus_{i_2\in S_2^{\sigma_2}} \cdots \bigoplus_{i_d\in S_d^{\sigma_d}} X_{i_1,i_2,\ldots,i_d}$. Then $X_{t_1,t_2,\ldots,t_d} = \bigoplus_{\sigma_1\sigma_2\cdots\sigma_d\in\{0,1\}^d} A_{\sigma_1\sigma_2\cdots\sigma_d}$.

## 3. Time-Efficient 2-PIR

To analyze time complexity, we will make the following (practical) assumption: Let $r = 32$ or $r = 64$ and $\mathcal{U}$ and $\mathcal{DB}_i$ are equipped with an $r$-bit CPU. Since we are interested in information-theoretic (not complexity-theoretic) private information retrieval, $\mathcal{U}$ is allowed to make access to *truly* random bit sequence $\rho$ and requires a single step to read a random bit from $\rho$. Let $r \ll \ell$. For any $S \subseteqq [\ell]$, pointing index $i \in S$ requires $\lceil(\lg \ell)/r\rceil$ steps. Note that $\lceil(\lg \ell)/r\rceil = 1$ for $r = 32$ if $\ell \leqq 10^9$ and $\lceil(\lg \ell)/r\rceil$ for $r = 64$ if $\ell \leqq 10^{19}$.

### 3.1 The Covering Codes Scheme

To be self-contained, we show the 2-PIR ($\mathsf{PIR}_{\mathsf{CC}}$) by Chor et. al. [4] that is based on the covering codes [10]. Let $d = 3$ and $\ell = n^{1/3}$ and assume that $\mathcal{U}$ wishes to retrieve $x_t$ for some $t \in [n]$. Then $(t_1, t_2, t_3) \in [\ell]^3$ is the associated 3-tuple of $t$ in the 3-dimensional cube.

The Covering Codes Scheme: $\mathsf{PIR}_{\mathsf{CC}}$

U-1: $\mathcal{U}$ chooses $S_j^1 \subseteqq [\ell]$ uniformly and independently for each $1 \leqq j \leqq 3$.

U-2: $\mathcal{U}$ computes $S_j^2 = S_j^1 \oplus t_j$ for each $1 \leqq j \leqq 3$.

$\mathcal{U} \to \mathcal{DB}_1$: $S_1^1, S_2^1, S_3^1 \subseteqq [\ell]$.

$\mathcal{U} \to \mathcal{DB}_2$: $S_1^2, S_2^2, S_3^2 \subseteqq [\ell]$.

DB1-1: $\mathcal{DB}_1$ computes

$$b_{000} = \bigoplus_{i_1\in S_1^1} \bigoplus_{i_2\in S_2^1} \bigoplus_{i_3\in S_3^1} x_{i_1,i_2,i_3}.$$

DB1-2: $\mathcal{DB}_1$ computes

$$b_{100}^h = \bigoplus_{i_1\in S_1^1\oplus h} \bigoplus_{i_2\in S_2^1} \bigoplus_{i_3\in S_3^1} x_{i_1,i_2,i_3};$$

$$b_{010}^h = \bigoplus_{i_1\in S_1^1} \bigoplus_{i_2\in S_2^1\oplus h} \bigoplus_{i_3\in S_3^1} x_{i_1,i_2,i_3};$$

$$b_{001}^h = \bigoplus_{i_1\in S_1^1} \bigoplus_{i_2\in S_2^1} \bigoplus_{i_3\in S_3^1\oplus h} x_{i_1,i_2,i_3},$$

for each $h \in [\ell]$, and defines

$$B_{100} = (b_{100}^1, b_{100}^2, \ldots, b_{100}^\ell);$$
$$B_{010} = (b_{010}^1, b_{010}^2, \ldots, b_{010}^\ell);$$
$$B_{001} = (b_{001}^1, b_{001}^2, \ldots, b_{001}^\ell).$$

$\mathcal{DB}_1 \to \mathcal{U}$: $b_{000} \in \{0,1\}$, $B_{100}, B_{010}, B_{001} \in \{0,1\}^\ell$.

DB2-1: $\mathcal{DB}_2$ computes

$$c_{111} = \bigoplus_{i_1\in S_1^2} \bigoplus_{i_2\in S_2^2} \bigoplus_{i_3\in S_3^2} x_{i_1,i_2,i_3}.$$

DB2-2: $\mathcal{DB}_2$ computes

$$c_{011}^h = \bigoplus_{i_1\in S_1^2\oplus h} \bigoplus_{i_2\in S_2^2} \bigoplus_{i_3\in S_3^2} x_{i_1,i_2,i_3};$$

$$c_{101}^h = \bigoplus_{i_1\in S_1^2} \bigoplus_{i_2\in S_2^2\oplus h} \bigoplus_{i_3\in S_3^2} x_{i_1,i_2,i_3};$$

$$c_{110}^h = \bigoplus_{i_1\in S_1^2} \bigoplus_{i_2\in S_2^2} \bigoplus_{i_3\in S_3^2\oplus h} x_{i_1,i_2,i_3},$$

for each $h \in [\ell]$, and defines

$$C_{011} = (c_{011}^1, c_{011}^2, \ldots, c_{011}^\ell);$$
$$C_{101} = (c_{101}^1, c_{101}^2, \ldots, c_{101}^\ell);$$
$$C_{110} = (c_{110}^1, c_{110}^2, \ldots, c_{110}^\ell).$$

$\mathcal{DB}_2 \to \mathcal{U}$: $c_{111} \in \{0,1\}$, $C_{011}, C_{101}, C_{110} \in \{0,1\}^\ell$.

U-3: $\mathcal{U}$ computes $b_{000} \oplus b_{100}^{t_1} \oplus b_{010}^{t_2} \oplus b_{001}^{t_3} \oplus c_{111} \oplus c_{011}^{t_1} \oplus c_{101}^{t_2} \oplus c_{110}^{t_3}$ $(= x_t)$.

Obviously, $(S_1^1, S_2^1, S_3^1) \subseteqq [\ell]^3$ and $(S_1^2, S_2^2, S_3^2) \subseteqq [\ell]^3$ are uniformly distributed over $[\ell]^3$ for any $t \in [n]$. Thus it follows from Proposition 2.6 that $\mathsf{PIR}_{\mathsf{CC}}$ is 2-PIR.

**Lemma 3.1** [4]**:** $\mathrm{COM}_{\mathsf{CC}}(n) = 12n^{1/3} + 2$.

**Lemma 3.2:** $\mathrm{TIME}_{\mathsf{CC}}^{wc}(n) = 13n + 6n^{1/3} + 19$ and $\mathrm{TIME}_{\mathsf{CC}}^{av}(n) = (11/4)\cdot n + 6n^{1/3} + 19$.

**Proof:** In the practical point of view, we assume that $n \leqq 10^{20}$ and also that pointing $i \in S \subseteqq [\ell] = [n^{1/3}]$ requires a single step. From the description of $\mathsf{PIR}_{\mathsf{CC}}$, we immediately have the following: In both the worst and average cases, $3\ell$ steps for random generation in U-1, 3 steps for index pointing and 3 steps for XOR in U-2, and 6 steps for index pointing and 7 steps for XOR in U-3 are required. To compute $b_{000}$ in DB1-1, $\mathcal{DB}_1$ requires $3\ell^3$ steps for index pointing and $\ell^3$ steps for XOR in the worst case and $3(\ell/2)^3$ steps for index pointing and $(\ell/2)^3$ steps for XOR in the average case. To compute $B_{100}$ in DB1-2, $\mathcal{DB}_1$ requires $(2\ell^2 + 1)\ell$ steps for index pointing and $\ell^3$ steps for XOR in the

worst case and $\{2(\ell/2)^2+1\}\ell$ steps for index pointing and $(\ell/2)^2\ell$ steps for XOR in the average case by evaluating $b^h_{100} = b_{000} \oplus (\bigoplus_{i_2\in S^1_2} \bigoplus_{i_3\in S^1_3} x_{h,i_2,i_3})$ for each $h \in [\ell]$. The analysis similar to this can be applied to $B_{010}, B_{001}$. Since the number of steps made by $\mathcal{DB}_1$ is the same with that made by $\mathcal{DB}_2$ and $\ell = n^{1/3}$, we have that $\mathrm{TIME}^{wc}_{\mathsf{CC}}(n) = 13n + 6n^{1/3} + 19$ and $\mathrm{TIME}^{av}_{\mathsf{CC}}(n) = (11/4)\cdot n + 6n^{1/3} + 19$. □

### 3.2 The Block Division Scheme

In this subsection, we show new 2-PIR ($\mathsf{PIR_{BD}}$) that is based on Lemma 2.7. Let $m = \lceil n/s \rceil$ for some $s \geqq 1$ ($s$ will be fixed later), and $X = (X_1, X_2, \ldots, X_m) = x \in \{0,1\}^n$, where $X_i \in \{0,1\}^s$ for each block $i \in [m]$. Let $d = 2$ and $\ell = m^{1/2}$ and we embed $X$ in a 2-dimensional cube by associating each block $i \in [m]$ with a 2-tuple $(i_1, i_2)$ in a natural manner.

Here we assume that $\mathcal{U}$ wishes to retrieve $x_t$ for some $t \in [n]$ and that $x_t$ is located at the $\tau$th bit of $X_\beta$, where $\tau \in [s]$ and $\beta \in [m]$. Let $(\beta_1, \beta_2) \in [\ell]^2$ be the associated 2-tuple of $\beta$, and for any $A = (A_1, A_2, \ldots, A_\ell) \in \{0,1\}^\ell$ and $\alpha \in [\ell]$, let $\mathsf{pick}_\alpha(A) = A_\alpha$.

The Block Division Scheme: $\mathsf{PIR_{BD}}$

U-1: $\mathcal{U}$ chooses $S^1_1, S^1_2 \subseteqq [\ell]$ and $T^1 \subseteqq [s]$ uniformly and independently.

U-2: $\mathcal{U}$ computes $S^2_1 = S^1_1 \oplus \beta_1$, $S^2_2 = S^1_2 \oplus \beta_2$, and $T^2 = T^1 \oplus \tau$.

$\mathcal{U} \to \mathcal{DB}_1$: $S^1_1, S^1_2 \subseteqq [\ell]$, $T^1 \subseteqq [s]$.

$\mathcal{U} \to \mathcal{DB}_2$: $S^2_1, S^2_2 \subseteqq [\ell]$, $T^2 \subseteqq [s]$.

DB1-1: $\mathcal{DB}_1$ computes

$$B_{00} = \bigoplus_{i_1\in S^1_1} \bigoplus_{i_2\in S^1_2} X_{i_1,i_2}.$$

DB1-2: For each $h \in [\ell]$, $\mathcal{DB}_1$ computes

$$b^h_{10} = \bigoplus_{i_1\in S^1_1\oplus h} \bigoplus_{i_2\in S^1_2} (X_{i_1,i_2}, \vec{T}^1);$$
$$b^h_{01} = \bigoplus_{i_1\in S^1_1} \bigoplus_{i_2\in S^1_2\oplus h} (X_{i_1,i_2}, \vec{T}^1),$$

and defines $B_{10} = (b^1_{10}, b^2_{10}, \ldots, b^\ell_{10})$, $B_{01} = (b^1_{01}, b^2_{01}, \ldots, b^\ell_{01})$.

$\mathcal{DB}_1 \to \mathcal{U}$: $B_{00} \in \{0,1\}^s$, $B_{10}, B_{01} \in \{0,1\}^\ell$.

DB2-1: $\mathcal{DB}_2$ computes

$$C_{11} = \bigoplus_{i_1\in S^2_1} \bigoplus_{i_2\in S^2_2} X_{i_1,i_2}.$$

DB2-2: For each $h \in [\ell]$, $\mathcal{DB}_2$ computes

$$c^h_{01} = \bigoplus_{i_1\in S^2_1\oplus h} \bigoplus_{i_2\in S^2_2} (X_{i_1,i_2}, \vec{T}^2);$$
$$c^h_{10} = \bigoplus_{i_1\in S^2_1} \bigoplus_{i_2\in S^2_2\oplus h} (X_{i_1,i_2}, \vec{T}^2),$$

and defines $C_{01} = (c^1_{01}, c^2_{01}, \ldots, c^\ell_{01})$, $C_{10} = (c^1_{10}, c^2_{10}, \ldots, c^\ell_{10})$.

$\mathcal{DB}_2 \to \mathcal{U}$: $C_{11} \in \{0,1\}^s$, $C_{10}, C_{11} \in \{0,1\}^\ell$.

U-3: $\mathcal{U}$ computes $\mathsf{pick}_\tau(B_{00}) \oplus \mathsf{pick}_\tau(C_{11}) \oplus (b^{\beta_1}_{10} \oplus c^{\beta_2}_{10}) \oplus (b^{\beta_2}_{01} \oplus c^{\beta_1}_{01})$ $(= x_t)$.

The intuition behind $\mathsf{PIR_{BD}}$ is as follows: For each $h \in [\ell]$, let $Y^h_{10} = \bigoplus_{i_1\in S^1_1\oplus h} \bigoplus_{i_2\in S^1_2} X_{i_1,i_2}$ and $Y^h_{01} = \bigoplus_{i_1\in S^1_1} \bigoplus_{i_2\in S^1_2\oplus h} X_{i_1,i_2}$ instead of $b^h_{10}$ and $b^h_{01}$ respectively, and let $Z^h_{01} = \bigoplus_{i_1\in S^2_1\oplus h} \bigoplus_{i_2\in S^2_2} X_{i_1,i_2}$ and $Z^h_{10} = \bigoplus_{i_1\in S^2_1} \bigoplus_{i_2\in S^2_2\oplus h} X_{i_1,i_2}$ instead of $c^h_{01}$ and $c^h_{10}$ respectively. Recall that $\mathcal{U}$ wishes to retrieve $x_t$ for some $t \in [n]$ and $x_t$ is located at the $\tau$th bit of $X_{\beta_1,\beta_2}$. Then it follows from Lemma 2.7 that $X_{\beta_1,\beta_2} = B_{00} \oplus Y^{\beta_1}_{10} \oplus Y^{\beta_2}_{01} \oplus C_{11}$, but this blows up communication complexity. Since $\mathcal{U}$ is interested in the $\tau$th bit of $X_{\beta_1,\beta_2}$ and $Y^{\beta_1}_{10} = Z^{\beta_2}_{10}$ and $Y^{\beta_2}_{01} = Z^{\beta_1}_{01}$ hold from Proposition 2.5 and the definitions of $T^1$ and $T^2$, $(Y^{\beta_1}_{10}, \vec{T}^1) \oplus (Z^{\beta_2}_{10}, \vec{T}^2) = b^{\beta_1}_{10} \oplus c^{\beta_2}_{10}$ provides the $\tau$th bit of $Y^{\beta_1}_{10}$ and $(Y^{\beta_2}_{01}, \vec{T}^1) \oplus (Z^{\beta_1}_{01}, \vec{T}^2) = b^{\beta_2}_{01} \oplus c^{\beta_1}_{01}$ provides the $\tau$th bit of $Y^{\beta_2}_{01}$. These are sufficient for $\mathcal{U}$ to retrieve $x_t$ and reduce communication complexity.

**Theorem 3.3:** The block division scheme $\mathsf{PIR_{BD}}$ is private information retrieval for 2 databases.

**Proof:** For $d = 2$, it follows from Lemma 2.7 that $X_{\beta_1,\beta_2} = A_{00} \oplus A_{01} \oplus A_{10} \oplus A_{11}$, where

$$\begin{aligned}
A_{00} &= \bigoplus_{i_1\in S^1_1} \bigoplus_{i_2\in S^1_2} X_{i_1,i_2};\\
A_{01} &= \bigoplus_{i_1\in S^1_1} \bigoplus_{i_2\in S^1_2\oplus\beta_2} X_{i_1,i_2}\\
&= \bigoplus_{i_1\in S^1_1} \bigoplus_{i_2\in S^2_2} X_{i_1,i_2};\\
A_{10} &= \bigoplus_{i_1\in S^1_1\oplus\beta_1} \bigoplus_{i_2\in S^1_2} X_{i_1,i_2}\\
&= \bigoplus_{i_1\in S^2_1} \bigoplus_{i_2\in S^1_2} X_{i_1,i_2};\\
A_{11} &= \bigoplus_{i_1\in S^1_1\oplus\beta_1} \bigoplus_{i_2\in S^1_2\oplus\beta_2} X_{i_1,i_2}\\
&= \bigoplus_{i_1\in S^2_1} \bigoplus_{i_2\in S^2_2} X_{i_1,i_2},
\end{aligned}$$

for any $S^1_1 \subseteqq [\ell]$ and any $S^2_2 \subseteqq [\ell]$. Since $x_t$ is located at the $\tau$th bit of $X_{\beta_1,\beta_2}$, we have that $x_t = \mathsf{pick}_\tau(X_{\beta_1,\beta_2}) =$

$\mathsf{pick}_\tau(A_{00})\oplus\mathsf{pick}_\tau(A_{01})\oplus\mathsf{pick}_\tau(A_{10})\oplus\mathsf{pick}_\tau(A_{11})$. From the definitions of $A_{00}, A_{11}$ and $B_{00}, C_{11}$, it is immediate that $A_{00} = B_{00}$ and $A_{11} = C_{11}$, and thus we have that

$$\begin{aligned}&\mathsf{pick}_\tau(B_{00}) \oplus \mathsf{pick}_\tau(C_{11})\\ &\quad = \mathsf{pick}_\tau(A_{00}) \oplus \mathsf{pick}_\tau(A_{11}).\end{aligned} \tag{1}$$

On the other hand, from the definitions of $B_{10}, C_{10}$ and $A_{10}$, it is obvious that

$$\begin{aligned}b_{10}^{\beta_1} &= \bigoplus_{i_1\in S_1^1\oplus\beta_1}\bigoplus_{i_2\in S_2^1}(X_{i_1,i_2},\vec{T}^1)\\ &= \left(\bigoplus_{i_1\in S_1^2}\bigoplus_{i_2\in S_2^1}X_{i_1,i_2},\vec{T}^1\right) = \left(A_{10},\vec{T}^1\right);\\ c_{10}^{\beta_2} &= \bigoplus_{i_1\in S_1^2}\bigoplus_{i_2\in S_2^2\oplus\beta_2}(X_{i_1,i_2},\vec{T}^2)\\ &= \left(\bigoplus_{i_1\in S_1^2}\bigoplus_{i_2\in S_2^1}X_{i_1,i_2},\vec{T}^2\right) = \left(A_{10},\vec{T}^2\right).\end{aligned}$$

Then from the fact that $\vec{T}^2 = \vec{T}^1 \oplus \tau$ and Proposition 2.5, it follows that $b_{10}^{\beta_1} \oplus c_{10}^{\beta_2} = \mathsf{pick}_\tau(A_{10})$ and $b_{01}^{\beta_2} \oplus c_{01}^{\beta_1} = \mathsf{pick}_\tau(A_{01})$. Thus we have that

$$\begin{aligned}b_{10}^{\beta_1} \oplus c_{10}^{\beta_2} &= \mathsf{pick}_{\beta_1}(B_{10}) \oplus \mathsf{pick}_{\beta_2}(C_{10})\\ &= \mathsf{pick}_\tau(A_{10});\end{aligned} \tag{2}$$

$$\begin{aligned}b_{01}^{\beta_2} \oplus c_{01}^{\beta_1} &= \mathsf{pick}_{\beta_2}(B_{01}) \oplus \mathsf{pick}_{\beta_1}(C_{01})\\ &= \mathsf{pick}_\tau(A_{01}).\end{aligned} \tag{3}$$

Then from Eqs. (1), (2), and (3), it turns out that

$$\begin{aligned}x_t &= \mathsf{pick}_\tau(X_{\beta_1,\beta_2})\\ &= \mathsf{pick}_\tau(A_{00}) \oplus \mathsf{pick}_\tau(A_{01})\\ &\quad \oplus\mathsf{pick}_\tau(A_{10}) \oplus \mathsf{pick}_\tau(A_{11})\\ &= \mathsf{pick}_\tau(B_{00}) \oplus \mathsf{pick}_\tau(C_{11})\\ &\quad \oplus(b_{10}^{\beta_1} \oplus c_{10}^{\beta_2}) \oplus (b_{01}^{\beta_2} \oplus c_{01}^{\beta_1}).\end{aligned}$$

It is obvious that for any $t \in [n]$, $(S_1^1, S_2^1, T^1) \subsetneqq [\ell]^2 \times [s]$ and $(S_1^2, S_2^2, T^2) \subsetneqq [\ell]^2 \times [s]$ are uniformly distributed over $[\ell]^2 \times [s]$. Thus $\mathsf{PIR}_{\mathsf{BD}}$ is 2-PIR. □

**Theorem 3.4:** $\mathrm{COM}_{\mathsf{BD}}(n) = 12n^{1/3}$.

**Proof:** From the description of $\mathsf{PIR}_{\mathsf{BD}}$, it is immediate that $\mathrm{COM}_{\mathsf{BD}}(n) = 8\ell + 4s = 8\cdot(n/s)^{1/2} + 4s$. Then we have that $\mathrm{COM}_{\mathsf{BD}}(n) = 12n^{1/3}$ by setting $s = n^{1/3}$. □

**Theorem 3.5:** $\mathrm{TIME}^{wc}_{\mathsf{BD}}(n) = 5n + 4n^{2/3} + 9n^{1/3} + 17$ and $\mathrm{TIME}^{av}_{\mathsf{BD}}(n) = (5/4)\cdot n + (3/2)\cdot n^{2/3} + 8n^{1/3} + 17$.

**Proof:** In a way similar to the proof of Lemma 3.2, we assume that pointing $i \in S \subsetneqq [\ell] = [n^{1/3}]$ requires a single step. From the description of $\mathsf{PIR}_{\mathsf{BD}}$, we immediately have the following: In both the worst and average cases, $2\ell + s$ steps for random generation in U-1, 3 steps for index pointing and 3 steps for XOR in U-2, and 6 steps for index pointing and 5 steps for XOR in U-3 are required. To compute $B_{00}$ in DB1-1, $\mathcal{DB}_1$ requires $2\ell^2$ steps for index pointing and $\ell^2 s$ steps for XOR in the worst case and $2(\ell/2)^2$ steps for index pointing and $(\ell/2)^2 s$ steps for XOR in the average case. To compute $B_{10}$ in DB1-2, $\mathcal{DB}_1$ first evaluate $b = (B_{00}, \vec{T}^1)$ for which $\mathcal{DB}_1$ requires $s$ steps for AND and $s$ steps for XOR in the worst case and $s/2$ steps for AND and $s/2$ steps for XOR in the average case. Then $\mathcal{DB}_1$ evaluates $b_{10}^h = b \oplus \{\bigoplus_{i_2\in S_2^1}(X_{h,i_2}, \vec{T}^1)\}$ for each $h \in [\ell]$. This requires $(\ell+1)\ell$ steps for index pointing, $\ell^2 s$ steps for AND, and $(\ell s+1)\ell$ steps for XOR in the worst case and $\{(\ell/2)+1\}\ell$ steps for index pointing, $(\ell/2)^2(s/2)$ steps for AND and $\{(\ell/2)(s/2)+1\}\ell$ steps for XOR in the average case. The analysis similar to this can be applied to $B_{01}$. Since the number of steps made by $\mathcal{DB}_1$ is the same with that made by $\mathcal{DB}_2$ and $s = \ell = n^{1/3}$, we have that $\mathrm{TIME}^{wc}_{\mathsf{BD}}(n) = 5n + 4n^{2/3} + 9n^{1/3} + 17$ and $\mathrm{TIME}^{av}_{\mathsf{BD}}(n) = (5/4)\cdot n + (3/2)\cdot n^{2/3} + 8n^{1/3} + 17$. □

## 4. Communication-Efficient $k$-PIR

Let $k \geqq 2$. Here wee assume without loss of generality that $n = \ell^{2k-1}$, i.e., $\ell = n^{1/(2k-1)}$, and we embed $x \in \{0,1\}^n$ in a $(2k-1)$-dimensional cube by associating each position $i \in [n]$ with a $(2k-1)$-dimensional tuple $(i_1, i_2, \ldots, i_{2k-1}) \in [\ell]^{2k-1}$ in a natural manner. We also assume that $\mathcal{U}$ wishes to retrieve $x_t$ for some $t \in [n]$ and let $(t_1, t_2, \ldots, t_{2k-1}) \in [\ell]^{2k-1}$ be the associated $(2k-1)$-tuple of $t$.

### 4.1 General Frameworks

We first show general recursive schemes for $k$-PIR ($k$-$\mathsf{PIR}_{\mathsf{GREC}}$) with communication complexity $O(n^{1/(2k-1)})$ that includes the scheme by Ambainis [1] as a special case. For any $d, r \geqq 1$ and any $c \in \{0,1\}^d$, let $B_d(c,r)$ be the set of all $d$-bit long strings that differ from $c$ in at most $r$ positions, i.e., $B_d(c,r) = \{v \in \{0,1\}^d : d(c,v) \leqq r\}$. For any $S = (S_1, S_2, \ldots, S_{2k-1}) \subsetneqq [\ell]^{2k-1}$, let $\vec{S} = (\vec{S}_1, \vec{S}_2, \ldots, \vec{S}_{2k-1}) \in \{0,1\}^{\ell(2k-1)}$. Then for each $\delta \in [2k-1]$, we define

$$\begin{aligned}C_k(S,\delta) = \Bigg\{ &T = (T_1, T_2, \ldots, T_{2k-1}) \in [\ell]^{2k-1} :\\ &\vec{T} \in B_{\ell(2k-1)}(\vec{S},\delta)\ \&\\ &\bigwedge_{j=1}^{2k-1}\left[\vec{T}_j \in B_\ell(\vec{S}_j, 1)\right]\Bigg\}\end{aligned}$$

i.e., $T = (T_1, T_2, \ldots, T_{2k-1}) \in C_k(S,\delta)$ if (1) for each $j \in [2k-1]$, $T_j = S_j \oplus i_j$ for some $i_j \in [\ell]$ or $T_j = S_j$; and (2) $T_j \neq S_j$ happens at most $\delta$ times. Let $\|A\|$ be

the number of elements in a finite set $A$. We number each element of $C_k(S,\delta)$ in a natural order. Obviously,

$$\mathsf{M}(k,\delta) = \|C_k(S,\delta)\| = \sum_{i=0}^{\delta} \binom{2k-1}{i} \ell^i.$$

Recall that $t \in [n]$ is the bit position $\mathcal{U}$ wishes to retrieve and $(t_1, t_2 \ldots, t_{2k-1}) \in [\ell]^{2k-1}$ is the associated $(2k-1)$-tuple of $t$.

For any $S = (S_1, S_2, \ldots, S_{2k-1}) \subseteqq [\ell]^{2k-1}$, define

$$H_k(S,t,\delta) = \Big\{ T = (T_1, T_2, \ldots, T_{2k-1}) \in [\ell]^{2k-1} : \\ T \in C_k(S,\delta) \ \& \\ \forall j \in [2k-1] \ \big[ T_j \neq S_j \\ \Rightarrow T_j = S_j \oplus t_j \big] \Big\},$$

i.e., $T \in H_k(S,t,\delta)$ if $T \in C_k(S,\delta)$ and $T_j$ differs from $S_j$ only on $t_j \in [\ell]$ for $j \in [2k-1]$. From the definition of $H_k(S,t,\delta)$, it is obvious that

$$\mathsf{N}(k,\delta) = \|H_k(S,t,\delta)\| = \sum_{i=0}^{\delta} \binom{2k-1}{i}.$$

Since $H_k(S,t,\delta) \subseteqq C_k(S,\delta)$, each element of $H_k(S,t,\delta)$ has the unique number assigned by the numbering for all elements of $C_k(S,\delta)$. Here we define $P_k(S,t,\delta) = (p_1, p_2, \ldots, p_{\mathsf{N}(k,\delta)})$ to be the set of the numbers that are assigned to all elements of $H_k(S,T,\delta)$.

Let $k_1, k_2 \in [k]$, where $k_1 + k_2 = k$. We present 2-PIR, $(k_1,k_2)$-BASIS, for $\mathcal{U}'$ and $\mathcal{DB}'_1, \mathcal{DB}'_2$ that will be a building block for constructing general recursive $k$-PIR ($k$-$\mathsf{PIR}_{\mathsf{GREC}}$). As noticed above, $k$-$\mathsf{PIR}_{\mathsf{GREC}}$ includes the recursive scheme by Ambainis [1] as a special case.

The Building Block: $(k_1,k_2)$-BASIS

U-1: $\mathcal{U}'$ chooses $S_j^1 \subseteqq [\ell]$ uniformly and independently for each $j \in [2k-1]$.

U-2: $\mathcal{U}'$ computes $S_j^2 = S_j^1 \oplus t_j$ for each $j \in [2k-1]$.

$\mathcal{U}' \to \mathcal{DB}'_1$: $Q_1 = (S_1^1, S_2^1, \ldots, S_{2k-1}^1) \subseteqq [\ell]^{2k-1}$.

$\mathcal{U}' \to \mathcal{DB}'_2$: $Q_2 = (S_1^2, S_2^2, \ldots, S_{2k-1}^2) \subseteqq [\ell]^{2k-1}$.

DB1-1: $\mathcal{DB}'_1$ enumerates $L_h = (L_1^h, \ldots, L_{2k-1}^h) \in C_k(Q_1, 2k_1-1)$ and computes

$$a_h^1 = \bigoplus_{i_1 \in L_1^h} \cdots \bigoplus_{i_{2k-1} \in L_{2k-1}^h} x_{i_1,\ldots,i_{2k-1}},$$

for each $h \in [\mathsf{M}(k, 2k_1-1)]$.

$\mathcal{DB}'_1 \to \mathcal{U}'$: $A_1 = (a_1^1, a_2^1, \ldots, a_{\mathsf{M}(k,2k_1-1)}^1)$.

DB2-1: $\mathcal{DB}'_2$ enumerates $R_h = (R_1^h, \ldots, R_{2k-1}^h) \in C_k(Q_2, 2k_2-1)$ and computes

$$a_h^2 = \bigoplus_{i_1 \in R_1^h} \cdots \bigoplus_{i_{2k-1} \in R_{2k-1}^h} x_{i_1,\ldots,i_{2k-1}},$$

for each $h \in [\mathsf{M}(k, 2k_2-1)]$.

$\mathcal{DB}'_2 \to \mathcal{U}'$: $A_2 = (a_1^2, a_2^2, \ldots, a_{\mathsf{M}(k,2k_2-1)}^2)$.

U-3: $\mathcal{U}'$ obtains $\{\bigoplus_{i \in P_k(Q_1,t,2k_1-1)} \mathsf{pick}_i(A_1)\} \oplus \{\bigoplus_{i \in P_k(Q_2,t,2k_2-1)} \mathsf{pick}_i(A_2)\}$ $(= x_t)$.

The intuition behind $(k_1,k_2)$-BASIS is as follows: From Proposition 2.6 and the definition of $C_k(S,\delta)$, it follows that $A_1$ and $A_2$ include enough information to retrieve $x_t$. After receiving $A_1$ and $A_2$, $\mathcal{U}$ retrieves $x_t$ by picking appropriate bits from $A_1$ and $A_2$.

**Lemma 4.1:** The building block $(k_1,k_2)$-BASIS is private information retrieval for 2 databases.

**Proof:** For each $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_{2k-1}) \in \{1,2\}^{2k-1}$, let $w_1(\sigma)$ be the number of $j \in [2k-1]$ such that $\sigma_j = 1$ and $w_2(\sigma)$ the number of $j \in [2k-1]$ such that $\sigma_j = 2$. We define $\Sigma_{k_1} = \{\sigma \in \{1,2\}^{2k-1} : w_2(\sigma) \leqq 2k_1 - 1\}$ and $\Sigma_{k_2} = \{\sigma \in \{1,2\}^{2k-1} : w_1(\sigma) \leqq 2k_2 - 1\}$. For any $\sigma$, if $w_2(\sigma) \leqq 2k_1 - 1$, then $w_1(\sigma) = 2k - 1 - w_2(\sigma) \geqq 2k - 1 - (2k_1 - 1) = 2(k - k_1) = 2k_2$, because $w_1(\sigma) + w_2(\sigma) = 2k-1$ and $k_1 + k_2 = k$. Thus we have that $\Sigma_{k_1} = \{\sigma \in \{1,2\}^{2k-1} : w_1(\sigma) \geqq 2k_2\}$, and this implies that $\Sigma_{k_1}, \Sigma_{k_2}$ is a bipartition of $\{1,2\}^{2k-1}$.

Let $\mathcal{Q}_k = \{S_\sigma = (S_1^{\sigma_1}, S_2^{\sigma_2}, \ldots, S_{2k-1}^{\sigma_{2k-1}}) \in [\ell]^{2k-1} : \sigma = (\sigma_1, \sigma_2, \ldots, \sigma_{2k-1}) \in \{1,2\}^{2k-1}\}$ be the set of questions for which Proposition 2.5 holds, and define

$$\mathcal{Q}_{k_1} = \{S_\sigma \in \mathcal{Q}_k : \sigma \in \Sigma_{k_1}\};$$
$$\mathcal{Q}_{k_2} = \{S_\sigma \in \mathcal{Q}_k : \sigma \in \Sigma_{k_2}\}.$$

Then $\mathcal{Q}_{k_1}, \mathcal{Q}_{k_2}$ is a bipartition of $\mathcal{Q}_k$, because $\Sigma_{k_1}, \Sigma_{k_2}$ is a bipartition of $\{1,2\}^{2k-1}$. From the definition of $H_k(S,t,\delta)$, $\mathcal{Q}_{k_1}$, and $\mathcal{Q}_{k_2}$, it is obvious that $\mathcal{Q}_{k_1} = H_k(Q_1,t,2k_1-1)$ and $\mathcal{Q}_{k_2} = H_k(Q_2,t,2k_2-1)$. Note that $P_k(Q_1,t,2k_1-1)$ and $P_k(Q_2,t,2k_2-1)$ respectively characterize the position for each element of $H_k(Q_1,t,2k_1-1)$ and $H_k(Q_2,t,2k_2-1)$. Then it follows from Proposition 2.5 that

$$x_t = \left\{ \bigoplus_{i \in P_k(Q_1,t,2k_1-1)} \mathsf{pick}_i(A_1) \right\} \\ \oplus \left\{ \bigoplus_{i \in P_k(Q_2,t,2k_2-1)} \mathsf{pick}_i(A_2) \right\}.$$

Obviously, $Q_1 = (S_1^1, S_2^1, \ldots, S_{2k-1}^1) \subseteqq [\ell]^{2k-1}$ and $Q_2 = (S_1^2, S_2^2, \ldots, S_{2k-1}^2) \subseteqq [\ell]^{2k-1}$ are uniformly distributed over $[\ell]^{2k-1}$ for any $t \in [n]$. Thus the building block $(k_1,k_2)$-BASIS is 2-PIR. □

Now we are ready to show general recursive $k$-PIR,

$k$-$\mathsf{PIR}_{\mathsf{GREC}}$, for $\mathcal{U}$, $\mathcal{DB}(k_1) = (\mathcal{DB}_1^1, \ldots, \mathcal{DB}_{k_1}^1)$, and $\mathcal{DB}(k_2) = (\mathcal{DB}_1^2, \ldots, \mathcal{DB}_{k_2}^2)$. Recall that $k_1, k_2 \in [k]$ such that $k_1 + k_2 = k$.

The General Recursive Scheme: $k$-$\mathsf{PIR}_{\mathsf{GREC}}$

U-1: $\mathcal{U}$ simulates $\mathcal{U}'$ of $(k_1, k_2)$-$\mathsf{BASIS}$ to generate $Q_1, Q_2 \subseteqq [\ell]^{2k-1}$.

$\mathcal{U} \to \mathcal{DB}(k_1)$: $Q_1 \subseteqq [\ell]^{2k-1}$.

$\mathcal{U} \to \mathcal{DB}(k_2)$: $Q_2 \subseteqq [\ell]^{2k-1}$.

DB($k_1$)-1: $\mathcal{DB}(k_1)$ simulates $\mathcal{DB}_1'$ of $(k_1, k_2)$-$\mathsf{BASIS}$ on $Q_1$ to generate $A_1 \in \{0,1\}^{\mathsf{M}(k,2k_1-1)}$.

DB($k_2$)-1: $\mathcal{DB}(k_2)$ simulates $\mathcal{DB}_2'$ of $(k_1, k_2)$-$\mathsf{BASIS}$ on $Q_2$ to generate $A_2 \in \{0,1\}^{\mathsf{M}(k,2k_2-1)}$.

$\mathcal{U} \leftrightarrow \mathcal{DB}(k_1)$: For each $h \in P_k(Q_1, t, 2k_1 - 1)$, $\mathcal{U}$ and $\mathcal{DB}(k_1)$ run $k_1$-$\mathsf{PIR}_{\mathsf{GREC}}$ on $A_1$ to get $a_h^1$.

$\mathcal{U} \leftrightarrow \mathcal{DB}(k_2)$: For each $h \in P_k(Q_2, t, 2k_2 - 1)$, $\mathcal{U}$ and $\mathcal{DB}(k_2)$ run $k_2$-$\mathsf{PIR}_{\mathsf{GREC}}$ on $A_2$ to get $a_h^2$.

U-2: $\mathcal{U}$ computes $\{\bigoplus_{h \in P_k(Q_1,t,2k_1-1)} a_h^1\} \oplus \{\bigoplus_{h \in P_k(Q_2,t,2k_2-1)} a_h^2\}$ $(= x_t)$.

The intuition behind $k$-$\mathsf{PIR}_{\mathsf{GREC}}$ is as follows: We have already known 2-PIR (e.g., $\mathsf{PIR}_{\mathsf{CC}}$ and $\mathsf{PIR}_{\mathsf{BD}}$ in Sect. 3) with communication complexity $O(n^{1/3})$, and for each $k \geqq 3$, $k$-$\mathsf{PIR}_{\mathsf{GREC}}$ is inductively constructed from $k_1$-$\mathsf{PIR}_{\mathsf{GREC}}$ with communication complexity $O(n^{1/(2k_1-1)})$ and $k_2$-$\mathsf{PIR}_{\mathsf{GREC}}$ with communication complexity $O(n^{1/(2k_2-1)})$. Here we recall that $A_1$ and $A_2$ include enough information for $\mathcal{U}$ to retrieve $x_t$ (as we have noticed above). Since $|A_1| = O(n^{(2k_1-1)/(2k-1)})$ and $|A_2| = O(n^{(2k_2-1)/(2k-1)})$, we can achieve $O(n^{1/(2k-1)})$ communication complexity by picking appropriate bits from $A_1$ and $A_2$.

**Theorem 4.2:** For each $k \geqq 2$, the general recursive scheme $k$-$\mathsf{PIR}_{\mathsf{GREC}}$ is $k$-PIR with communication complexity $O(n^{1/(2k-1)})$.

**Proof:** We prove the theorem by induction on $k$. For convenience, 1-$\mathsf{PIR}_{\mathsf{GREC}}$ for $\mathcal{U}$ and $\mathcal{DB}$ is assumed to be a scheme that $\mathcal{U}$ asks nothing and $\mathcal{DB}$ responds $\mathcal{U}$ with the whole stored data.

Base Stage: Assume that $k = 2$. Since $k_1, k_2 \in [k]$ and $k_1 + k_2 = k$, we have that $k_1 = k_2 = 1$. This implies that $\mathcal{DB}(k_1) = \mathcal{DB}_1$ and $\mathcal{DB}(k_2) = \mathcal{DB}_2$. Note that $(1,1)$-$\mathsf{BASIS}$ is *exactly* the same with $\mathsf{PIR}_{\mathsf{CC}}$ in Sect. 3.1 and that $\mathcal{DB}_1$ and $\mathcal{DB}_2$ respond $\mathcal{U}$ with $A_1$ and $A_2$, respectively. Then it turns out that 2-$\mathsf{PIR}_{\mathsf{GREC}}$ is *exactly* the same with $\mathsf{PIR}_{\mathsf{CC}}$. Thus it follows from Lemma 3.1 that 2-$\mathsf{PIR}_{\mathsf{GREC}}$ is 2-PIR with communication complexity $12n^{1/3} + 2 = O(n^{1/(2\cdot 2-1)})$.

Induction Stage: Let $k \geqq 3$ and assume that for any $2 \leqq k' < k$, $k'$-$\mathsf{PIR}_{\mathsf{GREC}}$ is $k'$-PIR with communication complexity $O(n^{1/(2k'-1)})$. To complete the proof, we show in the following that $k$-$\mathsf{PIR}_{\mathsf{GREC}}$ is $k$-PIR with communication complexity $O(n^{1/(2k-1)})$.

From the induction hypothesis, it follows that $k_1$-$\mathsf{PIR}_{\mathsf{GREC}}$ is $k_1$-PIR with communication complexity $O(n^{1/(2k_1-1)})$ and $k_2$-$\mathsf{PIR}_{\mathsf{GREC}}$ is $k_2$-PIR with communication complexity $O(n^{1/(2k_2-1)})$. For each $j \in [k_1]$, let $\mathcal{Q}_j^1 = \{Q_1, q_j^1(1), q_j^1(2), \ldots, q_j^1(\mathsf{N}(k, 2k_1-1))\}$ be the set of questions made by $\mathcal{U}$ to $\mathcal{DB}_j^1$ and for each $j \in [k_2]$, let $\mathcal{Q}_j^2 = \{Q_2, q_j^2(1), q_j^2(2), \ldots, q_j^2(\mathsf{N}(k, 2k_2-1))\}$ be the set of questions made by $\mathcal{U}$ to $\mathcal{DB}_j^2$ Since $k_1$-$\mathsf{PIR}_{\mathsf{GREC}}$ is $k_1$-PIR, $q_j^1(h)$ is *identically* (and independently) distributed for each $h \in P_k(Q_1, t, 2k_1 - 1)$. Note that $Q_1$ is uniformly distributed for any $t \in [n]$ and is independently distributed of $q_j^1(h)$ for each $h \in P_k(Q_1, t, 2k_1 - 1)$. Then it follows that for any $t \in [n]$, $\mathcal{Q}_j^1$ is *identically* distributed for each $j \in [k_1]$. In a way similar to this, we can show that for any $t \in [n]$, $\mathcal{Q}_j^2$ is *identically* distributed for each $j \in [k_2]$. Thus $k$-$\mathsf{PIR}_{\mathsf{GREC}}$ is $k$-PIR for each $k \geqq 2$.

Finally, we analyze communication complexity of $k$-$\mathsf{PIR}_{\mathsf{GREC}}$. We use $\mathrm{COM}_{\mathsf{GREC}}^k(n)$ to denote communication complexity of $k$-$\mathsf{PIR}_{\mathsf{GREC}}$. Note that $|Q_1| = |Q_2| = (2k-1) \cdot \ell = (2k-1) \cdot n^{1/(2k-1)}$. The definition of $\mathsf{M}(k, \delta)$ guarantees that there exist $c_{k_1}, c_{k_2} > 0$ such that

$$\begin{aligned} |A_1| &= \mathsf{M}(k, 2k_1 - 1) \\ &= \sum_{i=0}^{2k_1-1} \binom{2k-1}{i} \ell^i \leqq c_{k_1} \cdot n^{\frac{2k_1-1}{2k-1}}; \\ |A_2| &= \mathsf{M}(k, 2k_2 - 1) \\ &= \sum_{i=0}^{2k_2-1} \binom{2k-1}{i} \ell^i \leqq c_{k_2} \cdot n^{\frac{2k_2-1}{2k-1}}. \end{aligned}$$

Since $\mathrm{COM}_{\mathsf{GREC}}^{k'}(n) = O(n^{1/(2k'-1)})$ for each $k' < k$ (induction hypothesis), there exist $d_{k_1}, d_{k_2} > 0$ such that

$$\begin{aligned} &\mathrm{COM}_{\mathsf{GREC}}^{k_1}(n) \leqq d_{k_1} \cdot n^{1/(2k_1-1)}; \\ &\mathrm{COM}_{\mathsf{GREC}}^{k_2}(n) \leqq d_{k_2} \cdot n^{1/(2k_2-1)}, \end{aligned}$$

respectively. Then we have that

$$\begin{aligned} &\mathrm{COM}_{\mathsf{GREC}}^{k}(n) \\ &\quad = k_1 \cdot |Q_1| + k_2 \cdot |Q_2| \\ &\qquad + \mathsf{N}(k, 2k_1 - 1) \cdot \mathrm{COM}_{\mathsf{GREC}}^{k_1}(|A_1|) \\ &\qquad + \mathsf{N}(k, 2k_2 - 1) \cdot \mathrm{COM}_{\mathsf{GREC}}^{k_2}(|A_2|) \\ &\quad \leqq k(2k-1) \cdot n^{1/(2k-1)} \\ &\qquad + \mathsf{N}(k, 2k_1 - 1) \cdot d_{k_1} \cdot |A_1|^{1/(2k_1-1)} \\ &\qquad + \mathsf{N}(k, 2k_2 - 1) \cdot d_{k_2} \cdot |A_2|^{1/(2k_2-1)} \\ &\quad < c_k \cdot n^{1/(2k-1)}, \end{aligned}$$

for some $c_k > 0$. Thus for each $k \geqq 2$, $\mathrm{COM}_{\mathsf{GREC}}^{k}(n) = O(n^{1/(2k-1)})$ for any $n > 0$. □

Note that in Theorem 4.2, $\mathrm{COM}_{\mathsf{GREC}}^{k}(n)$ is measured independently of the choices of $k_1, k_2 \in [k]$. To

analyze the inherent behavior of $\mathrm{COM}^k_{\mathsf{GREC}}(n)$ more precisely for each $k \geqq 2$, we define

$$\mathsf{D}^k_{\mathsf{GREC}} = \lim_{n\to\infty} \frac{\mathrm{COM}^k_{\mathsf{GREC}}(n)}{n^{1/(2k-1)}}.$$

It is obvious that $\mathsf{D}^k_{\mathsf{GREC}}$ is the coefficient for the *dominant* factor of $\mathrm{COM}^k_{\mathsf{GREC}}(n)$.

### 4.2 The Unbalanced Recursive Scheme

For each $k \geqq 2$, we assume that $k_1 = k-1$ and $k_2 = 1$. Then $k$-$\mathsf{PIR}_{\mathsf{GREC}}$ coincides with the recursive $k$-PIR by Ambainis [1] and we refer to the *unbalanced* recursive scheme as $k$-$\mathsf{PIR}_{\mathsf{UREC}}$. Let $\mathrm{COM}^k_{\mathsf{UREC}}(n)$ denote communication complexity of $k$-$\mathsf{PIR}_{\mathsf{UREC}}$ for each $k \geqq 2$.

**Theorem 4.3:** For each $k \geqq 4$, $\mathsf{D}^k_{\mathsf{UREC}} \geqq (2k-1)\cdot 2^{k(k-1)}$.

**Proof:** Since $k_1 = k-1$ and $k_2 = 1$ in $k$-$\mathsf{PIR}_{\mathsf{UREC}}$, $\mathcal{U}$ sends $\mathcal{DB}^1_j$ $Q_1 \subsetneqq [\ell]^{2k-1}$ for each $j \in [k_1] = [k-1]$, and $\mathcal{U}$ sends $\mathcal{DB}^2_j$ $Q_2 \subsetneqq [\ell]^{2k-1}$ for $j \in [k_1] = [1]$. Then $\mathcal{U}$ and $\mathcal{DB}^1_1, \mathcal{DB}^1_2, \ldots, \mathcal{DB}^1_{k-1}$ run $(k-1)$-$\mathsf{PIR}_{\mathsf{UREC}}$ $\mathsf{N}(k, 2k-3)$ times to retrieve $a^1_h$ for each $h \in P_k(Q_1, t, 2k-3)$, however, $\mathcal{U}$ asks nothing more and $\mathcal{DB}^2_1$ responds $\mathcal{U}$ with $A_2$ in 1-$\mathsf{PIR}_{\mathsf{UREC}}$. Here we note that $\mathsf{N}(k, 2k-3) = 2^{2k-1} - (2k-1) - 1 = 2^{2k-1} - 2k$ and $|A_2| = \mathsf{M}(k,1) = 1 + (2k-1)\cdot n^{1/(2k-1)}$. Then

$$\begin{aligned}
&\mathrm{COM}^k_{\mathsf{UREC}}(n)\\
&\quad= (k_1+k_2)(2k-1)\cdot n^{1/(2k-1)}\\
&\qquad+ \mathrm{COM}^1_{\mathsf{UREC}}(\mathsf{M}(k,1))\\
&\qquad+ \mathsf{N}(k,2k-3)\cdot \mathrm{COM}^{k-1}_{\mathsf{UREC}}(\mathsf{M}(k,2k-3))\\
&\quad= (2^{2k-1}-2k)\cdot \mathrm{COM}^{k-1}_{\mathsf{UREC}}(\mathsf{M}(k,2k-3))\\
&\qquad+ (2k-1)(k+1)\cdot n^{1/(2k-1)} + 1. \qquad (4)
\end{aligned}$$

From the definitions of $\mathsf{D}^k_{\mathsf{GREC}}$ and $\mathsf{M}(k,\delta)$, we have that

$$\begin{aligned}
&\lim_{n\to\infty} \frac{\mathrm{COM}^{k-1}_{\mathsf{UREC}}(\mathsf{M}(k,2k-3))}{n^{1/(2k-1)}}\\
&\quad= \lim_{n\to\infty}\left[\frac{\{\mathsf{M}(k,2k-3)\}^{1/(2k-3)}}{n^{1/(2k-1)}}\right.\\
&\qquad\left.\times \frac{\mathrm{COM}^{k-1}_{\mathsf{UREC}}(\mathsf{M}(k,2k-3))}{\{\mathsf{M}(k,2k-3)\}^{1/(2k-3)}}\right]\\
&\quad= \binom{2k-1}{2k-3}^{1/(2k-3)}\cdot \mathsf{D}^{k-1}_{\mathsf{UREC}}. \qquad (5)
\end{aligned}$$

Note that $2^{2k-1} - 2k \geqq 2^{2k-2}$ for each $k \geqq 2$ and that $\binom{n}{k} \geqq (n/k)^k$ [9, Proposition B.2]. Then from Eqs. (4) and (5), it follows that for each $k \geqq 4$,

$$\begin{aligned}
\mathsf{D}^k_{\mathsf{UREC}} &= \lim_{n\to\infty} \frac{\mathrm{COM}^k_{\mathsf{UREC}}(n)}{n^{1/(2k-1)}}\\
&= \lim_{n\to\infty}\left\{(2k-1)(k+1) + \frac{1}{n^{1/(2k-1)}}\right.\\
&\quad+ (2^{2k-1}-2k)\\
&\quad\left.\times \frac{\mathrm{COM}^{k-1}_{\mathsf{UREC}}(\mathsf{M}(k,2k-3))}{n^{1/(2k-1)}}\right\}\\
&= (2^{2k-1}-2k)\cdot\binom{2k-1}{2k-3}^{1/(2k-3)}\cdot \mathsf{D}^{k-1}_{\mathsf{UREC}}\\
&\quad+ (2k-1)(k+1) \qquad (6)\\
&\geqq (2^{2k-1}-2k)\cdot\binom{2k-1}{2k-3}^{1/(2k-3)}\cdot \mathsf{D}^{k-1}_{\mathsf{UREC}}\\
&\geqq 2^{2k-2}\cdot\binom{2k-1}{2k-3}^{1/(2k-3)}\cdot \mathsf{D}^{k-1}_{\mathsf{UREC}}\\
&\geqq 2^{2k-2}\cdot\frac{2k-1}{2k-3}\cdot \mathsf{D}^{k-1}_{\mathsf{UREC}}. \qquad (7)
\end{aligned}$$

Since 2-$\mathsf{PIR}_{\mathsf{UREC}}$ is exactly the same with $\mathsf{PIR}_{\mathsf{CC}}$, we have that $\mathrm{COM}^2_{\mathsf{UREC}}(n) = 12n^{1/3} + 2$, and thus $\mathsf{D}^2_{\mathsf{UREC}} = 12$. Then it follows from Eq. (7) that

$$\begin{aligned}
\mathsf{D}^k_{\mathsf{UREC}} &\geqq \left\{\prod_{i=2}^{k-1}\left(2^{2i}\cdot\frac{2i+1}{2i-1}\right)\right\}\cdot \mathsf{D}^2_{\mathsf{UREC}}\\
&= \frac{2k-1}{3}\cdot 2^{(k+1)(k-2)}\cdot 12.
\end{aligned}$$

Hence $\mathsf{D}^k_{\mathsf{UREC}} \geqq (2k-1)\cdot 2^{k(k-1)}$ for each $k \geqq 4$. □

### 4.3 The Balanced Recursive Scheme

In this subsection, we present new $k$-PIR with (much) less communication complexity than $k$-$\mathsf{PIR}_{\mathsf{UREC}}$ by taking $k_1 = \lceil k/2 \rceil$ and $k_2 = k - \lceil k/2 \rceil$. The way of taking $k_1, k_2 \in [k]$ is more *balanced* than that of $k$-$\mathsf{PIR}_{\mathsf{UREC}}$. We refer to the *balanced* recursive scheme as $k$-$\mathsf{PIR}_{\mathsf{BREC}}$ and use $\mathrm{COM}^k_{\mathsf{BREC}}(n)$ to denote communication complexity of $k$-$\mathsf{PIR}_{\mathsf{BREC}}$ for each $k \geqq 2$.

In $k$-$\mathsf{PIR}_{\mathsf{UREC}}$, we take $k_1 = k-1$ and $k_2 = 1$, and this causes $k$-$\mathsf{PIR}_{\mathsf{UREC}}$ to recursively call $k'$-$\mathsf{PIR}_{\mathsf{UREC}}$ (for $k' < k$) $k$ times. On the other hand, $k$-$\mathsf{PIR}_{\mathsf{BREC}}$ needs to recursively call $k'$-$\mathsf{PIR}_{\mathsf{BREC}}$ (for $k' < k$) only $\lg k$ times because of its balanced way of taking $k_1, k_2 \in [k]$. Intuitively, the difference between $k$-$\mathsf{PIR}_{\mathsf{UREC}}$ and $k$-$\mathsf{PIR}_{\mathsf{BREC}}$ provides a huge gap between $\mathsf{D}^k_{\mathsf{UREC}}$ and $\mathsf{D}^k_{\mathsf{BREC}}$.

**Theorem 4.4:** For each $k \geqq 4$, $\mathsf{D}^k_{\mathsf{BREC}} < k\cdot 2^{5k}$.

**Proof:** We first note that 2-$\mathsf{PIR}_{\mathsf{BREC}}$ is the same with $\mathsf{PIR}_{\mathsf{CC}}$ and 3-$\mathsf{PIR}_{\mathsf{BREC}}$ is the same with 3-$\mathsf{PIR}_{\mathsf{UREC}}$. Then from Lemma 3.1, it follows that $\mathsf{D}^2_{\mathsf{BREC}} = 12 < 2048 = 2\cdot 2^{5\times 2}$ and from Eq. (6), it follows that $\mathsf{D}^3_{\mathsf{BREC}} = 692 < 98304 = 3\cdot 2^{5\times 3}$.

For each $k \geqq 4$, let $k_1 = \lceil k/2 \rceil$ and $k_2 = k - \lceil k/2 \rceil$. For each $j \in [k_1]$, $\mathcal{U}$ sends $Q_1 \subsetneqq [\ell]^{2k-1}$ to $\mathcal{DB}^1_j$ and for

each $j \in [k_2]$, $\mathcal{U}$ sends $Q_2 \subseteqq [\ell]^{2k-1}$ to $\mathcal{DB}_j^2$. Then $\mathcal{U}$ and $\mathcal{DB}(k_1)$ run $k_1$-$\mathsf{PIR}_{\mathsf{UREC}}$ $\mathsf{N}(k, 2k_1-1)$ times to get $a_h^1$ for each $h \in P_k(Q_1, t, 2k_1-1)$, and $\mathcal{U}$ and $\mathcal{DB}(k_2)$ run $k_2$-$\mathsf{PIR}_{\mathsf{UREC}}$ $\mathsf{N}(k, 2k_2-1)$ times to get $a_h^2$ for each $h \in P_k(Q_2, t, 2k_2-1)$. Thus it follows that

$$\begin{aligned}
&\mathrm{COM}^k_{\mathsf{BREC}}(n) \\
&\quad = k(2k-1)\cdot n^{1/(2k-1)} \\
&\qquad + \mathsf{N}(k,2k_1-1)\cdot \mathrm{COM}^{k_1}_{\mathsf{BREC}}(\mathsf{M}(k,2k_1-1)) \\
&\qquad + \mathsf{N}(k,2k_2-1)\cdot \mathrm{COM}^{k_2}_{\mathsf{BREC}}(\mathsf{M}(k,2k_2-1)).
\end{aligned} \tag{8}$$

From the definition of $\mathsf{D}^k_{\mathsf{GREC}}$ and $\mathsf{M}(k,\delta)$, we have that

$$\begin{aligned}
&\lim_{n\to\infty} \frac{\mathrm{COM}^{k_1}_{\mathsf{BREC}}(\mathsf{M}(k,2k_1-1))}{n^{1/(2k-1)}} \\
&\quad = \lim_{n\to\infty}\left[\frac{\{\mathsf{M}(k,2k_1-1)\}^{1/(2k_1-1)}}{n^{1/(2k-1)}}\right. \\
&\qquad \left.\times \frac{\mathrm{COM}^{k_1}_{\mathsf{BREC}}(\mathsf{M}(k,2k_1-1))}{\{\mathsf{M}(k,2k_1-1)\}^{1/(2k_1-1)}}\right] \\
&\quad = \binom{2k-1}{2k_1-1}^{1/(2k_1-1)} \cdot \mathsf{D}^{k_1}_{\mathsf{BREC}};
\end{aligned} \tag{9}$$

$$\begin{aligned}
&\lim_{n\to\infty} \frac{\mathrm{COM}^{k_2}_{\mathsf{BREC}}(\mathsf{M}(k,2k_2-1))}{n^{1/(2k-1)}} \\
&\quad = \lim_{n\to\infty}\left[\frac{\{\mathsf{M}(k,2k_2-1)\}^{1/(2k_2-1)}}{n^{1/(2k-1)}}\right. \\
&\qquad \left.\times \frac{\mathrm{COM}^{k_2}_{\mathsf{BREC}}(\mathsf{M}(k,2k_2-1))}{\{\mathsf{M}(k,2k_2-1)\}^{1/(2k_2-1)}}\right] \\
&\quad = \binom{2k-1}{2k_2-1}^{1/(2k_2-1)} \cdot \mathsf{D}^{k_2}_{\mathsf{BREC}}.
\end{aligned} \tag{10}$$

Thus from the definition of $\mathsf{D}^k_{\mathsf{GREC}}$ and Eqs. (8) to (10), it follows that

$$\begin{aligned}
\mathsf{D}^k_{\mathsf{BREC}} &= \lim_{n\to\infty} \frac{\mathrm{COM}^k_{\mathsf{BREC}}(n)}{n^{1/(2k-1)}} \\
&= k(2k-1) \\
&\quad + \mathsf{N}(k,2k_1-1) \\
&\quad \times \left\{\lim_{n\to\infty} \frac{\mathrm{COM}^{k_1}_{\mathsf{BREC}}(\mathsf{M}(k,2k_1-1))}{n^{1/(2k-1)}}\right\} \\
&\quad + \mathsf{N}(k,2k_2-1) \\
&\quad \times \left\{\lim_{n\to\infty} \frac{\mathrm{COM}^{k_2}_{\mathsf{BREC}}(\mathsf{M}(k,2k_2-1))}{n^{1/(2k-1)}}\right\} \\
&= k(2k-1) \\
&\quad + \mathsf{N}(k,2k_1-1) \\
&\quad \times \binom{2k-1}{2k_1-1}^{1/(2k_1-1)} \cdot \mathsf{D}^{k_1}_{\mathsf{BREC}} \\
&\quad + \mathsf{N}(k,2k_2-1) \\
&\quad \times \binom{2k-1}{2k_2-1}^{1/(2k_2-1)} \cdot \mathsf{D}^{k_2}_{\mathsf{BREC}},
\end{aligned} \tag{11}$$

for each $k \geqq 4$. In the following, we show the theorem by induction on $k \geqq 4$.

Base Stage: Let $k=4$ and we have that $k_1 = k_2 = 2$. From the definition of $\mathsf{N}(k,\delta)$, it immediate to see that $\mathsf{N}(4,3) = 64$. Thus it follows from Eq. (11) that $\mathsf{D}^4_{\mathsf{BREC}} = 5052 < 4194304 = 4\cdot 2^{5\times 4}$.

Induction Stage: Let $k \geqq 5$ and assume that $\mathsf{D}^{k'}_{\mathsf{BREC}} \leqq k'\cdot 2^{5k'}$ for each $2 \leqq k' \leqq k-1$. Note that $2k_2-1 \leqq 2\cdot\{k-(k/2)\}-1 = k-1 \leqq (2k-1)/2$. Then we have that for each $k \geqq 5$,

$$\mathsf{N}(k,2k_1-1) = \sum_{i=0}^{2k_1-1}\binom{2k-1}{i} < \frac{3}{4}\cdot 2^{2k-1}; \tag{12}$$

$$\mathsf{N}(k,2k_2-1) = \sum_{i=0}^{2k_2-1}\binom{2k-1}{i} < 2^{2k-2}. \tag{13}$$

We also note that $2k_1-1 \geqq k-1 = (2k-2)/2$ and $2k_2-1 \geqq 2\cdot\{k-(k/2)-(1/2)\}-1 = k-2 \geqq (2k-2)/3$ for each $k \geqq 5$. Then we have that for each $k \geqq 5$,

$$\begin{aligned}
&\lim_{n\to\infty} \frac{\{\mathsf{M}(k,2k_1-1)\}^{1/(2k_1-1)}}{n^{1/(2k-1)}} \\
&\quad = \binom{2k-1}{2k_1-1}^{1/(2k_1-1)} \\
&\quad < 2^{\frac{2k-2}{2k_1-1}} \leqq 4;
\end{aligned} \tag{14}$$

$$\begin{aligned}
&\lim_{n\to\infty} \frac{\{\mathsf{M}(k,2k_2-1)\}^{1/(2k_2-1)}}{n^{1/(2k-1)}} \\
&\quad = \binom{2k-1}{2k_2-1}^{1/(2k_2-1)} \\
&\quad < 2^{\frac{2k-2}{2k_2-1}} \leqq 8.
\end{aligned} \tag{15}$$

Thus from Eq. (11) and Eqs. (12) to (15), it immediately follows that

$$\begin{aligned}
\mathsf{D}^k_{\mathsf{BREC}} &< k(2k-1) + \frac{3}{4}\cdot 2^{2k+1}\cdot \mathsf{D}^{k_1}_{\mathsf{BREC}} \\
&\quad + 2^{2k+1}\cdot \mathsf{D}^{k_2}_{\mathsf{BREC}}.
\end{aligned}$$

From the assumption that $\mathsf{D}^{k'}_{\mathsf{BREC}} < k'\cdot 2^{5k'}$ for each $2 \leqq k' < k$, it turns out that

$$\begin{aligned}
\mathsf{D}^k_{\mathsf{BREC}} &< k(2k-1) + \frac{3}{4}\cdot 2^{2k+1}\cdot k_1\cdot 2^{5k_1} \\
&\quad + 2^{2k+1}\cdot k_2\cdot 2^{5k_2} \\
&= k\cdot 2^{5k}\cdot\left\{\frac{2k-1}{2^{5k}} + \frac{3}{4}\cdot\frac{k_1}{k}\cdot 2^{5k_1+1-3k}\right. \\
&\quad \left. + \frac{k_2}{k}\cdot 2^{5k_2+1-3k}\right\} \\
&\leqq k\cdot 2^{5k},
\end{aligned}$$

because $2 \leqq k_2 \leqq k_1 < k$ for $k \geqq 5$. Thus we have that $\mathsf{D}^k_{\mathsf{BREC}} < k\cdot 2^{5k}$ for each $k \geqq 4$. □

**Theorem 4.5:** For each $k \geqq 4$, $\mathsf{D}^k_{\mathsf{BREC}} < \mathsf{D}^k_{\mathsf{UREC}}$.

**Proof:** Recall that $\mathsf{D}^2_{\mathsf{UREC}} = \mathsf{D}^2_{\mathsf{BREC}} = 12$ (Lemma 3.1). From Eqs. (6) and (11), it is immediate that

$$\mathsf{D}^4_{\mathsf{BREC}} = 5052 < 152741 = \mathsf{D}^4_{\mathsf{UREC}};$$
$$\mathsf{D}^5_{\mathsf{BREC}} = 702492 < 127934898 = \mathsf{D}^5_{\mathsf{UREC}}.$$

For each $k \geqq 6$, it follows from Theorems 4.3 and 4.4 that $\mathsf{D}^k_{\mathsf{BREC}} < k \cdot 2^{5k} < (2k-1) \cdot 2^{k(k-1)} \leqq \mathsf{D}^k_{\mathsf{UREC}}$. Thus we have that $\mathsf{D}^k_{\mathsf{BREC}} < \mathsf{D}^k_{\mathsf{UREC}}$ for each $k \geqq 4$. □

## 5. Concluding Remarks

In this paper, we have presented 2-PIR ($\mathsf{PIR}_{\mathsf{BD}}$) with communication complexity $12n^{1/3}$ that is more time-efficient than 2-PIR ($\mathsf{PIR}_{\mathsf{CC}}$ [1]). Then for each $k \geqq 2$, we have generally formulated (unbalanced) recursive scheme $k$-PIR ($k$-$\mathsf{PIR}_{\mathsf{UREC}}$) given by Ambainis [1] and have presented (balanced) recursive scheme $k$-PIR ($k$-$\mathsf{PIR}_{\mathsf{BREC}}$) with communication complexity $O(n^{1/(2k-1)})$ that is more communication-efficient than $k$-$\mathsf{PIR}_{\mathsf{UREC}}$.

Recall that $k$-$\mathsf{PIR}_{\mathsf{BREC}}$ (and $k$-$\mathsf{PIR}_{\mathsf{UREC}}$) is constructed from $\mathsf{PIR}_{\mathsf{CC}}$ for each $k \geqq 2$. Then it seems natural to expect that we could reduce communication complexity of $k$-$\mathsf{PIR}_{\mathsf{BREC}}$ (and/or $k$-$\mathsf{PIR}_{\mathsf{UREC}}$) if we would have more communication-efficient 2-PIR. Actually, we can show that if there exists 2-PIR with communication complexity $O(n^{1/(3+e)})$ for some integer $e \geqq 1$, then for each $k \geqq 2$, there exists $k$-PIR with communication complexity $O(n^{1/(2k-1+e)})$. Then we have

(1) For some $\varepsilon > 0$, find 2-PIR with communication complexity $O(n^{1/(3+\varepsilon)})$.

(2) For some $\varepsilon > 0$, find $k$-PIR with communication complexity $O(n^{1/(2k-1+\varepsilon)})$ for each $k \geqq 3$.

(3) Show a (nontrivial) lower bound on communication complexity of 2-PIR.

(4) Show a (nontrivial) lower bound on communication complexity of $k$-PIR for each $k \geqq 3$.

Presumably, time complexity of $k$-$\mathsf{PIR}_{\mathsf{BREC}}$ is much less than that of $k$-$\mathsf{PIR}_{\mathsf{UREC}}$, but we have not analyzed them due to their recursive structures. Then we finally have

(5) Analyze the time complexity of $k$-$\mathsf{PIR}_{\mathsf{BREC}}$ and $k$-$\mathsf{PIR}_{\mathsf{BREC}}$ for each $k \geqq 3$.

## Acknowledgments

The author thanks anonymous referees for their several valuable comments, especially for the comments on the analysis for time complexity of 2-PIR.

## References

[1] A. Ambainis, "Upper bound on the communication complexity of private information retrieval," Proc. ICALP, Lecture Notes in Computer Science, vol.1256, pp.401–409, 1997.
[2] M. Ben-Or, S. Goldwasser, J. Kilian, and A. Wigderson, "Multi-prover interactive proofs: How to remove intractability assumptions," Proc. 20th ACM Symposium on Theory of Computing, pp.113–131, 1988.
[3] B. Chor and N. Gilboa, "Computationally information retrieval," Proc. 29th ACM Symposium on Theory of Computing, pp.304–313, 1997.
[4] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," Proc. 36th IEEE Symposium on Foundations of Computer Science, pp.41–50, 1995.
[5] S. Goldwasser and S. Micali, "Probabilistic encryption," Journal of Computer and System Sciences, vol.28, pp.270–299, 1984.
[6] J. Håstad, "Pseudo-random generators with uniform assumptions," Proc. 22nd ACM Symposium on Theory of Computing, pp.395–404, 1990.
[7] R. Impagliazzo, L.A. Levin, and M. Luby, "Pseudo-random generation from one-way functions," Proc. 30th IEEE Symposium on Foundations of Computer Science, pp.12–24, 1989.
[8] E. Kushilevitz and R. Ostrovsky, "Replication is not needed: Single database, computationally-private information retrieval," Proc. 38th IEEE Symposium on Foundations of Computer Science, pp.364–373, 1997.
[9] R. Motwani and P. Raghavan, "Randomized Algorithms," Cambridge University Press, 1995.
[10] F.J. MacWilliams and N.J.A. Sloane, "The Theory of Error-Correcting Codes," 9th edition, North-Holland, 1996.

**Toshiya Itoh** was born in Urawa, Japan, in 1959. He received the B. Eng., the M.S. Eng., and the Dr. Eng. degree in electronic engineering in 1982, 1984, and 1988, respectively from Tokyo Institute of Technology, Tokyo, Japan. From 1985 to 1990, he was an Assistant Professor in the Department of Electrical and Electronic Engineering at Tokyo Institute of Technology, and from 1990 to 1992, he was a Lecturer in the Department of Information Processing at Tokyo Institute of Technology. Since 1992, he has been an Associate Professor in the Department of Information Processing at Tokyo Institute of Technology. His current interests are modern cryptographies, finite field arithmetics, and complexity theory. Dr. Itoh is a member of the Information Processing Society of Japan, the International Association for Cryptologic Research, the Association for Computing Machinery, and LA.