New Generic Constructions of Error-Correcting PIR and Efficient Instantiations

Reo Eriguchi[1,4], Kaoru Kurosawa[2,4], and Koji Nuida[3,4]

[1] Graduate School of Information Science and Technology, The University of Tokyo, Tokyo, Japan
reo-eriguchi@g.ecc.u-tokyo.ac.jp
[2] Research and Development Initiative, Chuo University, Tokyo, Japan
kaoru.kurosawa.kk@vc.ibaraki.ac.jp
[3] Institute of Mathematics for Industry, Kyushu University, Fukuoka, Japan
nuida@imi.kyushu-u.ac.jp
[4] National Institute of Advanced Industrial Science and Technology, Tokyo, Japan

Abstract. A b-error-correcting m-server Private Information Retrieval (PIR) protocol enables a client to privately retrieve a data item of a database from m servers even in the presence of b malicious servers. List-decodable PIR is a generalization of error-correcting PIR to achieve a smaller number of servers at the cost of giving up unique decoding. Previous constructions of error-correcting and list-decodable PIR have exponential computational complexity in m or cannot achieve sub-polynomial communication complexity $n^{o(1)}$, where n is the database size. Recently, Zhang, Wang and Wang (ASIACCS 2022) presented a non-explicit construction of error-correcting PIR with $n^{o(1)}$ communication and polynomial computational overhead in m. However, their protocol requires the number of servers to be larger than the minimum one $m = 2b + 1$ and they left it as an open problem to reduce it. As for list-decodable PIR, there is no construction with $n^{o(1)}$ communication.

In this paper, we propose new generic constructions of error-correcting and list-decodable PIR from any one-round regular PIR. Our constructions increase computational complexity only by a polynomial factor in m while the previous generic constructions incur $\binom{m}{b}$ multiplicative overheads. Instantiated with the best-known protocols, our construction provides for the first time an explicit error-correcting PIR protocol with $n^{o(1)}$ communication, which reduces the number of servers of the protocol by Zhang, Wang and Wang (ASIACCS 2022). For sufficiently large b, we also show the existence of b-error-correcting PIR with $n^{o(1)}$ communication achieving the minimum number of servers, by allowing for two rounds of interaction. Furthermore, we show an extension to list-decodable PIR and obtain for the first time a protocol with $n^{o(1)}$ communication. Other instantiations improve the communication complexity of the state-of-the-art t-private protocols in which t servers may collude. Along the way, we formalize the notion of *locally surjective map families*, which generalize perfect hash families and may be of independent interest.

Keywords: Private information retrieval · Error correction · List decoding

1 Introduction

Private Information Retrieval (PIR) [10] is a cryptographic primitive that enables a client to retrieve a data item a_τ of his choice from a database $\boldsymbol{a} = (a_1, \ldots, a_n)$ while hiding the identity τ from m servers who hold copies of the database. PIR has many real-world applications including private messaging [27,25], password checkup [29], private media browsing [19] and Safe Browsing [21]. A trivial solution is to send the entire database $\boldsymbol{a}$ to the client, whose communication complexity is proportional to the database size n. When $m = 1$, the trivial solution cannot be improved since it was shown in [10] that an information-theoretically secure single-server PIR protocol must have communication complexity $\Omega(n)$. Single-server PIR protocols were then constructed based on computational assumptions (e.g., [24,26,11] and references therein). This paper focuses on information-theoretically secure multi-server PIR protocols, which are typically more efficient than single-server protocols. Chor et al. [10] considered PIR protocols with $o(n)$ communication assuming $m \geq 2$ non-colluding servers. The privacy requirement is naturally generalized to t-private PIR, which keeps τ private from any collusion of t out of m servers. Since then, many multi-server PIR protocols have been developed to improve communication complexity [1,5,6,9,10,13,14,20,32].

As more servers are involved, there is a higher possibility that servers return incorrect answers. For example, servers may be malicious or faulty, or compute answers from an out-of-date copy of the database. Beimel and Stahl [7] introduced b-error-correcting PIR, in which a client can obtain a correct value a_τ even in the presence of b malicious servers. It is known that b-error correction is possible only if $m \geq 2b + 1$ [7]. To further reduce the number of servers (or equivalently handle a larger number of errors), Goldberg [17] proposed a generalized notion of (b, L)-list-decodable PIR, which allows a client to output a list of L possibilities one of which is correct. Indeed, (b, L)-list-decoding is possible even for a smaller number of servers $m > b(L + 1)/L$.[5]

First of all, there are generic constructions of error-correcting PIR. Beimel and Stahl [7] proposed a generic construction of b-error-correcting m-server PIR from any regular k-server PIR for $m \geq 2b+k$. Recently, it was shown in [16] that the number of servers can be reduced if a small probability ϵ that a client fails to obtain a correct value is allowed[6]. However, the constructions of [7,16] increase client-side computational complexity by $\binom{m}{b}$ times, which is exponential in general. The exponential multiplicative overheads can be serious in practice if we have to tolerate a large number of malicious servers. Several works then directly constructed error-correcting and list-decodable PIR protocols whose communication and computational complexity is polynomial in m.

[5] This result is a folklore and is included in Appendix A for completeness.

[6] We call error-correcting or list-decodable PIR *perfect* if $\epsilon = 0$ and *statistical* if $\epsilon > 0$.

For $t \geq 1$, Kurosawa [23] showed a t-private perfect b-error-correcting PIR protocol. The number of servers was then reduced at the cost of statistical error correction [16]. These protocols cannot attain sub-polynomial communication complexity $n^{o(1)}$. Recently, Zhang, Wang and Wang [34] presented a 1-private perfect b-error-correcting protocol with $n^{o(1)}$ communication. However, the protocol in [34] is not explicitly given and requires the number of servers to be at least $m \geq 8b + 4$. Constructing an explicit protocol with a smaller number of servers than [34] is an open problem. More generally, it is unknown whether there exists a b-error-correcting m-server PIR protocol with $n^{o(1)}$ communication achieving both the minimum number of servers $m = 2b + 1$ and polynomial computational complexity in m.

As for list-decodable PIR, Goldberg [17] proposed a t-private perfect (b, L)-list-decodable protocol for $t \geq 1$. The number of servers was then reduced at the cost of statistical error correction [12]. However, the list-decodable protocols of [17,12] only achieve communication complexity $\sqrt{n}$, omitting a polynomial factor in m. It is an open problem whether there exists a (b, L)-list-decodable m-server PIR protocol with $o(\sqrt{n})$ communication for $b(L + 1)/L < m \leq 2b$.

1.1 Our Contributions

In this paper, we propose new generic constructions of error-correcting and list-decodable PIR protocols from any one-round regular PIR protocol, with different trade-offs. Our constructions only incur polynomial multiplicative overheads in m to communication and computational complexity. We refer to Table 1 for a comparison between the previous generic constructions. Our technical novelty is devising different techniques to verify the correctness of answers from servers. We successfully get around the necessity of carrying out an exhaustive search over all the possible sets of honest servers, which caused the $\binom{m}{b}$ multiplicative overheads in [7,16]. Along the way, we introduce a new combinatorial object called a *locally surjective map family*, which is an extension of a perfect hash family and may be of independent interest. More technical details will be provided in Section 2.

By instantiating our generic constructions with existing regular PIR protocols, we advance the state of the art in error-correcting and list-decodable PIR.

Error-Correcting PIR. We propose for the first time an explicit 1-private perfect b-error-correcting PIR protocol with $n^{o(1)}$ communication, which reduces the number of servers in [34] by half. We also propose a non-explicit protocol, further reducing the number of servers. Moreover, for sufficiently large b, we non-explicitly show the existence of a 1-private statistical b-error-correcting PIR protocol with $n^{o(1)}$ communication achieving both the minimum number of servers $m = 2b + 1$ and polynomial computational complexity in m, by allowing for two rounds of interaction. For $t \geq 1$, we improve the communication complexity of the t-private error-correcting protocol in [16] in terms of n, by allowing for $O(m^2)$ rounds of interaction. See Table 2 for a comparison.

List-Decodable PIR. For sufficiently large b, we non-explicitly show the existence of a 1-private statistical (b, L)-list-decodable PIR protocol with $n^{o(1)}$ communication achieving both the minimum number of servers $m > b(L+1)/L$ and polynomial computational complexity in m, by allowing for two rounds of interaction. Note that this is the first list-decodable PIR protocol with $n^{o(1)}$ communication for $b(L+1)/L < m \leq 2b$. In addition, for $t \geq 1$, we non-explicitly present a two-round t-private statistical list-decodable protocol with $o(\sqrt{n})$ communication. See Table 3 for a comparison.

In what follows, we discuss our contributions in more detail.

Table 1. Generic constructions from t-private regular k-server PIR to a t'-private b-error-correcting m-server PIR protocol Π.

	Reference	Multiplicative overhead to client-side computation	Num. of servers m	Rounds	Privacy of Π	Remarks
Perfect	[7]	$m^{O(b)}2^{O(k)}$	$2b+k$	1	t	
	Ours (Thm. 2)	$O(bm^2)$	$2b+\widetilde{\Omega}(k)$	1	t	non-explicit
	Ours (Thm. 3)	$O(m)$	$(2b+1)k$	1	t	
Statistical	[16]	$m^{O(b)}$	$\max\{2b+1, b+k\}$	1	t	
	Ours (Thm. 5)	$\widetilde{O}(bm^5)$	$\max\{2b+1, b+\widetilde{\Omega}(k)\}$	2	$t-1$	non-explicit
	Ours (Thm. 6)	$\widetilde{O}(m^5)$	$\max\{2b+1, b+k\}$	$O(m^2)$	t	$b \leq t$

Computational complexity omits a factor of $\log \epsilon^{-1}$ for the probability of failure ϵ. The notations $\widetilde{O}(\cdot)$ and $\widetilde{\Omega}(\cdot)$ hide a logarithmic factor of k or m.

Generic Construction of Perfect Error-Correcting PIR. We propose a generic construction of one-round perfect b-error-correcting m-server PIR protocols from any one-round regular k-server PIR protocol for $m \geq 2b + \Omega(k \log k)$. The construction in [7] incurs a multiplicative overhead $\binom{m}{b} = m^{O(b)}$ to computational complexity while attaining a smaller number of servers $m \geq 2b+k$. Our construction only incurs a multiplicative overhead $O(bm^2)$ to communication and computational complexity. To obtain our result, we introduce a new combinatorial object called a *locally surjective map family*, which we believe to be of independent interest. Roughly speaking, our error-correcting protocol replicates queries generated by a regular PIR protocol using a map from a locally surjective map family (see Section 2.1 for details). We note that the above number of servers is based on a probabilistic construction of locally surjective map families. We also show an explicit construction, but with worse parameters. As a result,

Table 2. Comparison between t-private b-error-correcting m-server PIR protocols with polynomial computational complexity in m.

	Reference	Error correction	Num. of servers m	Communication	Rounds	Remarks
$t = 1$	[34]	perfect	$(2b+1)2^r$	$\exp(\widetilde{O}_r((\log n)^{1/r}))$	1	non-explicit
	Ours (Cor. 1)	perfect	$2b + \Omega(r2^r)$	$\exp(\widetilde{O}_r((\log n)^{1/r}))$	1	non-explicit
	Ours (Cor. 2)	perfect	$(2b+1)2^{r-1}$	$\exp(\widetilde{O}_r((\log n)^{1/r}))$	1	
	Ours (Cor. 4)	statistical	$\max\{2b+1, b+\Omega(r4^r)\}$	$\exp(\widetilde{O}_r((\log n)^{1/r}))$	2	non-explicit
$t \geq 1$	[23]	perfect	$2b + \Omega(td)$	$n^{1/d}$	1	
	[16]	statistical	$\max\{2b+1, b+\Omega(td)\}$	$n^{1/d}\log n$	1	
	Ours (Cor. 5)	statistical	$\max\{2b+1, b+\Omega(td)\}$	$n^{1/d}$	$O(m^2)$	$b \leq t$

Communication complexity omits a polynomial factor in m and $\log \epsilon^{-1}$ for the probability of failure ϵ. The notation $\widetilde{O}_r(\cdot)$ hides a factor of $\log\log n$ and constants that depend on r only. Note that $\exp(\widetilde{O}_r((\log n)^{1/r})) = n^{o(1)}$ if $r \geq 2$.

Table 3. Comparison between t-private (b, L)-list-decodable m-server PIR protocols with polynomial computational complexity in m.

	Reference	List decoding	Num. of servers m	Communication	Rounds	Remarks
$t = 1$	Ours (Cor. 4)	statistical	$\max\left\{\frac{L+1}{L}b+1, b+\Omega(r4^r)\right\}$	$\exp(\widetilde{O}_r((\log n)^{1/r}))$	2	non-explicit
$t \geq 1$	[17]	perfect	$\frac{L+1}{L}b + \Omega(Lt)$	$n^{1/2}$	1	
	[12]	statistical	$\frac{L+1}{L}b + \Omega(t)$	$n^{1/2}$	1	
	Ours (Cor. 3)	statistical	$\max\left\{\frac{L+1}{L}b+1, b+\Omega(td)\right\}$	$n^{1/d}$	2	non-explicit

Communication complexity omits a polynomial factor in m and $\log \epsilon^{-1}$ for the probability of failure ϵ. The notation $\widetilde{O}_r(\cdot)$ hides a factor of $\log\log n$ and constants that depend on r only. Note that $\exp(\widetilde{O}_r((\log n)^{1/r})) = n^{o(1)}$ if $r \geq 2$.

we explicitly construct a b-error-correcting m-server PIR protocol from any regular k-server PIR protocol for $m \geq (2b+1)k$. Instantiating our constructions with the best-known 2^{r-1}-server protocol in [13], we show that there exist a 1-private perfect b-error-correcting m-server PIR protocol with communication complexity $\mathcal{L}_n[r^{-1}, O_r(1)] \cdot bm^2$ for $m \geq 2b + \max\{r2^{r-1}, 15\}$; and an explicit one with communication complexity $\mathcal{L}_n[r^{-1}, O_r(1)] \cdot m$ for $m \geq (2b+1)2^{r-1}$, where $\mathcal{L}_n[s, c] := \exp(c(\log n)^s(\log\log n)^{1-s}) = n^{o(1)}$ if $s < 1$. As a comparison, the protocol in [34] needs $m \geq (2b+1)2^r$ servers to achieve the same communication complexity $\mathcal{L}_n[r^{-1}, O_r(1)] \cdot m$. It is not explicitly given due to the necessity of linear codes with special properties (see also Section 1.2). Ours is thus the first explicit error-correcting protocol achieving both $n^{o(1)}$ communication and polynomial computational complexity in m. Note that it is impossible to realize *perfect* b-error-correcting $(2b+1)$-server PIR with $o(n)$ communication since it implies single-server PIR [16]. Below, we achieve the minimum number of servers $m = 2b+1$ by allowing for *statistical* error correction.

Generic Construction of Statistical List-Decodable PIR. We propose a generic construction of statistical (b, L)-list-decodable m-server PIR protocols from any one-round regular k-server PIR protocol for $m \geq \max\{b(L+1)/L + 1, b + \Omega(k \log k)\}$, where a client outputs L candidates of a correct value. If $L = 1$, it further reduces the number of servers of our perfect error-correcting protocols at the cost of allowing for a negligible probability of failure. Note that we do not relax the privacy requirement to statistical privacy, i.e., it still satisfies perfect privacy. Our construction only incurs a multiplicative overhead $bm^{O(1)}$ to communication and computational complexity while the construction in [16] incurs an $\binom{m}{b}$ overhead. As a drawback, our resulting protocol runs in two rounds and is $(t-1)$-private if an initial protocol is t-private. Nevertheless, we can compensate the loss of a privacy level by combining our construction with the method to increase t-privacy [4]. As a result, our construction instantiated with the 2^r-server protocol in [14] provides a 1-private statistical b-error-correcting m-server PIR protocol with communication complexity $\mathcal{L}_n[r^{-1}, O_r(1)] \cdot \widetilde{O}(bm^3 \log \epsilon^{-1})$ for $m \geq \max\{2b+1, b+3r4^r\}$. When $r = 2$ and $b \geq 68$, we obtain the first b-error-correcting PIR protocol with $n^{o(1)}$ communication, achieving both the minimum number of servers $m = 2b+1$ and polynomial computational complexity in m, at the cost of two rounds of interaction. Similarly, we obtain the first 1-private (b, L)-list-decodable PIR protocol with $n^{o(1)}$ communication. For $t \geq 1$, our construction instantiated with [30] provides a t-private (b, L)-list-decodable m-server PIR protocol with communication complexity $\widetilde{O}(n^{1/d} bm^2 \log \epsilon^{-1})$ for $d \geq 2$ and $m \geq \max\{b(L+1)/L+1, b+\Omega(td)\}$. As a comparison, the previous list-decodable protocols [17,12] cannot achieve $o(\sqrt{n})$ communication.

Generic Construction of Statistical Error-Correcting PIR Preserving a Privacy Threshold. We propose a generic construction of $O(m^2)$-round statistical b-error-correcting m-server PIR protocols from any one-round regular k-server PIR protocol for $m \geq \max\{2b+1, b+k\}$. It has an advantage over

our above construction that it preserves the privacy threshold of the underlying regular PIR protocol. If we instantiate it with [30], we obtain a t-private statistical b-error-correcting m-server PIR protocol with communication complexity $n^{1/d}(\log \epsilon^{-1})m^{5+o(1)}$ for $m \geq \max\{2b+1, b+(td+1)/2\}$. In terms of n, it improves the communication complexity $n^{1/d}(\log n\epsilon^{-1})m^{2+o(1)}$ of [16] with the same number of servers. A drawback is that our resulting protocol assumes $b \leq t$, that is, it cannot correct more errors than the privacy threshold. For that reason, our protocol is especially important to guarantee the privacy of a client and the correctness of an output against an adversary who corrupts some b servers actively and other $t-b$ servers passively.

1.2 Related Work

Error-correcting PIR was also considered in a special setting where the length of each entry of a database is sufficiently large (see [3,31] and references therein). An efficiency measure considered there is *download rate*, which is defined as the asymptotic ratio between the total length of servers' answers to a query and the length of each entry. The error-correcting PIR protocols in [3,31] assume that the length of each entry of a database is at least $(m-2b)^n$. Hence they result in exponentially large communication complexity in n.

Zhang et al. [34] showed that there exists a 1-private perfect b-error-correcting $(2b+1)2^r$-server PIR protocol with $n^{o(1)}$ communication. We note that their construction does not give an explicit protocol. This is because their decoding algorithm works for a suitable linear code but the existence of such a code is only demonstrated by a non-constructive proof and by a brute-force algorithm.

Beimel and Stahl [7] showed a construction of error-correcting PIR from any regular k-server PIR protocol. They constructed a (k,m)-*robust* PIR protocol with $2^{O(k)}m\log m$ multiplicative overhead, in which a client can compute a correct value from any k out of m answers. They then showed that the (k,m)-robust protocol is b-error-correcting if $m \geq 2b+k$. However, a client needs to check the consistency of answers for every potential set of $m-b$ honest servers, which incurs an $\binom{m}{b}$ multiplicative overhead to computational complexity. Eriguchi et al. [16] showed a similar result for statistical b-error-correcting PIR. In their construction, a regular PIR protocol is first transformed into a b-*error-detecting* PIR protocol, in which a client can detect up to b errors[7]. As in [7], a client then executes independent instances of the error-detecting protocol with every potential set of honest servers, which again incurs an $\binom{m}{b}$ multiplicative overhead.

Goldberg [17] proposed a t-private perfect (b,L)-list-decodable PIR protocol with communication complexity $O(\sqrt{n}m)$ for $m \geq b(L+1)/L + \Omega(Lt)$. They chose $L = \Theta(\sqrt{m/t})$ to maximize the number of tolerable errors. The protocol in [12] was based on [17] but their decoding algorithm collects L tuples of servers' answers and decode them simultaneously. It can then decode at most $b \leq (m-t-1)L/(L+1)$ errors and hence $m > b(L+1)/L + \Omega(t)$. Kurosawa [23]

[7] See Appendix E for the formal definition of error-detecting PIR.

showed a t-private perfect b-error-correcting protocol with communication complexity $n^{1/d}$ for $m \geq 2b + \Omega(td)$, omitting a polynomial factor in m. Augot et al. [2] proposed a 1-private b-error-correcting m-server PIR protocol with $O(n^{1/d})$ communication for $m \geq 2b + \Omega(d)$, based on Reed-Muller codes. It is subsumed by the protocol in [23]. The PIR protocols in [33] detect errors made by servers with high probability but cannot correct errors, while supporting a more general kind of queries expressed by low-degree polynomials.

2 Technical Overview

In this section, we provide an overview of our techniques. We give more detailed descriptions and security proofs in the following sections.

2.1 Generic Construction of Perfect Error-Correcting PIR

A high-level idea is to enable a client C to find a pair of servers such that at least one of them is malicious, and remove their answers in a reconstruction phase. Let Π be an underlying k-server PIR protocol and T be the set of m servers. Note that any map $f : T \to [k] := \{1, 2, \ldots, k\}$ defines a partition $T = G_1 \cup \cdots \cup G_k$ of T, where $G_\ell = f^{-1}(\ell)$. As a simple example, we first consider a map f such that $|G_\ell| \geq 2b + 1$ for all $\ell \in [k]$. The basic observation is that C can compute the correct output once C obtains a correct answer to each of k queries in Π. For this goal, C sends the ℓ-th query to every server in G_ℓ and receives $|G_\ell| \geq 2b+1$ answers. Then C can determine the correct answer just by using a majority vote. In this protocol, the number of servers needs to be $m \geq (2b + 1)k$. Even this simple protocol improves the number of servers of the protocol in [34].

We show a refined way to reduce $m = (2b + 1)k$ to $m = 2b + \Omega(k \log k)$. Consider a *family $\mathcal{F}$ of maps* from T to $[k]$. We have $|\mathcal{F}|$ different partitions $(G_{f,1}, \ldots, G_{f,k})_{f \in \mathcal{F}}$ of T, where $(G_{f,1}, \ldots, G_{f,k})$ is a partition defined by $G_{f,\ell} = \{\mathsf{S}_i : f(\mathsf{S}_i) = \ell\}$. Instead of removing all malicious servers at once as in the above example, here C executes instances of Π in parallel with the $|\mathcal{F}|$ partitions of T. Our strategy is that C proceeds in b steps in total to detect and remove at least one new malicious server per step. In each step,

- If for every (f, ℓ), all the remaining servers in $G_{f,\ell}$ return the same answer $\mathsf{ans}_{f,\ell}$, then C computes an output x_f of Π from $(\mathsf{ans}_{f,1}, \ldots, \mathsf{ans}_{f,k})$ for each f and decides the final output by the majority vote over the x_f's;
- Otherwise, i.e., if two remaining servers in some $G_{f,\ell}$ give different answers, then C removes these two servers and proceeds to the next step.

Observe that in the latter case, at least one of the two servers is malicious and hence at least one malicious server is always removed. The requirement for C to succeed is that if he is in the former case, more than half of the x_f's are correct. A sufficient condition is that there remains at least one honest server in each of $G_{f,1}, \ldots, G_{f,k}$ for more than half of the f's. Indeed, for such f's, C receives the correct answer from servers in each of $G_{f,1}, \ldots, G_{f,k}$, or proceeds to the latter

case and remove a malicious server. Since there remains at least $m - 2b$ honest servers at every step, the condition can be formulated as the family $\mathcal{F}$ of maps satisfying that for any subset $H \subseteq T$ of size $m - 2b$, $f(H) = [k]$ holds for more than half of the f's. We name such a family as a *locally surjective map family*.

We can prove by a probabilistic argument the existence of a locally surjective map family $\mathcal{F}$ of size $O(m)$ if $k = O((m - 2b)/\log(m - 2b))$. Therefore, we can obtain a b-error-correcting PIR protocol from any regular k-server one if $m \geq 2b + \Omega(k \log k)$. It incurs a $O(bm|\mathcal{F}|) = O(bm^2)$ multiplicative overhead to communication and computational complexity.

2.2 Generic Construction of Statistical List-Decodable PIR

First, we show a construction of a b-error-correcting m-server PIR protocol assuming $m > 2b$, i.e., the majority of servers are honest. Let Π be an $(m-b, m)$-robust PIR protocol, in which a client C can obtain a correct value from any $m - b$ out of m answers. It is known that if $k \leq (m - b)/\ln(m - b)$, i.e., $m \geq b + \Omega(k \log k)$, an $(m - b, m)$-robust PIR protocol can be constructed from any regular k-server one with $O(m^2)$ multiplicative overhead [7]. We start with a naive way that: In the first round, C receives answers from m servers in Π as usual; In the second round, C asks a special common query (specified below) to all servers. He then finds a set A of at least $m - b$ servers who returns the same reply to the common query, and performs the reconstruction of Π by trusting their $m - b$ answers in the first round. This set A is unique as we assume $m - b > m/2$, and contains all honest servers. Hence, a malicious server is successfully excluded if it behaves maliciously in the second round. However, there is an obvious attack that a malicious server behaves honestly in the second round not to be excluded, but maliciously in the first round to cause errors in the reconstruction.

Our countermeasure is to choose a target server S_ℓ uniformly at random and ask "What is S_ℓ's correct answer to the query in the first round?" to all servers other than S_ℓ. It leaks one additional query in Π and decreases the privacy level from t to $t - 1$. In the reconstruction phase, C forms a set A of all the servers who sent the same reply to the above query, where we fix the reply of S_ℓ to its answer in the first round. Clearly, $\mathsf{S}_\ell \in A$ if S_ℓ is honest. This successfully prevents the above attack if a malicious server is chosen as a target server S_ℓ, but the success probability is only $1/m$. To make it overwhelming, we let C run many (say, κ) independent instances in parallel. C then defines a final set A to be the intersection of the A's in all instances, which is still unique and contains all honest servers. He determines a final output as the majority of outputs in all instances (if it exists). Now, C outputs an incorrect result only if a coalition of b malicious servers cause errors in at least $\kappa/2$ instances. It implies that one of them must behave maliciously without being detected in at least $\kappa/(2b)$ instances. We can show that such a probability of failure is made negligible in a parameter λ if we set $\kappa = \Omega(bm\lambda)$. To summarize, we obtain a two-round $(t-1)$-private statistical b-error-correcting m-server PIR protocol from any one-round regular t-private k-server PIR protocol if $m \geq \max\{2b+1, b+\Omega(k \log k)\}$.

The communication and computational overhead is a multiplicative polynomial factor in m.

Finally, we show how to extend the above construction to the case of list decoding. The only remaining problem is that if the number of servers does not satisfy $m > 2b$, the above set A of at least $m - b$ servers is no longer unique in general. Nevertheless, if $m > b(L+1)/L$ for a list size $L \geq 2$, the number of such sets A is still bounded by L since $(L+1)(m-b) > m$. More concretely, we define that two servers are equivalent if their replies in the second round are the same in all instances, and redefine A as each equivalence class of size at least $m - b$. This makes the above protocol (b, L)-list-decodable.

2.3 Generic Construction of Statistical Error-Correcting PIR Preserving a Privacy Threshold

Our basic idea is similar to our previous method to construct perfect error-correcting PIR: In each execution of a regular k-server PIR protocol, a client C not only detects errors but finds a pair of servers containing malicious ones. However, if C removes such pairs, the number of remaining servers is reduced to $m - 2b$ in the end, which requires the number of servers to be at least $m \geq 2b + k$. To achieve $m = \max\{2b + 1, b + k\}$, we introduce a stronger notion of *b-conflict-finding* PIR. Specifically, it guarantees that as long as there are at most b malicious servers, C obtains a correct result a_τ or a non-trivial partition (G_0, G_1) of the set of servers such that either of G_0 or G_1 contains all honest servers (and hence the other group consists of malicious servers only)[8].

More concretely, we consider a graph $\mathcal{G}$ with m vertices each of which represents a server. Our protocol starts with $\mathcal{G}$ being a complete graph, and repeats the following step: C executes a conflict-finding PIR protocol Π with some set V of servers forming a connected subgraph of size k in $\mathcal{G}$ (which can be efficiently found [28]). If all servers in V are honest (or some of them are malicious but behave honestly in this round), then C obtains the correct output. Otherwise C can find a partition (G_0, G_1) of V thanks to the conflict-finding property of Π. Note that there is always an edge between G_0 and G_1 since $G_0 \cup G_1 = V$ is connected. C removes an arbitrary edge between G_0 and G_1 from $\mathcal{G}$ and goes back to the first step. Since all edges in the set H of honest servers remain unremoved (and hence H remains connected), C successfully chooses a set of $m - b$ honest servers within $O(m^2)$ rounds. Note that in the above construction, C chooses a set of servers with which he interacts, depending on answers that are maliciously computed in the previous rounds. Thus servers may learn some information on a client's index by seeing which servers C removes. To address this problem, we impose an additional property on conflict-finding PIR that the distribution of the partition (G_0, G_1) is independent of a client's index τ regardless of how malicious servers behave. Then, an edge removed in each round leaks no information on τ and hence the t-privacy is preserved.

[8] We say that a partition (G_0, G_1) is non-trivial if neither of G_0 nor G_1 is empty.

Instantiation of Conflict-Finding PIR. The remaining problem is how to construct a b-conflict-finding k-server PIR protocol Π from a regular k-server protocol Π_0. For simplicity suppose that $k = 3$, $b = 1$, and S_1 is malicious. Let λ be a parameter. In Π, C randomly chooses $\mu \in [\lambda]$ and computes λ independent queries $(\mathsf{que}_1^{(h)}, \mathsf{que}_2^{(h)}, \mathsf{que}_3^{(h)})_{h\in[\lambda]}$ using Π_0 on input τ', where $\tau' = \tau$ (his true input) if $h = \mu$, and $\tau' = 1$ otherwise. After he receives λ tuples of answers $(\mathsf{ans}_1^{(h)}, \mathsf{ans}_2^{(h)}, \mathsf{ans}_3^{(h)})_{h\in[\lambda]}$, C broadcasts $(\mathsf{que}_1^{(h)}, \mathsf{que}_2^{(h)}, \mathsf{que}_3^{(h)})_{h\neq\mu}$ to all servers. Each server S_j returns an answer $\mathsf{verify}_{ij}^{(h)}$ to $\mathsf{que}_i^{(h)}$ as S_i would answer to $\mathsf{que}_i^{(h)}$. If S_1 behaves honestly in the first round, it holds that $\mathsf{ans}_1^{(h)} = \mathsf{verify}_{12}^{(h)} = \mathsf{verify}_{13}^{(h)}$ for any $h \neq \mu$. If S_1 returns an incorrect answer to the h-th query for some $h \neq \mu$, it is different from $\mathsf{verify}_{12}^{(h)}$ or $\mathsf{verify}_{13}^{(h)}$. From this observation, C computes and outputs a desired value from $(\mathsf{ans}_1^{(\mu)}, \mathsf{ans}_2^{(\mu)}, \mathsf{ans}_3^{(\mu)})$ if every $v_i := (\mathsf{verify}_{1i}^{(h)}, \mathsf{verify}_{2i}^{(h)}, \mathsf{verify}_{3i}^{(h)})_{h\neq\mu}$ takes the same value. Otherwise, he partitions the set of servers into equivalence classes by placing S_i and S_j into the same class if and only if $v_i = v_j$, and outputs a non-trivial partition (G_0, G_1) in some way. Since $v_2 = v_3$, honest servers $\mathsf{S}_2, \mathsf{S}_3$ are placed in the same class. We note that S_1 successfully submits an incorrect answer without being detected only if it guesses μ correctly. If Π_0 is b-private, it happens with probability at most $O(\lambda^{-1})$. Note that (G_0, G_1) is determined by the answers $(\mathsf{ans}_1^{(h)}, \mathsf{ans}_2^{(h)}, \mathsf{ans}_3^{(h)})_{h\neq\lambda}$, which are independent of τ due to the b-privacy of Π_0 since they can be simulated from one query $\mathsf{que}_1^{(\mu)}$ for τ. This implies that (G_0, G_1) leaks no information on τ. The cheating probability of malicious servers can be made negligible by executing sufficiently many (say, κ) instances in parallel. If a conflict is found in some instance, C outputs the non-trivial partition obtained in that instance. If he obtains valid values in all instances, he outputs the majority of the κ values if it exists. To let this modified protocol fail, malicious servers need to let the basic protocol output valid but incorrect values in at least $\kappa/2$ instances. The failure probability is thus $O(\lambda^{-\kappa/2})$, which is negligible.

To summarize, we obtain an $O(m^2)$-round t-private statistical b-error-correcting m-server PIR protocol from any one-round regular t-private k-server PIR protocol if $m \geq \max\{2b+1, b+k\}$ and $t \geq b$. The communication and computational overhead is a multiplicative polynomial factor in m.

3 Preliminaries

3.1 Notations

For $m \in \mathbb{N}$, define $[m] = \{1, \ldots, m\}$. Let X, Y be sets. If $X \subseteq Y$, we define $Y \setminus X = \{y \in Y : y \notin X\}$ and simply denote it by $\overline{X}$ if Y is clear from the context. We write $u \leftarrow_{\$} X$ if u is chosen uniformly at random from X. Define $\binom{X}{k}$ as the set of all subsets of X of size k. Define $\mathrm{Map}(X, Y)$ as the set of all maps from X to Y. If $X = [m]$ and $Y = [k]$, we simply denote it by $\mathrm{Map}(m, k)$. Let $\mathfrak{R}_{\mathcal{A}}$ denote the set of all random strings for a probabilistic algorithm $\mathcal{A}$. Namely, on input x, $\mathcal{A}$ outputs $\mathcal{A}(x; r)$ for $r \leftarrow_{\$} \mathfrak{R}_{\mathcal{A}}$. For a vector $\boldsymbol{x}$, let x_i denote

the i-th entry. Let $\log x$ denote the base-2 logarithm of x and $\ln x$ denote the base-e logarithm of x, where e is the Napier's constant. Let $\mathcal{L}_n[s,c]$ denote a function $\exp(c(\log n)^s(\log\log n)^{1-s})$ of n, where $0 \le s \le 1$ and $c > 0$. Note that $\mathcal{L}_n[s,c] = n^{o(1)}$ if $s < 1$.

Throughout the paper, we use the following notations:

- m denotes the total number of servers.
- t denotes a privacy threshold: no coalition of t servers learns a client's query.
- b denotes the number of malicious servers.
- $\mathcal{X}^n$ is the universe of databases: n is the number of elements in a database and $\mathcal{X}$ is the set from which each element takes a value.

The notation $\widetilde{O}(\cdot)$ hides a polylogarithmic factor in b and m, and $O_r(\cdot)$ hides any constant depending on a parameter r.

3.2 Private Information Retrieval (PIR)

We first recall the definition of one-round PIR schemes.

Definition 1 (One-round PIR). *A one-round t-private m-server PIR scheme Π consists of three algorithms $\Pi = (\mathcal{Q}, \mathcal{A}, \mathcal{D})$, where:*

- *A query algorithm $\mathcal{Q}$ takes a search index $\tau \in [n]$ as input. It then samples a random string $r \leftarrow_{\$} \mathfrak{R}_{\mathcal{Q}}$ and outputs $\mathsf{que}_i \in \{0,1\}^{c_{\mathsf{que}}}$ for $i \in [m]$ and $\mathsf{aux} \in \{0,1\}^{c_{\mathsf{aux}}}$. That is, $\mathcal{Q}(\tau; r) = (\mathsf{que}_1, \ldots, \mathsf{que}_m; \mathsf{aux})$;*
- *An answer algorithm $\mathcal{A}$ takes $i \in [m]$, $\mathsf{que}_i \in \{0,1\}^{c_{\mathsf{que}}}$ and $\boldsymbol{a} \in \mathcal{X}^n$ as input and outputs $\mathsf{ans}_i \in \{0,1\}^{c_{\mathsf{ans}}}$. That is, $\mathcal{A}(i, \mathsf{que}_i, \boldsymbol{a}) = \mathsf{ans}_i$;*
- *A reconstruction algorithm $\mathcal{D}$ takes $(\mathsf{ans}_1, \ldots, \mathsf{ans}_m) \in (\{0,1\}^{c_{\mathsf{ans}}})^m$ and $\mathsf{aux} \in \{0,1\}^{c_{\mathsf{aux}}}$ as input, and outputs $y \in \mathcal{X}$. That is, $\mathcal{D}(\mathsf{ans}_1, \ldots, \mathsf{ans}_m; \mathsf{aux}) = y$;*

satisfying the following properties:

Correctness. *For any database $\boldsymbol{a} = (a_1, \ldots, a_n) \in \mathcal{X}^n$ and any search index $\tau \in [n]$, it holds that $\Pr[r \leftarrow_{\$} \mathfrak{R}_{\mathcal{Q}} : \mathcal{D}(\mathsf{ans}_1, \ldots, \mathsf{ans}_m; \mathsf{aux}) = a_\tau] = 1$, where $(\mathsf{que}_1, \ldots, \mathsf{que}_m; \mathsf{aux}) = \mathcal{Q}(\tau; r)$ and $\mathsf{ans}_i = \mathcal{A}(i, \mathsf{que}_i, \boldsymbol{a})$ for $i \in [m]$.*

Privacy. *For any $X \in \binom{[m]}{t}$ and any $\tau, \tau' \in [n]$, the distributions of $(\mathsf{que}_i)_{i \in X}$ and $(\mathsf{que}'_i)_{i \in X}$ are perfectly identical, where $r, r' \leftarrow_{\$} \mathfrak{R}_{\mathcal{Q}}$, $(\mathsf{que}_1, \ldots, \mathsf{que}_m; \mathsf{aux}) = \mathcal{Q}(\tau; r)$ and $(\mathsf{que}'_1, \ldots, \mathsf{que}'_m; \mathsf{aux}') = \mathcal{Q}(\tau'; r')$.*

Define the (total) communication complexity $\mathrm{Comm}(\Pi)$ *of Π as $m(c_{\mathsf{que}} + c_{\mathsf{ans}})$. Define the client-side computational complexity* $\text{c-Comp}(\Pi)$ *as the total running time of $\mathcal{Q}$ and $\mathcal{D}$, and the server-side computational complexity* $\text{s-Comp}(\Pi)$ *as the running time of $\mathcal{A}$.*

As described below, a PIR scheme can be identified with a one-round protocol between $\mathsf{C}, \mathsf{S}_1, \ldots, \mathsf{S}_m$, where C is a client who has a search index τ and each S_i is a server who holds a copy of a database $\boldsymbol{a}$:

Query. On input $\tau \in [n]$, C chooses $r \leftarrow_{\$} \mathfrak{R}_{\mathcal{Q}}$ and computes $(\mathsf{que}_1, \ldots, \mathsf{que}_m; \mathsf{aux}) = \mathcal{Q}(\tau; r)$. C then sends que_i to S_i for each $i \in [m]$.
Answer. On input $\boldsymbol{a} \in \{0,1\}^n$, each S_i returns $\mathsf{ans}_i = \mathcal{A}(i, \mathsf{que}_i, \boldsymbol{a})$ to C.
Output. C outputs $y = \mathcal{D}(\mathsf{ans}_1, \ldots, \mathsf{ans}_m; \mathsf{aux})$.

The definition of one-round PIR can be generalized to the multi-round setting in a natural way.

Definition 2 (Multi-round PIR). *A t-private m-server PIR protocol is an interactive protocol Π between a client C holding an index $\tau \in [n]$ and m servers $\mathsf{S}_1, \ldots, \mathsf{S}_m$ holding a database $\boldsymbol{a} = (a_1, \ldots, a_n) \in \mathcal{X}^n$, satisfying the following properties:*

Interaction pattern. *The servers do not communicate with each other;*
Correctness. *If all servers behave honestly, C obtains a_τ at the end of Π with probability 1;*
Privacy. *For any set X of t servers, the joint distribution of all queries that servers in X receive during execution of Π is independent of τ.*

We say that Π is ℓ-round if it requires at most ℓ rounds of queries and answers. Define the communication complexity $\mathrm{Comm}(\Pi)$ *of Π as the total number of bits communicated between C and $\mathsf{S}_1, \ldots, \mathsf{S}_m$. Define the client-side computational complexity* $\text{c-Comp}(\Pi)$ *as the total running time of local computations performed by C and the server-side computational complexity* $\text{s-Comp}(\Pi)$ *as the maximum (over i) of the total running time of local computations performed by S_i.*

Note that Definition 2 with $\ell = 1$ coincides with Definition 1.

3.3 Robust PIR

Next, we introduce the notion of one-round (k, m)-robust PIR [7], which allows a client to obtain a data item from any k out of m answers.

Definition 3 (Robust PIR). *A one-round PIR scheme $\Pi = (\mathcal{Q}, \mathcal{A}, \mathcal{D})$ is said to be (k, m)-robust if*

- *$\mathcal{D}$ takes $X \in \binom{[m]}{k}$, $(\mathsf{ans}_i)_{i \in X} \in (\{0,1\}^{c_{\mathsf{ans}}})^k$ and $\mathsf{aux} \in \{0,1\}^{c_{\mathsf{aux}}}$ as input, and outputs $y \in \mathcal{X}$;*
- *For any database $\boldsymbol{a} \in \mathcal{X}^n$, any $\tau \in [N]$ and any $X \in \binom{[m]}{k}$, it holds that $\Pr[r \leftarrow_{\$} \mathfrak{R}_{\mathcal{Q}} : \mathcal{D}(X, (\mathsf{ans}_i)_{i \in X}; \mathsf{aux}) = a_\tau] = 1$, where $(\mathsf{que}_1, \ldots, \mathsf{que}_m; \mathsf{aux}) = \mathcal{Q}(\tau; r)$ and $\mathsf{ans}_i = \mathcal{A}(i, \mathsf{que}_i, \boldsymbol{a})$ for $i \in [m]$.*

Beimel and Stahl [7] showed constructions of robust PIR schemes from any regular PIR scheme. One of their constructions is based on the following combinatorial object.

Definition 4. *Let $m, h, k \in \mathbb{N}$ and $\mathcal{F}$ be a family of maps from $[m]$ to $[k]$. We call $\mathcal{F}$ an (m, h, k)-nearly perfect hash family if for any $H \in \binom{[m]}{h}$, there exists a map $f \in \mathcal{F}$ such that $f(H) = [k]$.*

There is a probabilistic construction of (m, h, k)-nearly perfect hash families for $k = O(h/\log h)$ [7]. In the original construction in [7], the size of $\mathcal{F}$ is upper bounded by $(h \log m)/(\log \log h) = O((m \log m)/(\log \log h))$. We show a slightly different analysis on $|\mathcal{F}|$ and obtain an upper bound of $|\mathcal{F}| \leq 8m$. If $h = \Theta(m)$, our bound is better than [7]. See Appendix C for a formal proof.

Proposition 1. *Let $m, h, k \in \mathbb{N}$ be such that $m \geq h \geq 3$ and $k \leq h/\ln h$. Then, there exists an (m, h, k)-nearly perfect hash family $\mathcal{F}$ such that $|\mathcal{F}| \leq 8m$.*

Although both of the construction in [7] and the above one are probabilistic and not explicit, we can make the success probability overwhelming in λ at the cost of incurring an $O(\lambda)$ additive overhead to $|\mathcal{F}|$. We formally prove it in Appendix C.

Beimel and Stahl [7] showed that given an (m, h, k)-nearly perfect hash family of size w, an (h, m)-robust PIR scheme can be obtained from any k-server PIR scheme with $O(mw)$ communication overhead. We thus have the following fact.

Proposition 2. *Let $m \geq h \geq 3$ and $k \leq h/\ln h$. Let Π_0 be a one-round t-private k-server PIR protocol. Then, there exists a one-round t-private (h, m)-robust m-server PIR protocol Π such that $\mathrm{Comm}(\Pi) = O(m^2 \cdot \mathrm{Comm}(\Pi_0))$, $\text{c-Comp}(\Pi) = O(m^2 \cdot \text{c-Comp}(\Pi_0))$ and $\text{s-Comp}(\Pi) = O(m \cdot \text{s-Comp}(\Pi_0))$.*

3.4 Existing Constructions of PIR

First, we recall a fundamental combinatorial object.

Definition 5. *Let $\ell \in \mathbb{Z}$ and $S \subseteq \mathbb{Z}_m \setminus \{0\}$. We say that $\mathcal{U} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n)$ and $\mathcal{V} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n)$, where $\boldsymbol{u}_i, \boldsymbol{v}_i \in \mathbb{Z}_\ell^h$, form an S-matching vector code if $\langle \boldsymbol{u}_i, \boldsymbol{v}_i \rangle = 0$ for every $i \in [n]$ and $\langle \boldsymbol{u}_i, \boldsymbol{v}_j \rangle \in S$ for every $i \neq j$.*

There exists an explicit construction of matching vector codes.

Proposition 3 ([18]). *Let $r \geq 2$ and $p_1 < \cdots < p_r$ be the smallest r primes. Set $\ell_r = p_1 \cdots p_r$. For any integer $n > 1$, there exists an S-matching vector code $\mathcal{U} = \mathcal{V} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n)$ over $\mathbb{Z}_\ell^{h_r(n)}$ such that $h_r(n) = \mathcal{L}_n[r^{-1}, c_r]$, where $c_r = O(r \log r)$ is independent of n.*

Next, there are PIR schemes with sub-polynomial communication based on matching vector codes.

Proposition 4 ([13]). *For $r \geq 2$, there exists a one-round 1-private 2^{r-1}-server PIR scheme Π_r^{DG} such that*

- $\mathrm{Comm}(\Pi_r^{\mathrm{DG}}) = O(2^r \ell_r (\log \ell_r)) \cdot h_r(n) \log |\mathcal{X}|$;
- $\text{c-Comp}(\Pi_r^{\mathrm{DG}}) = 2^{O(r)} \cdot h_r(n) \log |\mathcal{X}|$;
- $\text{s-Comp}(\Pi_r^{\mathrm{DG}}) = r^{O(1)} \cdot n h_r(n) \log |\mathcal{X}|$.

Proposition 5 ([14]). *For $r \geq 2$, there exists a one-round 1-private 2^r-server PIR scheme Π_r^{Efr} such that*

– *The query length is* $c^{\mathrm{Efr}}_{\mathsf{que}} = O((\log \ell_r) h_r(n))$ *and the answer length is* $c^{\mathrm{Efr}}_{\mathsf{ans}} = O((\log \ell_r) \log |\mathcal{X}|)$. *In particular,* $\mathrm{Comm}(\Pi^{\mathrm{Efr}}_r) = O(2^r (\log \ell_r) h_r(n) \log |\mathcal{X}|)$;
– $\text{c-Comp}(\Pi^{\mathrm{Efr}}_r) = 2^{O(r)} \cdot h_r(n) \log |\mathcal{X}|$;
– $\text{s-Comp}(\Pi^{\mathrm{Efr}}_r) = r^{O(1)} \cdot n h_r(n) \log |\mathcal{X}|$.

Although the original PIR schemes in [13,14] assume that $\mathcal{X} = \{0,1\}$ or $\mathcal{X}$ is a finite field, it is straightforward to extend them to the case of an arbitrary set $\mathcal{X}$. Indeed, a client and servers run the schemes in parallel in such a way that the client inputs the same index τ while servers input the i-th bit of data items in the i-th instance. This incurs a multiplicative factor of $O(\log |\mathcal{X}|)$. Note that these $O(\log |\mathcal{X}|)$ executions need not be independent. In particular, the query length does not involve a factor of $\log |\mathcal{X}|$.

Let Π be a t-private m-server PIR scheme whose query length is c_{que} and whose answer length is c_{ans}. It is known that Π can be generically transformed into a $2t$-private m^2-server PIR scheme whose query length is $2c_{\mathsf{que}}$ and whose answer length is c^2_{ans} [4]. In particular, by applying this transformation to the 1-private 2^r-server PIR scheme Π^{Efr}_r with $\mathcal{X} = \{0,1\}$, we obtain a 2-private 4^r-server PIR scheme with $\mathcal{X} = \{0,1\}$ whose communication complexity is $O(4^r (\log \ell_r)^2 h_r(n))$. We can extend the resulting scheme to the case of $|\mathcal{X}| > 2$ by allowing for a multiplicative factor of $O(\log |\mathcal{X}|)$. We summarize this fact in the following proposition.

Proposition 6 ([14,4]). *For $r \geq 2$, there exists a one-round 2-private 4^r-server PIR scheme $\widetilde{\Pi}^{\mathrm{Efr}}_r$ such that*

– $\mathrm{Comm}(\widetilde{\Pi}^{\mathrm{Efr}}_r) = O(4^r (\log \ell_r)^2) \cdot h_r(n) \log |\mathcal{X}|$;
– $\text{c-Comp}(\widetilde{\Pi}^{\mathrm{Efr}}_r) = 2^{O(r)} (\log \ell_r) \cdot h_r(n)^2 \log |\mathcal{X}|$;
– $\text{s-Comp}(\widetilde{\Pi}^{\mathrm{Efr}}_r) = 2^{O(r)} (\log \ell_r) \cdot n h_r(n)^2 \log |\mathcal{X}|$.

Finally, we recall a construction of t-private robust PIR schemes.

Proposition 7 ([30]). *Let $d, k \in \mathbb{N}$ be such that $k \leq m$ and $d \leq (2k-1)/t$. There exists a one-round t-private (k,m)-robust PIR scheme $\Pi^{\mathrm{WY}}_{d,k}$ such that*

– $\mathrm{Comm}(\Pi^{\mathrm{WY}}_{d,k}) = O(m \log m) \cdot d n^{1/d} \log |\mathcal{X}|$;
– $\text{c-Comp}(\Pi^{\mathrm{WY}}_{d,k}) = m^{O(1)} \cdot n^{1/d} \log |\mathcal{X}|$;
– $\text{s-Comp}(\Pi^{\mathrm{WY}}_{d,k}) = (\log m)^{O(1)} \cdot n^{1+1/d} \log |\mathcal{X}|$.

3.5 Error-Correcting PIR

Error-correcting PIR enables a client to obtain a correct value even if some servers return incorrect answers. We consider a malicious adversary $\mathcal{B}$ who corrupts at most b servers. Corrupted servers can deviate from a PIR protocol arbitrarily. The following definition is a generalization of those of [7,17] to the multi-round setting.

Definition 6. *A PIR protocol is said to be $(b, L; 1-\epsilon)$-list-decodable if for any $\boldsymbol{a} \in \mathcal{X}^n$, any $\tau \in [n]$ and any adversary $\mathcal{B}$ who corrupts at most b servers, the probability that a client C outputs a list $\mathcal{Y}$ such that $a_\tau \in \mathcal{Y}$ and $|\mathcal{Y}| \leq L$ at the end of the protocol is at least $1-\epsilon$. We call a $(b, 1; 1-\epsilon)$-list-decodable PIR protocol $(b; 1-\epsilon)$-error-correcting.*

We define the t-privacy of a multi-round $(b, L; 1-\epsilon)$-error-correcting PIR protocol as follows: For any set X of t servers and any adversary $\mathcal{B}$ who corrupts at most b servers, the joint distribution of all queries that servers in X receive during the protocol is independent of τ even if the servers corrupted by $\mathcal{B}$ behave maliciously.

If $\epsilon = 0$, error correction or list decoding is said to be *perfect* and otherwise, *statistical*.

List decoding is a trivial task if $|\mathcal{X}| \leq L$ since a client can then achieve the task just by outputting $\mathcal{Y} = \mathcal{X}$. In addition, it can be seen that there exists a $(b, L; 1-\epsilon)$-list-decodable m-server PIR protocol for negligible ϵ if and only if $m > b(L+1)/L$. We include the proof in Appendix A for completeness.

Without loss of generality, we assume that the behavior of malicious servers is deterministic. Indeed, Definition 6 allows malicious servers to modify their answers in an arbitrary way based on messages from a client. Thus, as long as the randomness of malicious servers is independent of the client's private randomness, the probability of failure is at most ϵ even if the behavior of servers is randomized.

4 Generic Construction of Perfect Error-Correcting PIR

4.1 Generic Construction

In this section, we show a construction of one-round $(b; 1)$-error-correcting m-server PIR protocols from regular k-server PIR schemes for $m \geq 2b + \Omega(k \log k)$. The resulting protocols achieve perfect error correction, i.e., $\epsilon = 0$.

We first introduce a notion of locally surjective map families.

Definition 7. *Let $m, h, k \in \mathbb{N}$ and $\mathcal{F}$ be a family of maps from $[m]$ to $[k]$. We call $\mathcal{F}$ an (m, h, k)-locally surjective map family if $|A_H| > |\mathcal{F}|/2$ for any $H \in \binom{[m]}{h}$, where $A_H = \{f \in \mathcal{F} : f(H) = [k]\}$.*

We show a probabilistic construction of an (m, h, k)-locally surjective map family of size $O(m)$ for $k = O(h/\log h)$. The formal proof is given in Appendix B.

Proposition 8. *Let $m, h, k \in \mathbb{N}$ be such that $h \geq 15$, $m \geq 15$ and $k \leq h/(\gamma \ln h)$, where $\gamma := 1 + (\ln 3 - \ln\ln 15)/(\ln 15) < 1.04$. Then, there exists an (m, h, k)-locally surjective map family $\mathcal{F}$ such that $w := |\mathcal{F}| = 14m$.*

Remark 1. Although our construction of locally surjective map families is not explicit, we can make the success probability of our probabilistic construction overwhelming in λ at the cost of incurring an $O(\lambda)$ additive overhead to m. The formal proof is given in Appendix C.

We show that given an $(m, m-2b, k)$-locally surjective map family, a perfect b-error-correcting m-server PIR protocol can be constructed from any regular k-server PIR scheme.

Theorem 1. *Let $h = m - 2b$. Let $k \in \mathbb{N}$ be such that there exists an (m, h, k)-locally surjective map family $\mathcal{F}$ of size w. Let Π be a one-round t-private k-server PIR protocol. Then, there exists a one-round t-private $(b;1)$-error-correcting m-server PIR protocol Π' such that*

- $\mathrm{Comm}(\Pi') = O(bwm \cdot \mathrm{Comm}(\Pi))$;
- $\text{c-Comp}(\Pi') = O(bwm \cdot \text{c-Comp}(\Pi))$;
- $\text{s-Comp}(\Pi') = O(bw \cdot \text{s-Comp}(\Pi))$.

Proof. The PIR protocol Π' is described in Fig. 1. The t-privacy of Π' follows from that of Π. Indeed, the view of any set of t servers consists of $b+1$ tuples of t queries for Π independent of how malicious servers behave. As for the correctness, assume that all servers behave honestly. Consider the output phase of the protocol Π'. At Step 3 in the first iteration (i.e., $\ell = 1$), for all $u \in [w]$, $p \in [k]$ and $\mathsf{S}_i \in G_{u,p}$, it holds that $\widetilde{\mathsf{ans}}_i^{(u,1)} = \mathcal{A}(p, \mathsf{que}_p^{(u,1)}, \boldsymbol{a})$, that is, we have that $\alpha_p^{(u)} = \mathcal{A}(p, \mathsf{que}_p^{(u,1)}, \boldsymbol{a})$. Therefore, the client C obtains $x^{(u)} = a_\tau$ for all $u \in [w]$ and hence goes to Step 3(a)-i and obtains $y = a_\tau$.

We prove that Π' is $(b;1)$-error-correcting. If all malicious servers behave honestly in the ℓ-th instance for some ℓ, then C goes to Step 3(a)-i in the ℓ-th iteration in the output phase, and computes the correct value $y = a_\tau$. Hence, in every instance, at least one malicious server S_i must submit an incorrect answer $\widetilde{\mathsf{ans}}_i^{(u,\ell)} \neq \mathcal{A}(f_u(i), \mathsf{que}_{f_u(i)}^{(u,\ell)}, \boldsymbol{a})$ for some $u \in [w]$ to let the client output an incorrect value. Let H be the set of all honest servers. Let b_ℓ be the number of malicious servers remaining in T' after iterating Step 3 of the output phase ℓ times. Observe that $b_0 = b$.

Consider the first iteration, i.e., $\ell = 1$. We have that $T' = T$ and $|H| \geq m - 2b = h$. We will show that either of the following two events occurs: (1) C goes to Step 3(a)-i and outputs the correct result $y = a_\tau$; (2) C goes to Step 3(b) and removes at least one malicious server from T'. In other words, C never goes to Step 3(a)-ii. Let BAD be the set of maps $f_u \in \mathcal{F}$ such that at least one malicious server submits an incorrect answer in the instance corresponding to f_u. If $|\mathsf{BAD}| \geq w/2$, there exists $f_u \in \mathsf{BAD}$ such that $f_u(H) = [k]$ due to the property of the locally surjective map family $\mathcal{F}$.[9] Let S_i be a malicious server submitting an incorrect answer $\widetilde{\mathsf{ans}}_i^{(u,1)}$. If $\mathsf{S}_i \in G_{u,p}$, an answer submitted by an honest server $\mathsf{S}_j \in G_{u,p}$ conflicts with $\widetilde{\mathsf{ans}}_i^{(u,1)}$. Then, C goes to Step 3(b) in the output phase, i.e., the second case (2) occurs. Since any honest servers $\mathsf{S}_i, \mathsf{S}_j$ with $f_u(i) = f_u(j)$ return the same answer, at least one malicious server is then removed from T'. In particular, we obtain that $b_1 \leq b - 1$. On the other hand, suppose that $|\mathsf{BAD}| < w/2$ and C goes to Step 3(a). Note that $x^{(u)} =$

[9] Here, we identify H with the set of indices of honest servers.

Notations.

- Let $h = m - 2b$ and $k \in \mathbb{N}$ be such that there exists an (m, h, k)-locally surjective map family $\mathcal{F} = \{f_1, \ldots, f_w\}$ of size w.
- Let $\Pi = (\mathcal{Q}, \mathcal{A}, \mathcal{D})$ be a one-round k-server PIR protocol.
- A client C has an index $\tau \in [n]$ and each server S_i has the database $\boldsymbol{a} \in \mathcal{X}^n$.

Protocol.

Query.

1. Let T be the set of m servers. For each $u \in [w]$ and $p \in [k]$, let $G_{u,p} = \{\mathsf{S}_i : f_u(i) = p\}$.
2. For each $u \in [w]$ and $\ell \in [b+1]$, C computes
$$(\mathsf{que}_1^{(u,\ell)}, \ldots, \mathsf{que}_k^{(u,\ell)}; \mathsf{aux}^{(u,\ell)}) \leftarrow \mathcal{Q}(\tau).$$
3. For each $u \in [w]$, $p \in [k]$ and $\ell \in [b+1]$, C sends $\mathsf{que}_p^{(u,\ell)}$ to all servers in $G_{u,p}$.

Answer. Each server S_i does the following:

1. For each $u \in [w]$ and $\ell \in [b+1]$, S_i computes $\mathsf{ans}_i^{(u,\ell)} = \mathcal{A}(p, \mathsf{que}_p^{(u,\ell)}, \boldsymbol{a})$, where $p = f_u(i)$, i.e., $\mathsf{S}_i \in G_{u,p}$.
2. S_i sends $(\mathsf{ans}_i^{(u,\ell)})_{u \in [w], \ell \in [b+1]}$ to C.

Output.

1. Let $\widetilde{\mathsf{ans}}_i^{(u,\ell)}$ be the answer returned by S_i as $\mathsf{ans}_i^{(u,\ell)}$. That is, $\widetilde{\mathsf{ans}}_i^{(u,\ell)} = \mathsf{ans}_i^{(u,\ell)}$ if S_i is honest and otherwise, $\widetilde{\mathsf{ans}}_i^{(u,\ell)}$ may be modified arbitrarily.
2. C sets $\ell \leftarrow 1$ and $T' \leftarrow T$.
3. If $T' = \emptyset$ or $\ell > b + 1$, C outputs a default value $x_0 \in \mathcal{X}$. Otherwise, C does the following:
   (a) If for all $u \in [w]$ and $p \in [k]$, there exists $\alpha_p^{(u)}$ such that
   $$\{\widetilde{\mathsf{ans}}_i^{(u,\ell)} : \mathsf{S}_i \in G_{u,p} \cap T'\} = \{\alpha_p^{(u)}\}, \tag{1}$$
   then C computes $x^{(u)} = \mathcal{D}(\alpha_1^{(u)}, \ldots, \alpha_k^{(u)}; \mathsf{aux}^{(u,\ell)})$.
      i. If there exists $y \in \mathcal{X}$ such that $|\{u \in [w] : x^{(u)} = y\}| > w/2$, then C outputs y.
      ii. Otherwise, C outputs a default value $x_0 \in \mathcal{X}$.
   (b) Otherwise, let (u, p) be such that the condition (1) does not hold, and $(\mathsf{S}_i, \mathsf{S}_j)$ be a pair of servers in $T' \cap G_{u,p}$ such that $\widetilde{\mathsf{ans}}_i^{(u,\ell)} \neq \widetilde{\mathsf{ans}}_j^{(u,\ell)}$. C sets $\ell \leftarrow \ell + 1$ and $T' \leftarrow T' \setminus \{\mathsf{S}_i, \mathsf{S}_j\}$. C repeats Step 3.

Fig. 1. A construction of perfect error-correcting PIR from regular PIR

$\mathcal{D}((\mathcal{A}(p, \mathsf{que}_p^{(u,1)}, \boldsymbol{a}))_{p\in[k]}; \mathsf{aux}^{(u,1)}) = a_\tau$ for any $u \in [w]$ such that $f_u \notin \mathsf{BAD}$. It holds that $|\{u \in [w] : x^{(u)} = a_\tau\}| > w/2$ and C obtains $y = a_\tau$, i.e., the first case (1) occurs.

Consider the case where the second case (2) continues to occur and C proceeds to the $(\ell+1)$-th iteration for $\ell \leq b$. We prove by induction that $b_\ell \leq b - \ell$. We have shown above that $b_1 \leq b - 1$. Assume that it holds that $b_\ell \leq b - \ell$ after the ℓ-th iteration ends. In the $(\ell+1)$-th iteration, we have that $|T'| = m - 2\ell$. We also have that $h' := |H \cap T'| \geq (m-b) - \ell \geq m - 2b = h$. The property of the locally surjective map family $\mathcal{F}$ implies that $|\{f \in \mathcal{F} : f(H \cap T') = [k]\}| > w/2$. On the other hand, as in the first iteration, the adversary must submit incorrect answers $\widetilde{\mathsf{ans}}_i^{(u,\ell+1)}$ for at least $w/2$ u's in order for C not to output the correct value. Then, C goes to Step 3(b) in this iteration again and successfully removes at least one malicious server. We obtain that $b_{\ell+1} \leq b_\ell - 1 \leq b - (\ell+1)$.

It follows from the induction on ℓ that we have $b_\ell = 0$ for some $\ell \leq b$. In the $(\ell+1)$-th iteration, there is no remaining malicious server. Now C goes to Step 3(a)-i and outputs the correct value.

As for the communication complexity of Π', observe that the client needs to send $O(bw)$ queries $\{\mathsf{que}_p^{(u,\ell)} : u \in [w], \ell \in [b+1]\}$ for Π to all servers in $G_{u,p}$. Since $|G_{u,p}| \leq m$, we have that $\mathrm{Comm}(\Pi') = O(bwm \cdot \mathrm{Comm}(\Pi))$. As for the computational complexity, the client needs to generate $O(bw)$ queries of Π and to make $O(m)$ copies of each of them. In the output phase, the client needs to check consistency among $O(wm)$ answers in each of $b+1$ iterations and finally runs the decoding algorithm $\mathcal{D}$. We note that it is possible to find the majority of a sequence $x^{(1)}, \ldots, x^{(w)} \in \mathcal{X}$ in time $O(w \log |\mathcal{X}|)$, e.g., by the Boyer-Moore algorithm [8]. Thus, we have that $\text{c-Comp}(\Pi') = O(bwm \cdot \text{c-Comp}(\Pi))$. It can be easily seen that $\text{s-Comp}(\Pi') = O(bw \cdot \text{s-Comp}(\Pi))$. □

To obtain a concrete construction from Theorem 1, we plug in the (m, h, k)-locally surjective map family in Proposition 8 with $h = m - 2b$.

Theorem 2. *Suppose that $m \geq 2b + 15$. Let*

$$k \leq \frac{m - 2b}{\gamma \ln(m - 2b)},$$

where $1 < \gamma < 1.04$ is the constant in Proposition 8. Let Π be a one-round t-private k-server PIR protocol. Then, there exists a one-round t-private $(b;1)$-error-correcting m-server PIR protocol Π' such that

- $\mathrm{Comm}(\Pi') = O(bm^2 \cdot \mathrm{Comm}(\Pi))$;
- $\text{c-Comp}(\Pi') = O(bm^2 \cdot \text{c-Comp}(\Pi))$;
- $\text{s-Comp}(\Pi') = O(bm \cdot \text{s-Comp}(\Pi))$.

Remark 2. The computational complexity of the construction in Theorem 2 does not take into account that of finding a locally surjective map family $\mathcal{F}$. We note that the choice of $\mathcal{F}$ does not affect the security of a protocol. Hence we can construct it before the protocol starts and the family is reusable any number

of times. Furthermore, even if such offline complexity is taken into account, the overhead in computational complexity is still polynomial in m, as we show a polynomial-time method to obtain $\mathcal{F}$ in Appendix C. A price to pay is that the resulting protocol is statistical since that construction has a negligible but non-zero probability of failure.

We can also plug in the following locally surjective map family: Assume that $m \geq (2b+1)k$ and let $(G_1, \ldots, G_k)$ be a partition of $[m]$ such that $|G_p| \geq 2b+1$ for all $p \in [k]$. Define a map $f : [m] \to [k]$ as $f(i) = p$ if and only if $i \in G_p$. Then, $\mathcal{F} = \{f\}$ is an $(m, m-2b, k)$-locally surjective map family since $\overline{H} \not\supseteq G_p$ for any $H \in \binom{[m]}{m-2b}$ and $p \in [k]$. This construction gives an explicit way to obtain an error-correcting PIR protocol from a regular one. Furthermore, since $|G_p| \geq 2b+1$, each group G_p has a majority of honest servers. This implies that we can determine the correct answer that servers in G_p should return using a majority vote and can remove dishonest servers at one time. Thus, we do not need to execute many instances in parallel while we did so in Theorem 1, which reduces the multiplicative factor in communication complexity.

Theorem 3. *Suppose that $m \geq 2b+1$. Let $k \leq m/(2b+1)$. Let Π be a one-round t-private k-server PIR protocol. Then, there exists a one-round t-private $(b;1)$-error-correcting m-server PIR protocol Π' such that* $\mathrm{Comm}(\Pi') = O(m \cdot \mathrm{Comm}(\Pi))$, $\text{c-Comp}(\Pi') = O(m \cdot \text{c-Comp}(\Pi))$ *and* $\text{s-Comp}(\Pi') = \text{s-Comp}(\Pi)$.

4.2 Instantiations

If $m \geq 2b + r2^{r-1}$ for $r \geq 2$, then $(m-2b)/(\gamma \ln(m-2b)) \geq 2^{r-1}$. We thus obtain the following corollary by instantiating Theorem 2 with the 2^{r-1}-server PIR scheme Π_r^{DG} in Proposition 4.

Corollary 1. *Let $r \geq 2$. Suppose that $m \geq 2b + \max\{r2^{r-1}, 15\}$. Then, there exists a one-round 1-private $(b;1)$-error-correcting m-server PIR protocol Π' such that*

- $\mathrm{Comm}(\Pi') = O_r(bm^2) \cdot h_r(n) \log |\mathcal{X}|$;
- $\text{c-Comp}(\Pi') = O_r(bm^2) \cdot h_r(n) \log |\mathcal{X}|$;
- $\text{s-Comp}(\Pi') = O_r(bm) \cdot nh_r(n) \log |\mathcal{X}|$.

We obtain the following corollary by instantiating Theorem 3 with the 2^{r-1}-server PIR scheme Π_r^{DG} in Proposition 4.

Corollary 2. *Let $r \geq 2$. Suppose that $m \geq (2b+1)2^{r-1}$. Then, there exists a one-round 1-private $(b;1)$-error-correcting m-server PIR protocol Π' such that*

- $\mathrm{Comm}(\Pi') = O_r(m) \cdot h_r(n) \log |\mathcal{X}|$;
- $\text{c-Comp}(\Pi') = O_r(m) \cdot h_r(n) \log |\mathcal{X}|$;
- $\text{s-Comp}(\Pi') = r^{O(1)} \cdot nh_r(n) \log |\mathcal{X}|$.

5 Generic Construction of Statistical List-Decodable PIR

5.1 Generic Construction

In this section, we show a construction of two-round $(b, L; 1-\epsilon)$-list-decodable m-server PIR protocols from regular k-server PIR schemes for $m \geq \max\{b(L+1)/L+1, b+\Omega(k \log k)\}$. The resulting protocol has a non-zero probability of failure, which can be negligible in λ with $O(\lambda)$ multiplicative overhead. Its privacy threshold t is lower by 1 than the underlying regular PIR scheme.

Theorem 4. *Suppose that $m > b(L+1)/L$. Let Π be a one-round $(t+1)$-private $(m-b, m)$-robust PIR scheme. Then, for any $\epsilon > 0$, there exists a two-round t-private $(b, L; 1-\epsilon)$-list-decodable m-server PIR protocol Π' such that*

- $\mathrm{Comm}(\Pi') = O(bm \log(b\epsilon^{-1}) \cdot \mathrm{Comm}(\Pi))$;
- $\text{c-Comp}(\Pi') = O(bm^3 \log(b\epsilon^{-1}) \cdot \text{c-Comp}(\Pi))$;
- $\text{s-Comp}(\Pi') = O(bm \log(b\epsilon^{-1}) \cdot \text{s-Comp}(\Pi))$.

Proof. Let $\kappa \in \mathbb{N}$ be the smallest integer such that $\kappa \geq 2bm \ln(b\epsilon^{-1})$. Consider the protocol Π' described in Fig. 2. Note that the sub-protocol Π_1 is t-private since any t servers obtain at most $t+1$ queries of the $(t+1)$-private scheme Π. Since Π' simply executes Π_1 in parallel, Π' also satisfies t-privacy. As for the correctness, if every server S_i is honest, it holds that

$$\widetilde{\mathsf{ans}}_i^{(u)}(1,i) = \mathcal{A}(i, \mathsf{que}_i^{(u)}, \boldsymbol{a}) \text{ and } \widetilde{\mathsf{ans}}_{\ell^{(u)}}^{(u)}(2,i) = \mathcal{A}(\ell^{(u)}, \mathsf{que}_{\ell^{(u)}}^{(u)}, \boldsymbol{a})$$

for all $u \in [\kappa]$, where $\mathsf{que}_i^{(u)}$ is the query for S_i generated at the u-th instance of Π_1. All servers are then equivalent under the equivalence relation of Step 3 of Π' and hence $q = 1$ and $A_1 = \{\mathsf{S}_1, \ldots, \mathsf{S}_m\}$. A client C has that $x_1^{(u)} = a_\tau$ for all $u \in [\kappa]$ and outputs $\mathcal{Y} = \{a_\tau\}$.

We show that Π' is $(b, L; 1-\epsilon)$-list-decodable. Let H be the set of all $m-b$ honest servers. Any pair of honest servers $\mathsf{S}_i, \mathsf{S}_j \in H$ are equivalent under the equivalence relation defined at Step 3 of Π'. This is because

$$\widetilde{\mathsf{ans}}_{\ell^{(u)}}^{(u)}(2,i) = \widetilde{\mathsf{ans}}_{\ell^{(u)}}^{(u)}(2,j) = \mathcal{A}(\ell^{(u)}, \mathsf{que}_{\ell^{(u)}}^{(u)}, \boldsymbol{a}).$$

for all $u \in [\kappa]$. Therefore, there exists $p \in [q]$ such that $H \subseteq A_p$ at Step 4 of Π', which in particular implies that C never outputs the default value x_0 at this step. In addition, we can see that $q \leq L$. Indeed, if $q \geq L+1$, we would have that $(m-b)(L+1) \leq m$ and hence $m \leq b(L+1)/L$, which contradicts the assumption $m > b(L+1)/L$.

Let F be the event that C outputs a list $\mathcal{Y}$ such that $a_\tau \notin \mathcal{Y}$. Note that the size of $\mathcal{Y}$ is at most L with probability 1 since $q \leq L$. Let $r^{(u)} \in \mathfrak{R}_{\mathcal{Q}}$ be a random string used to generate queries in the u-th instance of Π_1, i.e., $(\mathsf{que}_1^{(u)}, \ldots, \mathsf{que}_m^{(u)}; \mathsf{aux}^{(u)}) = \mathcal{Q}(\tau; r^{(u)})$. For $r^{(1)}, \ldots, r^{(\kappa)} \in \mathfrak{R}_{\mathcal{Q}}$ let $\mathsf{E}(r^{(1)}, \ldots, r^{(\kappa)})$ be the event that C actually chooses these strings $r^{(1)}, \ldots, r^{(\kappa)}$ at the first round

Notations.

- Let $\Pi = (\mathcal{Q}, \mathcal{A}, \mathcal{D})$ be a one-round $(m-b, m)$-robust PIR protocol.
- Let $\kappa \in \mathbb{N}$.
- A client C has an index $\tau \in [n]$ and each server S_i has the database $\boldsymbol{a} \in \mathcal{X}^n$.

Sub-protocol Π_1.

First round.

Query.

1. C computes $(\mathsf{que}_1, \ldots, \mathsf{que}_m; \mathsf{aux}) \leftarrow \mathcal{Q}(\tau)$.
2. C sends que_i to each server S_i.

Answer. Each server S_i sends $\mathsf{ans}_i(1,i) := \mathcal{A}(i, \mathsf{que}_i, \boldsymbol{a})$ to C.

Second round.

Query.

1. C chooses $\ell \leftarrow\!\!\$\, [m]$.
2. C sends que_ℓ to servers S_i for $i \in [m] \setminus \{\ell\}$.

Answer. A server S_i for $i \in [m] \setminus \{\ell\}$ sends $\mathsf{ans}_\ell(2,i) := \mathcal{A}(\ell, \mathsf{que}_\ell, \boldsymbol{a})$ to C.

Output.

1. Let $\widetilde{\mathsf{ans}}_i(1,i)$ and $\widetilde{\mathsf{ans}}_\ell(2,i)$ denote the answers returned by S_i at the first and second rounds, respectively, where we define $\widetilde{\mathsf{ans}}_\ell(2,\ell) := \widetilde{\mathsf{ans}}_\ell(1,\ell)$.
2. Output $(\widetilde{\mathsf{ans}}_i(1,i), \widetilde{\mathsf{ans}}_\ell(2,i))_{i\in[m]}$ and aux.

Protocol Π'.

1. C and $\mathsf{S}_1, \ldots, \mathsf{S}_m$ execute κ independent instances of Π_1 in parallel.
2. For $u \in [\kappa]$, let $\ell^{(u)}$ be the index chosen by C in the u-th instance of Π_1 and let $(\widetilde{\mathsf{ans}}_i^{(u)}(1,i), \widetilde{\mathsf{ans}}_{\ell^{(u)}}^{(u)}(2,i))_{i\in[m]}$ and $\mathsf{aux}^{(u)}$ be the outputs of the u-th instance of Π_1.
3. C partitions the set of servers into equivalence classes under the following equivalence relation:

$$\mathsf{S}_i \sim \mathsf{S}_j \overset{\text{def}}{\iff} \widetilde{\mathsf{ans}}_{\ell^{(u)}}^{(u)}(2,i) = \widetilde{\mathsf{ans}}_{\ell^{(u)}}^{(u)}(2,j)\ (\forall u \in [\kappa]).$$

4. Let $A_1, \ldots, A_q$ be all the equivalence classes of size at least $m-b$. If there does not exist such a class, C outputs a default value $x_0 \in \mathcal{X}$.
5. For each $p \in [q]$, C does the following:
   (a) C computes

$$x_p^{(u)} = \mathcal{D}(A_p, (\widetilde{\mathsf{ans}}_i^{(u)}(1,i))_{i\in A_p}; \mathsf{aux}^{(u)})$$

   for all $u \in [\kappa]$.
   (b) If there exists $y \in \mathcal{X}$ such that $|\{u \in [\kappa] : x_p^{(u)} = y\}| > \kappa/2$, then C sets $y_p := y$. Otherwise, C sets y_p to the default value x_0.
6. C outputs $\mathcal{Y} = \{y_p : p \in [q]\}$.

Fig. 2. A construction of list-decodable PIR from robust PIR

of Π_1. It is sufficient to show that $\Pr\left[\mathsf{F} \;\middle|\; \mathsf{E}(r^{(1)},\ldots,r^{(\kappa)})\right] \leq \epsilon$ since we then obtain that

$$\Pr[\mathsf{F}] = \sum_{r^{(1)},\ldots,r^{(\kappa)}} \Pr\left[\mathsf{E}(r^{(1)},\ldots,r^{(\kappa)})\right] \cdot \Pr\left[\mathsf{F} \;\middle|\; \mathsf{E}(r^{(1)},\ldots,r^{(\kappa)})\right] \leq \epsilon.$$

Fix random strings $r^{(1)},\ldots,r^{(\kappa)} \in \mathfrak{R}_{\mathcal{Q}}$ such that $\Pr\left[\mathsf{F} \;\middle|\; \mathsf{E}(r^{(1)},\ldots,r^{(\kappa)})\right] > 0$. Note that $(\widetilde{\mathsf{ans}}_i^{(u)}(1,i))_{i\in[m]}$ is also fixed since we assume that servers' (possibly malicious) behavior is deterministic. Let $B_0 \subseteq \overline{H}$ be the set of malicious servers S_i such that $\widetilde{\mathsf{ans}}_i^{(u)}(1,i) \neq \mathcal{A}(i,\mathsf{que}_i^{(u)},\boldsymbol{a})$ for some $u \in [\kappa]$. For $i \in [m]$ with $\mathsf{S}_i \in B_0$, let $K_i \subseteq [\kappa]$ be the set of all u's such that $\widetilde{\mathsf{ans}}_i^{(u)}(1,i) \neq \mathcal{A}(i,\mathsf{que}_i^{(u)},\boldsymbol{a})$. The event F happens only if there exists $B_1 \subseteq B_0$ such that servers in B_1 successfully submit incorrect answers along with honest servers and let C output an incorrect result, i.e.,

$$B_1 \cup H \subseteq A_p \text{ and } \left|\bigcup_{i\in B_1} K_i\right| \geq \frac{\kappa}{2}.$$

Since $|B_1| \leq b$, there exists $\mathsf{S}_i \in B_1 \subseteq \overline{H}$ such that $|K_i| \geq \kappa/(2b)$. For this server S_i, it follows from $\mathsf{S}_i \in A_p$ that $\ell^{(u)} \neq i$ for all $u \in K_i$. The probability that it happens is at most

$$\left(1-\frac{1}{m}\right)^{\kappa/(2b)} \leq \exp\left(-\frac{\kappa}{2bm}\right) \leq \frac{\epsilon}{b}.$$

It therefore follows from the union bound that the probability that F happens is at most ϵ.

As for the communication complexity of Π', observe that the communication complexity of the sub-protocol Π_1 is at most twice larger than that of Π. Since the client and servers run $\kappa = O(bm\log(b\epsilon^{-1}))$ instances of Π_1, we obtain that $\mathrm{Comm}(\Pi') = O(bm\log(b\epsilon^{-1})\cdot\mathrm{Comm}(\Pi))$.

As for the computational complexity, first observe that the client needs to generate one query for Π and receive two sets of answers for Π in each instance of Π_1. Hence the client-side computational complexity of Step 1 of Π' is at most $2\kappa\cdot\text{c-Comp}(\Pi)$. Next, the computational complexity of Step 3 is at most $m^2\kappa\cdot\mathrm{Comm}(\Pi) \leq m^2\kappa\cdot\text{c-Comp}(\Pi)$. This is because the client can verify the equivalence between each pair of servers in $O(\mathrm{Comm}(\Pi)\cdot\kappa)$ time. The computational complexity of Step 5 is at most $q\kappa\cdot\text{c-Comp}(\Pi) \leq m\kappa\cdot\text{c-Comp}(\Pi)$. It is possible to find the majority of a sequence $x^{(1)},\ldots,x^{(\kappa)} \in \mathcal{X}$ in time $O(\kappa\log|\mathcal{X}|)$, e.g., by the Boyer-Moore algorithm [8]. Thus we obtain that $\text{c-Comp}(\Pi') = O(bm^3\log(b\epsilon^{-1})\cdot\text{c-Comp}(\Pi))$. Finally, the server-side computational complexity of Π' is clearly at most $\kappa\cdot\text{s-Comp}(\Pi_1) \leq 2\kappa\cdot\text{s-Comp}(\Pi) = O(bm\log(b\epsilon^{-1})\cdot\text{s-Comp}(\Pi))$. □

It is known that an $(m-b,m)$-robust PIR scheme can be constructed from a regular k-server PIR scheme with $\mathsf{poly}(m,k)$ multiplicative overhead if $k = (m-$

$b)/\ln(m-b)$ (see Proposition 2). We therefore obtain our generic construction of list-decodable PIR from regular PIR.

Theorem 5. *Suppose that $m > b(L+1)/L$. Let*

$$k \leq \frac{m-b}{\ln(m-b)}.$$

Let Π_0 be a one-round $(t+1)$-private k-server PIR protocol. Then for any $\epsilon > 0$, there exists a two-round t-private $(b, L; 1-\epsilon)$-list-decodable m-server PIR protocol Π' such that

- $\mathrm{Comm}(\Pi') = O(bm^3 \log(b\epsilon^{-1}) \cdot \mathrm{Comm}(\Pi_0))$;
- $\text{c-Comp}(\Pi') = O(bm^5 \log(b\epsilon^{-1}) \cdot \text{c-Comp}(\Pi_0))$;
- $\text{s-Comp}(\Pi') = O(bm^2 \log(b\epsilon^{-1}) \cdot \text{s-Comp}(\Pi_0))$.

5.2 Instantiations

If $d \leq (2(m-b)-1)/(t+1)$, then there exists a $(t+1)$-private $(m-b, m)$-robust PIR scheme $\Pi^{\mathrm{WY}}_{d,m-b}$ (Proposition 7). We obtain the following corollary by instantiating Theorem 4 with $\Pi^{\mathrm{WY}}_{d,m-b}$.

Corollary 3. *Let $d \geq 2$. Suppose that*

$$m \geq \max\left\{\frac{L+1}{L}b + 1, b + \frac{(t+1)d+1}{2}\right\}.$$

Then for any $\epsilon > 0$, there exists a two-round t-private $(b, L; 1-\epsilon)$-list-decodable m-server PIR protocol Π' such that

- $\mathrm{Comm}(\Pi') = \widetilde{O}(bm^2 \log \epsilon^{-1}) \cdot dn^{1/d} \log|\mathcal{X}|$;
- $\text{c-Comp}(\Pi') = m^{O(1)} \log \epsilon^{-1} \cdot n^{1/d} \log|\mathcal{X}|$;
- $\text{s-Comp}(\Pi') = \widetilde{O}(bm \log \epsilon^{-1}) \cdot n^{1+1/d} \log|\mathcal{X}|$

It can be seen that $(m-b)/\ln(m-b) \geq 4^r$ if $m \geq b + 3r4^r$. We thus obtain the following corollary by instantiating Theorem 5 with the 2-private 4^r-server PIR scheme $\widetilde{\Pi}^{\mathrm{Efr}}_r$ in Proposition 6.

Corollary 4. *Let $r \geq 2$. Suppose that*

$$m \geq \max\left\{\frac{L+1}{L}b + 1, b + 3r4^r\right\}.$$

Then for any $\epsilon > 0$, there exists a two-round 1-private $(b, L; 1-\epsilon)$-list-decodable m-server PIR protocol Π' such that

- $\mathrm{Comm}(\Pi') = \widetilde{O}_r(bm^3 \log \epsilon^{-1}) \cdot h_r(n) \log|\mathcal{X}|$;
- $\text{c-Comp}(\Pi') = \widetilde{O}_r(bm^5 \log \epsilon^{-1}) \cdot h_r(n)^2 \log|\mathcal{X}|$;
- $\text{s-Comp}(\Pi') = \widetilde{O}_r(bm^2 \log \epsilon^{-1}) \cdot nh_r(n)^2 \log|\mathcal{X}|$.

Remark 3. If $r = 2$ and $m \geq b+68$, then $(m-b)/\ln(m-b) \geq 4^r = 16$. Therefore, if $b \geq 68 \cdot L$, Corollary 4 provides a statistical (b, L)-list-decodable m-server PIR protocol with $n^{o(1)}$ communication achieving the minimum number of servers $m > b(L+1)/L$ and polynomial computational complexity in m.

6 Generic Construction of Statistical Error-Correcting PIR Preserving a Privacy Threshold

In this section, we show a generic construction of $O(m^2)$-round $(b; 1-\epsilon)$-error-correcting m-server PIR protocols from regular k-server PIR protocols for $m \geq \max\{2b+1, b+k\}$. It has an advantage over the construction in Section 5 that it does not decrease the privacy threshold of the underlying regular PIR protocols.

We first introduce a notion of conflict-finding PIR. We next construct error-correcting PIR from conflict-finding PIR. Finally we show a generic construction of conflict-finding PIR from regular PIR, and its instantiation.

Graph Theory. To begin with, we recall the standard terminology of graph theory (see [22, Chapter 2] for instance). A (simple and undirected) graph $\mathcal{G}$ is a pair (V, E), where V is a set of vertices and E is a set of edges $(i, j) \in V \times V$. A graph $\mathcal{G}$ is called connected if there is a walk between each pair of vertices. It is a standard result in graph theory that any graph can be decomposed into connected components in linear time $O(|V| + |E|)$ [28]. For $S \subseteq V$, we denote by $\mathcal{G}[S]$ the induced subgraph, i.e., the graph whose vertex set is S and whose edge set consists of the edges in E that have both endpoints in S.

Let $\mathcal{G} = (V, E)$ be a connected graph with at least k vertices. There is a simple way to find a connected subgraph of $\mathcal{G}$ with exactly k vertices in time $O(k(|V| + |E|))$. Indeed, it is clear for $k = 1$. Suppose that we have found a connected subgraph $\mathcal{C} = (V', E')$ with k vertices. One can choose a pair of vertices $i \in V'$ and $j \in V \setminus V'$, and a walk $(i_0 = i, i_1, \ldots, i_\ell = j)$ between them in linear time $O(|V| + |E|)$, e.g., by breadth-first search. Then, $\mathcal{C}' = \mathcal{G}[V' \cup \{i_p\}]$ is a connected subgraph with $k+1$ vertices, where p is the smallest integer such that $i_p \notin V'$. The claim follows from the induction on k.

6.1 Construction of Error-Correcting PIR from Conflict-Finding PIR

Let V be the set of m servers. We consider a variant of an m-server PIR protocol as follows: Instead of just outputting a value $x \in \mathcal{X}$, a client outputs (y, z) such that (1) $y \in \mathcal{X}$ and $z = \star$, (2) $y = \mathsf{conflict}$ and $z = (G_0, G_1)$, which is a partition of V such that $G_0 \neq \emptyset$ and $G_1 \neq \emptyset$, or (3) $y = \perp$ and $z = \mathsf{failure}$. We call the first (resp. second) component of a client's output the y-output (resp. z-output).

Definition 8 (Conflict-finding PIR). *We say that the above variant of an m-server PIR protocol is $(b; 1-\epsilon)$-conflict-finding if for any $\boldsymbol{a} \in \mathcal{X}^n$, any $\tau \in [n]$ and any adversary $\mathcal{B}$ who corrupts at most b servers, the following holds:*

Correctness. *If all servers behave honestly, a client C outputs $(y, z) = (a_\tau, \star)$ with probability 1.*

Soundness. *The probability that C outputs $y \in \mathcal{X} \setminus \{a_\tau\}$ and $z = \star$, or $(y, z) = (\perp, \mathsf{failure})$ is at most ϵ.*

Conflict finding. *Let H be the set of honest servers. We say that a partition (G_0, G_1) of V is bad if $H \not\subseteq G_0$ and $H \not\subseteq G_1$. The client C never outputs a bad partition (G_0, G_1).*

z-Independence. *The distribution of the z-output over $\{\star, \mathsf{failure}\} \cup \{(G_0, G_1) : G_0 \cup G_1 = V, G_0 \neq \emptyset, G_1 \neq \emptyset\}$ is independent of τ.*

We construct multi-round error-correcting PIR from conflict-finding PIR.

Proposition 9. *Suppose that $m > 2b$. Let $\epsilon_1 > 0$ and $k \leq m - b$. Let Π_{CF} be an ℓ-round t-private $(b; 1-\epsilon_1)$-conflict-finding k-server PIR protocol. Then, for any $\epsilon \geq 2\binom{m}{2}\epsilon_1$, there exists an $O(\ell m^2)$-round t-private $(b; 1-\epsilon)$-error-correcting m-server PIR protocol Π such that*

- $\mathrm{Comm}(\Pi) = O(m^2 \cdot \mathrm{Comm}(\Pi_{\mathrm{CF}}))$;
- $\text{c-Comp}(\Pi) = O(m^2 \cdot \text{c-Comp}(\Pi_{\mathrm{CF}}) + m^5)$;
- $\text{s-Comp}(\Pi) = O(m^2 \cdot \text{c-Comp}(\Pi_{\mathrm{CF}}))$.

Notations.

- Let $k \leq m - b$.
- Let Π_{CF} be a conflict-finding k-server PIR protocol.
- A client C has an index $\tau \in [n]$ and each server S_i has the database $\boldsymbol{a} \in \mathcal{X}^n$.

Protocol.

1. C sets $\mathcal{Y} = \emptyset$ and $\mathcal{G} = (V, E)$ as the complete graph such that $V = \{\mathsf{S}_1, \ldots, \mathsf{S}_m\}$.
2. C decomposes $\mathcal{G}$ into connected components $\mathcal{C}_1 = (V_1, E_1), \ldots, \mathcal{C}_q = (V_q, E_q)$.
3. If $|V_p| < k$ for all p, C outputs $\mathcal{Y}$. Otherwise, C chooses $\mathcal{C}_p$ such that $|V_p| \geq k$ uniformly at random, and chooses a connected subgraph $\mathcal{C}'_p = (V'_p, E'_p)$ of $\mathcal{C}_p$ with exactly k vertices (see the remarks before Section 6.1).
4. C executes Π_{CF} with servers in V'_p.
5. Let (y, z) be the output of Π_{CF}.
6. If $\mathcal{Y} \neq \emptyset$, C chooses an edge $e \in E'_p$ uniformly at random and goes to Step 7. Otherwise, C does the following:
   (a) If $z = \star$, C adds $y = x \in \mathcal{X}$ to $\mathcal{Y}$ and chooses an edge $e \in E'_p$ uniformly at random.
   (b) If $z = (G_0, G_1)$, where (G_0, G_1) is a partition of V'_p, C chooses an edge $e = (\mathsf{S}_i, \mathsf{S}_j) \in E'_p$ such that $\mathsf{S}_i \in G_0$ and $\mathsf{S}_j \in G_1$ uniformly at random.
   (c) If $z = \mathsf{failure}$, C chooses an edge $e \in E'_p$ uniformly at random.
7. C sets $\mathcal{G} \leftarrow (V, E \setminus \{e\})$ and goes back to Step 2.

Fig. 3. A construction of error-correcting PIR from conflict-finding PIR

Proof. Consider a PIR protocol Π described in Fig. 3. We first see that the protocol always terminates. Observe that at least one edge of $\mathcal{G}$ is removed

in each iteration and the protocol ends when there is no connected subgraph of size at least k. Therefore Π_{CF} is executed at most $\binom{m}{2}$ times. Hence the communication complexity of Π is at most $O(m^2 \cdot \mathrm{Comm}(\Pi_{\mathrm{CF}}))$. To prove the correctness, we consider the first iteration. If all servers are honest, then $q = 1$ and $\mathcal{C}_1 = \mathcal{G}$ is the complete graph on V. C executes Π_{CF} with some set of k servers and obtains $(a_\tau, \star)$. He then sets $\mathcal{Y} = \{a_\tau\}$ and just loops Steps 2 to 7 without changing $\mathcal{Y}$. Finally, he outputs the correct result $\mathcal{Y} = \{a_\tau\}$.

We next see that the protocol is t-private. Let e_ℓ be the edge removed in the ℓ-th round and z_ℓ be the z-output of Π_{CF} in the ℓ-th round. We see that the joint distribution of $e_1, \ldots, e_\ell$ is independent of τ for any ℓ. To prove it by the induction on ℓ, assume that the claim holds for $e_1, \ldots, e_{\ell-1}$. The set V'_p of servers with which Π_{CF} is executed in the ℓ-th round is determined by $e_1, \ldots, e_{\ell-1}$. Note that z_ℓ depends only on V'_p and is independent of τ due to the z-independence property in Definition 8. Since a client C chooses e_ℓ depending on $e_1, \ldots, e_{\ell-1}$ and z_ℓ, the claim also holds for $e_1, \ldots, e_\ell$. Let A be any set of t servers and $Q_\ell = \{\mathsf{que}_i^{(\ell)}\}_{i \in X_\ell}$ be the queries that servers in A receive from the client C in the ℓ-th round, where X_ℓ is the subset of A participating in Π_{CF} in that round. We prove that the joint distribution of $Q_1, \ldots, Q_\ell$ is independent of τ for any ℓ, which implies the t-privacy of the protocol. Assume that the claim holds for $Q_1, \ldots, Q_{\ell-1}$. The set X_ℓ can be simulated only from $e_1, \ldots, e_{\ell-1}$, which are independent of τ. Since Π_{CF} is t-private and C runs Π_{CF} on independent randomness in each round, Q_ℓ can be simulated only from X_ℓ and hence is independent of τ. The claim follows from the induction on ℓ.

We see that Π is $(b; 1-\epsilon)$-error-correcting. Let H be a set of k honest servers. Let $\mathcal{Y}$ be an output of Π. Assume that $\mathcal{Y} \neq \{a_\tau\}$. There are two possible cases: $\mathcal{Y} = \emptyset$ or $\mathcal{Y} = \{x\}$ for $x \neq a_\tau$. The first case occurs only if $\mathcal{G}$ has no connected components of size at least k just before Π terminates. In particular, the set H of size k must have become a non-connected subset. Therefore an edge between two honest servers $\mathsf{S}_i, \mathsf{S}_j \in H$ must be removed at some iteration. It is not at Step 6(b) that the edge is removed since the conflict finding property in Definition 8 ensures that C never outputs a bad partition (G_0, G_1). Thus it must be at Step 6(c). That is, there is a set T of servers such that the z-output of Π_{CF} executed with servers in T is $z = \mathsf{failure}$. Since T includes at most b malicious servers, this event happens with probability at most ϵ_1 from the soundness property of Π_{CF}. On the other hand, the second case that $\mathcal{Y} = \{x\}$ for $x \neq a_\tau$ happens only if C obtains the incorrect result $x \neq a_\tau$ at Step 6(a). It implies that the y-output of Π_{CF} executed with servers in V'_p is incorrect. Since there are at most b malicious servers in V'_p, this event also happens with probability at most ϵ_1 from the soundness property of Π_{CF}. Since Π_{CF} is executed at most $\binom{m}{2}$ times, the failure probability of Π is at most $2\binom{m}{2}\epsilon_1 \leq \epsilon$.

Finally, we analyze the computational complexity of Π. From the facts in graph theory, in each iteration, the client can decompose $\mathcal{G}$ into connected components in time $O(m^2)$ and can find a connected subgraph with k vertices in time $O(km^2) = O(m^3)$ (see the remarks before Section 6.1). Since the total number of iterations is $\binom{m}{2}$, the client-side computational complexity is at most

$O(m^2 \cdot \text{c-Comp}(\Pi_{\text{CF}}) + m^5)$. It is easy to see that the server-side computational complexity is $O(m^2 \cdot \text{c-Comp}(\Pi_{\text{CF}}))$. □

Remark 4. Proposition 9 can be applied to a $(b;1)$-conflict-finding PIR protocol (i.e., $\epsilon_1 = 0$) and then a resulting PIR protocol satisfies perfect error correction (i.e., $\epsilon = 0$). Although we here use it only to obtain a statistical error-correcting protocol, Proposition 9 might be used to obtain some results in the prefect case if an instantiation of $(b;1)$-conflict-finding PIR is found.

6.2 Construction of Conflict-Finding PIR from Regular PIR

First, we construct a basic two-round PIR protocol that is $(b;1-\epsilon_0)$-conflict-finding for non-negligible ϵ_0.

Proposition 10. *Suppose that $t \geq b$. Let $\lambda \in \mathbb{N}$ and $\epsilon_0 = m/\lambda$. Let Π be a one-round t-private m-server PIR protocol. Then, there exists a two-round t-private $(b;1-\epsilon_0)$-conflict-finding m-server PIR protocol Π_0 such that*

- $\text{Comm}(\Pi_0) = O(m\lambda \cdot \text{Comm}(\Pi))$;
- $\text{c-Comp}(\Pi_0) = O(m^2\lambda \cdot \text{c-Comp}(\Pi))$;
- $\text{s-Comp}(\Pi_0) = O(m\lambda \cdot \text{s-Comp}(\Pi))$.

Proof. Consider a PIR protocol Π_0 described in Fig. 4. The communication complexity in the first round is the same as that of Π. In the second round, the client sends at most $m\lambda$ queries for Π to every server. Thus, the communication complexity of Π_0 is $O(m\lambda \cdot \text{Comm}(\Pi))$. The t-privacy follows from that of Π since any t servers only see at most t queries regarding a client's index τ. It can be seen that the correctness holds as follows. If all servers are honest, $\widetilde{\mathsf{ans}}_i^{(h)}(2,i) = \widetilde{\mathsf{ans}}_i^{(h)}(1,i)$ for all $i \in [m]$ and $h \in [\lambda] \setminus \{\mu\}$, and $\widetilde{\mathsf{ans}}(2,i) = \widetilde{\mathsf{ans}}(2,j)$ for all $i,j \in [m]$ at the output stage. Then the client C outputs $\mathcal{D}((\widetilde{\mathsf{ans}}_i^{(\mu)}(1,i))_{i\in[m]}; \mathsf{aux}^{(\mu)}) = \mathcal{D}((\mathcal{A}(i, \mathsf{que}_i^{(\mu)}, \boldsymbol{a}))_{i\in[m]}; \mathsf{aux}^{(\mu)}) = a_\tau$ since $((\mathsf{que}_i^{(\mu)})_{i\in[m]}; \mathsf{aux}^{(\mu)}) \leftarrow \mathcal{Q}(\tau)$.

We see that Π_0 is $(b;1-\epsilon_0)$-conflict-finding. Let B be the set of b corrupted servers. Assume that the y-output of Π_0 is $y = \mathsf{conflict}$ and the z-output is a partition of the set of m servers. It implies that $\widetilde{\mathsf{ans}}_i^{(h)}(1,i) \neq \widetilde{\mathsf{ans}}_i^{(h)}(2,i)$ for some $i \in [m]$ and $h \in [\lambda]\setminus\{\mu\}$ at Step 2 in the output stage, or that $q \geq 1$ at Step 3. In the former case, S_i is clearly malicious and $H := \overline{B} \subseteq \{\mathsf{S}_j : j \neq i\}$. In the latter case, it holds that $H = \overline{B} \subseteq G'_p$ for some $0 \leq p \leq q$ since $\widetilde{\mathsf{ans}}(2,i) = \widetilde{\mathsf{ans}}(2,j)$ for all $\mathsf{S}_i, \mathsf{S}_j \in H$. Thus the conflict-finding property in Definition 8 holds.

As for the soundness, it clearly follows from the description of Π_0 that a client never outputs $(y,z) = (\perp, \mathsf{failure})$. Let F be the event that Π_0 outputs $y \in \mathcal{X} \setminus \{a_\tau\}$ and $z = \star$. Also, for $i \in [m]$, let F_i be the event that $y \in \mathcal{X}$, $z = \star$ and $\widetilde{\mathsf{ans}}_i^{(\mu)}(1,i) \neq \mathcal{A}(i, \mathsf{que}_i^{(\mu)}, \boldsymbol{a})$. From the union bound, we have that $\Pr[\mathsf{F}] \leq \sum_{\mathsf{S}_i \in B} \Pr[\mathsf{F}_i]$. We show that $\Pr[\mathsf{F}_i] \leq 1/\lambda$ for all i in the following. If F_i occurs, it must hold that $\widetilde{\mathsf{ans}}_i^{(h)}(1,i) = \widetilde{\mathsf{ans}}_i^{(h)}(2,i) = \widetilde{\mathsf{ans}}_i^{(h)}(2,j) = \mathcal{A}(i, \mathsf{que}_i^{(h)}, \boldsymbol{a})$ for all $h \in [\lambda]\setminus\{\mu\}$ and $j \in [m]$ such that S_j is an honest server. Define a random

Notations.

- Let $\Pi = (\mathcal{Q}, \mathcal{A}, \mathcal{D})$ be a one-round m-server PIR protocol.
- Let $\lambda \in \mathbb{N}$.
- A client C has an index $\tau \in [n]$ and each server S_i has the database $\boldsymbol{a} \in \mathcal{X}^n$.

First round.

Query.

1. C chooses $\mu \leftarrow\!\!\$\, [\lambda]$ and computes

$$(\mathsf{que}_1^{(h)}, \ldots, \mathsf{que}_m^{(h)}; \mathsf{aux}^{(h)}) \leftarrow \begin{cases} \mathcal{Q}(\tau), & \text{if } h = \mu, \\ \mathcal{Q}(1), & \text{otherwise,} \end{cases}$$

   for all $h \in [\lambda]$.
2. C sends $(\mathsf{que}_i^{(h)})_{h\in[\lambda]}$ to each server S_i.

Answer.

1. Each server S_i computes $\mathsf{ans}_i^{(h)}(1,i) = \mathcal{A}(i, \mathsf{que}_i^{(h)}, \boldsymbol{a})$ for all $h \in [\lambda]$.
2. S_i sends $(\mathsf{ans}_i^{(h)}(1,i))_{h\in[\lambda]}$ to C.

Second round.

Query. C sends $(\mathsf{que}_1^{(h)}, \ldots, \mathsf{que}_m^{(h)})_{h\in[\lambda]\setminus\{\mu\}}$ to all servers.

Answer.

1. Each server S_i computes $\mathsf{ans}_k^{(h)}(2,i) = \mathcal{A}(k, \mathsf{que}_k^{(h)}, \boldsymbol{a})$ for all $k \in [m]$ and $h \in [\lambda] \setminus \{\mu\}$.
2. S_i sends $(\mathsf{ans}_1^{(h)}(2,i), \ldots, \mathsf{ans}_m^{(h)}(2,i))_{h\in[\lambda]\setminus\{\mu\}}$ to C.

Output.

1. Let $(\widetilde{\mathsf{ans}}_i^{(h)}(1,i))_{h\in[\lambda]}$ and $\widetilde{\mathsf{ans}}(2,i) = (\widetilde{\mathsf{ans}}_1^{(h)}(2,i), \ldots, \widetilde{\mathsf{ans}}_m^{(h)}(2,i))_{h\in[\lambda]\setminus\{\mu\}}$ denote the answers returned by S_i at the first and second rounds, respectively.
2. If $\widetilde{\mathsf{ans}}_i^{(h)}(1,i) \neq \widetilde{\mathsf{ans}}_i^{(h)}(2,i)$ for some $i \in [m]$ and $h \in [\lambda] \setminus \{\mu\}$, then C chooses such i uniformly at random. C outputs $y = \mathsf{conflict}$ and $z = (\{\mathsf{S}_i\}, \{\mathsf{S}_j : j \neq i\})$.
3. C partitions the set of servers into equivalence classes under the following equivalence relation:

$$\mathsf{S}_i \sim \mathsf{S}_j \overset{\text{def}}{\iff} \widetilde{\mathsf{ans}}(2,i) = \widetilde{\mathsf{ans}}(2,j).$$

   C shuffles the equivalence classes uniformly at random and labels them $G'_0, G'_1, \ldots, G'_q$.
4. If all servers are equivalent, i.e., $q = 0$, C outputs

$$y = \mathcal{D}(\widetilde{\mathsf{ans}}_1^{(\mu)}(1,1), \ldots, \widetilde{\mathsf{ans}}_m^{(\mu)}(1,m); \mathsf{aux}^{(\mu)})$$

   and $z = \star$. Otherwise, C outputs $y = \mathsf{conflict}$ and $z = (G_0, G_1)$, where $G_0 = G'_0$ and $G_1 = G'_1 \cup \cdots \cup G'_q$.

Fig. 4. A basic construction of conflict-finding PIR from regular PIR

variable Z_i as follows: Z_i takes a value $\mu' \in [\lambda]$ if $\widetilde{\mathsf{ans}}_i^{(\mu')}(1,i) \neq \mathcal{A}(i, \mathsf{que}_i^{(\mu')}, \boldsymbol{a})$ and $\widetilde{\mathsf{ans}}_i^{(h)}(1,i) = \mathcal{A}(i, \mathsf{que}_i^{(h)}, \boldsymbol{a})$ for all $h \neq [\lambda] \setminus \{\mu'\}$, and $Z_i = 0$ otherwise. Let U be a random variable representing $\mu \leftarrow_{\$} [\lambda]$ chosen by the client. Since Π is t-private and $b \leq t$, the view $(\mathsf{que}_{i'}^{(h)})_{h\in[\lambda],\mathsf{S}_{i'}\in B}$ of corrupted servers at the first round is independent of U. In particular, $(\widetilde{\mathsf{ans}}_i^{(h)}(1,i))_{h\in[\lambda]}$ (and hence Z_i) is independent of U. Therefore we have that $\Pr[\mathsf{F}_i] \leq \Pr[Z_i = U] \leq 1/\lambda$, which implies that $\Pr[\mathsf{F}] \leq m/\lambda = \epsilon_0$ and that the soundness holds.

To see the z-independence property in Definition 8, observe that the z-output of Π_0 is determined by $(\widetilde{\mathsf{ans}}_i^{(h)}(1,i))_{i\in[m],h\in[\lambda]\setminus\{\mu\}}$ and $(\widetilde{\mathsf{ans}}(2,i))_{i\in[m]}$. For $\mathsf{S}_i \in B$, $(\widetilde{\mathsf{ans}}_i^{(h)}(1,i))_{h\in[\lambda]\setminus\{\mu\}}$ and $\widetilde{\mathsf{ans}}(2,i)$ are determined by $(\mathsf{que}_{i'}^{(\mu)})_{i'\in B}$ and other queries for the index $1 \in [n]$, which are independent of τ due to the t-privacy of Π. Furthermore, for $\mathsf{S}_i \notin B$, $(\widetilde{\mathsf{ans}}_i^{(h)}(1,i))_{h\in[\lambda]\setminus\{\mu\}}$ and $\widetilde{\mathsf{ans}}(2,i)$ are determined by queries for the index 1. We conclude that the z-output is independent of τ.

Finally, we analyze the computational complexity of Π. In the first and second rounds, the client needs to compute λ queries for Π and then the computational complexity is at most $O(\lambda \cdot \text{c-Comp}(\Pi))$. The computational complexity of Step 3 in the output phase is at most $m^2\lambda \cdot \text{Comm}(\Pi) \leq m^2\lambda \cdot \text{c-Comp}(\Pi)$ since the client can verify the equivalence between each pair of servers in $O(\lambda \cdot \text{Comm}(\Pi))$ time. The computational complexity of Step 4 is at most $O(\text{c-Comp}(\Pi))$. Thus, we have that $\text{c-Comp}(\Pi_0) = O(m^2\lambda \cdot \text{c-Comp}(\Pi))$. Every server computes answers to at most $O(m\lambda)$ queries for Π and hence $\text{s-Comp}(\Pi_0) = O(m\lambda \cdot \text{s-Comp}(\Pi))$. □

Next, we construct a two-round PIR protocol that is $(b; 1-\epsilon)$-conflict-finding for negligible ϵ. We defer the proof to Appendix D.

Proposition 11. *Suppose that $t \geq b$. Let Π be a one-round t-private m-server PIR protocol. Then, for any $\epsilon_1 > 0$, there exists a two-round t-private $(b; 1-\epsilon_1)$-conflict-finding m-server PIR protocol Π_1 such that*

- $\text{Comm}(\Pi_1) = O(m^2(\log \epsilon_1^{-1}) \cdot \text{Comm}(\Pi))$;
- $\text{c-Comp}(\Pi_1) = O(m^3(\log \epsilon_1^{-1}) \cdot \text{c-Comp}(\Pi))$;
- $\text{s-Comp}(\Pi_1) = O(m^2(\log \epsilon_1^{-1}) \cdot \text{s-Comp}(\Pi))$.

Finally, by combining Propositions 9 and 11, we obtain our generic construction of $O(m^2)$-round statistical b-error-correcting m-server PIR from one-round regular k-server PIR for $m \geq \max\{2b+1, b+k\}$.

Theorem 6. *Suppose that $m > 2b$ and $t \geq b$. Let $k \leq m - b$. Let Π be a one-round t-private k-server PIR protocol. Then, for any $\epsilon > 0$, there exists an $O(m^2)$-round t-private $(b; 1-\epsilon)$-error-correcting m-server PIR protocol Π_1 such that*

- $\text{Comm}(\Pi_1) = O(m^4 \log(m\epsilon^{-1}) \cdot \text{Comm}(\Pi))$;
- $\text{c-Comp}(\Pi_1) = O(m^5 \log(m\epsilon^{-1}) \cdot \text{c-Comp}(\Pi))$;
- $\text{s-Comp}(\Pi_1) = O(m^4 \log(m\epsilon^{-1}) \cdot \text{s-Comp}(\Pi))$.

6.3 Instantiation

We obtain the following corollary by instantiating Theorem 6 with the scheme $\Pi^{\mathrm{WY}}_{d,m-b}$ in Proposition 7.

Corollary 5. *Let $d \geq 2$. Suppose that $t \geq b$ and $m \geq \max\{2b+1, b+(td+1)/2\}$. Then, for $\epsilon > 0$, there exists an $O(m^2)$-round t-private $(b; 1-\epsilon)$-error-correcting m-server PIR protocol Π_1 such that*

- $\mathrm{Comm}(\Pi_1) = \widetilde{O}(m^5 \log \epsilon^{-1}) \cdot dn^{1/d} \log |\mathcal{X}|$;
- $\text{c-Comp}(\Pi_1) = m^{O(1)} \log \epsilon^{-1} \cdot n^{1/d} \log |\mathcal{X}|$;
- $\text{s-Comp}(\Pi_1) = \widetilde{O}(m^4 \log \epsilon^{-1}) \cdot n^{1+1/d} \log |\mathcal{X}|$.

Remark 5. Corollary 3 also provides a t-private $(b; 1-\epsilon)$-error-correcting m-server PIR protocol. The advantage of Corollary 5 is that it achieves a smaller number of servers $\max\{2b+1, b+(td+1)/2\}$ while Corollary 3 assumes $\max\{2b+1, b+((t+1)d+1)/2\}$ servers.

Acknowledgements

This research was partially supported by JSPS KAKENHI Grant Numbers JP20J20797 and JP19H01109, Japan, JST CREST Grant Numbers JPMJCR2113 and JPMJCR22M1, Japan, and JST AIP Acceleration Research JPMJCR22U5, Japan.

References

1. Ambainis, A.: Upper bound on the communication complexity of private information retrieval. In: Automata, Languages and Programming. pp. 401–407 (1997)
2. Augot, D., Levy-dit Vehel, F., Shikfa, A.: A storage-efficient and robust private information retrieval scheme allowing few servers. In: Cryptology and Network Security. pp. 222–239 (2014)
3. Banawan, K., Ulukus, S.: The capacity of private information retrieval from byzantine and colluding databases. IEEE Transactions on Information Theory **65**(2), 1206–1219 (2019)
4. Barkol, O., Ishai, Y., Weinreb, E.: On locally decodable codes, self-correctable codes, and t-private PIR. Algorithmica **58**(4), 831–859 (2010)
5. Beimel, A., Ishai, Y., Kushilevitz, E., Raymond, J.F.: 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. pp. 261–270 (2002)
6. Beimel, A., Ishai, Y.: Information-theoretic private information retrieval: A unified construction. In: Automata, Languages and Programming. pp. 912–926 (2001)
7. Beimel, A., Stahl, Y.: Robust information-theoretic private information retrieval. Journal of Cryptology **20**(3), 295–321 (2007)
8. Boyer, R.S., Moore, J.S.: MJRTY—a fast majority vote algorithm. Automated Reasoning: Essays in Honor of Woody Bledsoe pp. 105–117 (1991)

9. Chee, Y.M., Feng, T., Ling, S., Wang, H., Zhang, L.F.: Query-efficient locally decodable codes of subexponential length. computational complexity **22**(1), 159–189 (2013)
10. Chor, B., Goldreich, O., Kushilevitz, E., Sudan, M.: Private information retrieval. Journal of the ACM **45**(6), 965–982 (1998)
11. Corrigan-Gibbs, H., Henzinger, A., Kogan, D.: Single-server private information retrieval with sublinear amortized time. In: Advances in Cryptology – EUROCRYPT 2022. pp. 3–33 (2022)
12. Devet, C., Goldberg, I., Heninger, N.: Optimally robust private information retrieval. In: 21st USENIX Security Symposium (USENIX Security 12). pp. 269–283 (2012)
13. Dvir, Z., Gopi, S.: 2-server PIR with subpolynomial communication. Journal of the ACM **63**(4), 1–15 (2016)
14. Efremenko, K.: 3-query locally decodable codes of subexponential length. SIAM Journal on Computing **41**(6), 1694–1703 (2012)
15. Eriguchi, R., Kurosawa, K., Nuida, K.: Multi-server PIR with full error detection and limited error correction. In: 3rd Conference on Information-Theoretic Cryptography (ITC 2022). pp. 1:1–1:20 (2022)
16. Eriguchi, R., Kurosawa, K., Nuida, K.: On the optimal communication complexity of error-correcting multi-server PIR. In: Theory of Cryptography. pp. 60–88 (2022)
17. Goldberg, I.: Improving the robustness of private information retrieval. In: 2007 IEEE Symposium on Security and Privacy (SP'07). pp. 131–148 (2007)
18. Grolmusz, V.: Superpolynomial size set-systems with restricted intersections mod 6 and explicit ramsey graphs. Combinatorica **20**(1), 71–86 (2000)
19. Gupta, T., Crooks, N., Mulhern, W., Setty, S., Alvisi, L., Walfish, M.: Scalable and private media consumption with popcorn. In: 13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16). pp. 91–107 (2016)
20. Itoh, T., Suzuki, Y.: Improved constructions for query-efficient locally decodable codes of subexponential length. IEICE Transactions on Information and Systems **E93.D**(2), 263–270 (2010)
21. Kogan, D., Corrigan-Gibbs, H.: Private blocklist lookups with checklist. In: USENIX Security Symposium. pp. 875–892 (2021)
22. Korte, B.H., Vygen, J.: Combinatorial optimization, vol. 1. Springer (2011)
23. Kurosawa, K.: How to correct errors in multi-server PIR. In: Advances in Cryptology – ASIACRYPT 2019. pp. 564–574 (2019)
24. Kushilevitz, E., Ostrovsky, R.: Replication is not needed: single database, computationally-private information retrieval. In: Proceedings 38th Annual Symposium on Foundations of Computer Science. pp. 364–373 (1997)
25. Mittal, P., Olumofin, F., Troncoso, C., Borisov, N., Goldberg, I.: Pir-tor: Scalable anonymous communication using private information retrieval. In: USENIX Security Symposium. vol. 31 (2011)
26. Mughees, M.H., Chen, H., Ren, L.: OnionPIR: Response efficient single-server pir. In: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. pp. 2292–2306. CCS '21 (2021)
27. Sassaman, L., Cohen, B., Mathewson, N.: The pynchon gate: A secure method of pseudonymous mail retrieval. In: Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society. pp. 1–9. WPES '05 (2005)
28. Sharir, M.: A strong-connectivity algorithm and its applications in data flow analysis. Computers & Mathematics with Applications **7**(1), 67–72 (1981)

29. Thomas, K., Pullman, J., Yeo, K., Raghunathan, A., Kelley, P.G., Invernizzi, L., Benko, B., Pietraszek, T., Patel, S., Boneh, D., et al.: Protecting accounts from credential stuffing with password breach alerting. In: USENIX Security Symposium. pp. 1556–1571 (2019)
30. Woodruff, D., Yekhanin, S.: A geometric approach to information-theoretic private information retrieval. SIAM Journal on Computing **37**(4), 1046–1056 (2007)
31. Yao, X., Liu, N., Kang, W.: The capacity of multi-round private information retrieval from byzantine databases. In: 2019 IEEE International Symposium on Information Theory (ISIT). pp. 2124–2128 (2019)
32. Yekhanin, S.: Towards 3-query locally decodable codes of subexponential length. Journal of the ACM (JACM) **55**(1), 1–16 (2008)
33. Zhang, L.F., Wang, H.: Multi-server verifiable computation of low-degree polynomials. In: 2022 IEEE Symposium on Security and Privacy (SP). pp. 596–613 (2022)
34. Zhang, L.F., Wang, H., Wang, L.P.: Byzantine-robust private information retrieval with low communication and efficient decoding. In: Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security. pp. 1079–1085. ASIA CCS '22 (2022)

A (Im)possibility for List-Decodable PIR

We show that there exists a $(b, L; 1-\epsilon)$-list-decodable m-server PIR protocol for negligible ϵ only if $m > b(L+1)/L$.

Proposition 12. *Let Π be a (possibly multi-round) $(b, L; 1-\epsilon)$-list-decodable m-server PIR protocol for a universe of databases $\mathcal{X}^n$. If $m \leq b(L+1)/L$ and $|\mathcal{X}| \geq L+1$, then $\epsilon \geq 1/(L+2)$.*

Proof. Since $(L+1)(m-b) \leq m$, there exists a partition $(G_1, \ldots, G_{L+1})$ of the set of servers such that $|G_j| \geq m-b$ for all $j \in [L+1]$. Let $x^{(1)}, \ldots, x^{(L+1)} \in \mathcal{X}$ be distinct elements. Let $\boldsymbol{a}^{(1)}, \ldots, \boldsymbol{a}^{(L+1)} \in \mathcal{X}^n$ be $L+1$ databases such that the first element of $\boldsymbol{a}^{(j)}$ is $x^{(j)}$, i.e., $a_1^{(j)} = x^{(j)}$ for all $j \in [L+1]$.

Consider an experiment in which the client has an input $1 \in [n]$ and for any $j \in [L+1]$, servers in G_j follow the rules of Π with input $\boldsymbol{a}^{(j)}$. Let E_j be an event that a list $\mathcal{Y}$ outputted by the client contains $x^{(j)}$ in the experiment. For any $j \in [L+1]$, the experiment can be viewed as an attack where the correct database is $\boldsymbol{a}^{(j)}$, servers in G_j are honest, and servers in G_k for $k \neq j$ behave honestly except that they compute their answers based on $\boldsymbol{a}^{(k)}$ instead of $\boldsymbol{a}^{(j)}$. Since $|\overline{G_j}| \leq b$ and Π is $(b, L; 1-\epsilon)$-list-decodable, the probability that E_j occurs in the experiment is at least $1-\epsilon$. Thus, the probability that all E_j's simultaneously occur is at least $1-(L+1)\epsilon$. On the other hand, the probability that the size of $\mathcal{Y}$ exceeds L is at most ϵ. Since $|\mathcal{Y}| \geq L+1$ if all E_j's simultaneously occur, we obtain that $1-(L+1)\epsilon \leq \epsilon$ and hence $\epsilon \geq 1/(L+2)$. □

Since list decoding is a trivial task if $|\mathcal{X}| \leq L$, it is reasonable to assume that $|\mathcal{X}| \geq L+1$.

Conversely, if $m > b(L+1)/L$, it is possible to realize (b, L)-list-decodable PIR with communication complexity $O(nm \log |\mathcal{X}|)$: Each server sends the entire

database to a client. Given m answers $\boldsymbol{a}^{(1)}, \ldots, \boldsymbol{a}^{(m)}$, the client outputs a list of L values appearing most frequently among the m values $a_\tau^{(1)}, \ldots, a_\tau^{(m)}$.

B Proof of Proposition 8

Fix $H \in \binom{[m]}{h}$. We first show that $\Pr[f \leftarrow_{\$} \mathrm{Map}(m,k) : f(H) \neq [k]] < 1/3$. Indeed, if $f(H) \neq [k]$, there is $j \in [k]$ such that $f(s) \neq j$ for any $s \in H$. Since the total number of maps f such that $f(s) \neq j$ for any $s \in H$ is at most $k^{m-h}(k-1)^h$, we obtain that

$$\Pr[f \leftarrow_{\$} \mathrm{Map}(m,k) : f(H) \neq [k]] \leq \frac{k^{m-h}(k-1)^h k}{k^m} \leq k\left(1 - \frac{1}{k}\right)^h.$$

Since $1 - x \leq \exp(-x)$ and $k \leq h/(\gamma \ln h)$, this probability is further upper bounded by

$$k \cdot \exp\left(-\frac{h}{k}\right) \leq \frac{h}{\gamma \ln h} \exp(-\gamma \ln h) \leq \frac{1}{h^{\gamma-1} \ln h} \leq \frac{1}{15^{\gamma-1} \ln 15} = \frac{1}{3}.$$

Let X be a random variable over $\{0,1\}$ defined as $X = 1$ if and only if $f(H) = [k]$, where $f \leftarrow_{\$} \mathrm{Map}(m,k)$. Let $p = \Pr[X = 1]$. We have that $p \geq 2/3$. Let $X_1, \ldots, X_w$ be i.i.d. random variables over $\{0,1\}$ such that $\Pr[X_u = 1] = p$ for all u. Note that $p = \mathbb{E}\left[(1/w)\sum_{u \in [w]} X_u\right]$. From the Chernoff bound, we obtain that

$$\begin{aligned}
\Pr\left[\sum_{u \in [w]} X_u \leq \frac{w}{2}\right] &= \Pr\left[\frac{1}{w}\sum_{u \in [w]} X_u \leq p - \left(p - \frac{1}{2}\right)\right] \\
&\leq \left(\left(\frac{p}{1/2}\right)^{1/2}\left(\frac{1-p}{1/2}\right)^{1/2}\right)^w \\
&= (4p(1-p))^{w/2} \\
&\leq \left(\frac{8}{9}\right)^{w/2} \\
&\leq \exp\left(-\frac{w}{18}\right).
\end{aligned}$$

From the definition of X_u, we have that

$$\Pr\left[f_1, \ldots, f_w \leftarrow_{\$} \mathrm{Map}(m,k) : |\{u \in [w] : f_u(H) = [k]\}| \leq \frac{w}{2}\right] \leq \exp\left(-\frac{w}{18}\right).$$

It follows from the union bound that

$$\begin{aligned}
&\Pr\left[f_1,\ldots,f_w \leftarrow\!\!\$\, \mathrm{Map}(m,k) : \exists H \in \binom{[m]}{h}, |\{u \in [w] : f_u(H) = [k]\}| \leq \frac{w}{2}\right] \\
&\leq \binom{m}{h} \exp\left(-\frac{w}{18}\right) \\
&\leq 2^m \exp\left(-\frac{w}{18}\right) \\
&= \exp\left(m \ln 2 - \frac{w}{18}\right).
\end{aligned}$$

We can also see that if $f_1,\ldots,f_w \leftarrow\!\!\$\, \mathrm{Map}(m,k)$, the probability that there are $i,j \in [w]$ such that $f_i = f_j$ is at most

$$\binom{w}{2}\frac{1}{k^m} < \frac{w^2}{2}\cdot\frac{1}{2^m} = \frac{w^2}{2^{m+1}}.$$

Therefore, if $f_1,\ldots,f_w \leftarrow\!\!\$\, \mathrm{Map}(m,k)$, the probability that the set $\mathcal{F} = \{f_1,\ldots,f_w\}$ is of size w and $|\{f \in \mathcal{F} : f(H) = [k]\}| > w/2$ for all $H \in \binom{[m]}{h}$ is at least

$$1 - \exp\left(m\ln 2 - \frac{w}{18}\right) - \frac{w^2}{2^{m+1}}.$$

If we set $w = 14m$, then the above value is

$$\begin{aligned}
&1 - \exp\left(-\left(\frac{7}{9} - \ln 2\right)m\right) - \frac{14^2 m^2}{2^{m+1}} \\
&\geq 1 - \exp\left(-\left(\frac{7}{9} - \ln 2\right)\cdot 15\right) - \frac{14^2\cdot 15^2}{2^{16}} \\
&= 1 - 0.2809\cdots - 0.6729\cdots \\
&> 0
\end{aligned}$$

since $m \geq 15$ and $7/9 - \ln 2 > 0$. Therefore, an (m,h,k)-locally surjective map family of size $w = 14m$ indeed exists.

C Refined Constructions of Locally Surjective Map Families

We show that it is possible to make overwhelming the success probabilities of probabilistic constructions of nearly perfect hash families and locally surjective map families in Propositions 1 and 8.

Proposition 13. *Let $\epsilon > 0$ and $m,h,k \in \mathbb{N}$ be such that $m \geq h \geq 3$ and $k \leq h/\ln h$. Let w be any integer such that*

$$w > 8m + \frac{\ln \epsilon^{-1}}{\ln\ln 3}. \tag{2}$$

If we choose w functions $f_1, \ldots, f_w$ independently and uniformly at random from $\mathrm{Map}(m, k)$*, then $\mathcal{F} = \{f_1, \ldots, f_w\}$ is an (m, h, k)-nearly perfect hash family with probability at least $1 - \epsilon$.*

Proof. Fix $H \in \binom{[m]}{h}$. As in the proof of Proposition 8, we have that

$$\Pr[f \leftarrow_{\$} \mathrm{Map}(m, k) : f(H) \neq [k]] \leq k \left(1 - \frac{1}{k}\right)^h \leq \frac{1}{\ln h}.$$

We then obtain that

$$\Pr[f_1, \ldots, f_w \leftarrow_{\$} \mathrm{Map}(m, k) : \forall u \in [w], f_u(H) \neq [k]] \leq \frac{1}{(\ln h)^w}.$$

It follows from the union bound that

$$\begin{aligned}
&\Pr\left[f_1, \ldots, f_w \leftarrow_{\$} \mathrm{Map}(m, k) : \exists H \in \binom{[m]}{h}, \forall u \in [w], f_u(H) \neq [k]\right] \\
&\leq \frac{\binom{m}{h}}{(\ln h)^w} \\
&\leq \frac{2^m}{(\ln h)^w} \\
&\leq \exp(-(w \ln \ln 3 - m \ln 2)).
\end{aligned}$$

If we set $w = 8m$, then the above value is less than 1 and we obtain Proposition 1. If we choose w satisfying the condition (2), then the above probability is less than ϵ. □

Proposition 14. *Let $\epsilon > 0$ and $m, h, k \in \mathbb{N}$ be such that $h \geq 15$,*

$$m \geq 18 + \frac{\ln \epsilon^{-1}}{1 - \ln 2} \text{ and } k \leq \frac{h}{\gamma \ln h},$$

where $\gamma = 1 + (\ln 3 - \ln \ln 15)/(\ln 15) < 1.04$. If we choose $w := 18m$ functions $f_1, \ldots, f_w$ independently and uniformly at random from $\mathrm{Map}(m, k)$*, then $\mathcal{F} = \{f_1, \ldots, f_w\}$ is an (m, h, k)-locally surjective map family with probability at least $1 - \epsilon$.*

Proof. From the proof of Proposition 8, the probability that $\mathcal{F}$ is not an (m, h, k)-locally surjective map family is upper bounded by

$$q := \exp\left(m \ln 2 - \frac{w}{18}\right) + \frac{w^2}{2^{m+1}}.$$

Since $m \geq 15$, we have that

$$\frac{m^2}{\sqrt{2}^m} \leq \frac{15^2}{\sqrt{2}^{15}} =: c_1.$$

We thus obtain that

$$\begin{aligned} q &\leq \exp\left(-m(1-\ln 2)\right) + \frac{18^2 c_1}{2} \cdot (\sqrt{2})^{-m} \\ &= \exp\left(-m(1-\ln 2)\right) + c_2 \exp\left(-m \ln \sqrt{2}\right), \end{aligned}$$

where $c_2 := 18^2 c_1/2$. From the condition on m and the fact that $18(1-\ln 2) > \ln(c_2+1)$, we have that $\exp\left(-m(1-\ln 2)\right) \leq \epsilon/(c_2+1)$. In addition, since $1 - \ln 2 < \ln \sqrt{2}$, it holds that

$$m \geq \frac{\ln(c_2+1)}{1-\ln 2} + \frac{\ln \epsilon^{-1}}{\ln \sqrt{2}} \geq \frac{\ln(c_2+1) + \ln \epsilon^{-1}}{\ln \sqrt{2}}$$

and hence that $\exp\left(-m \ln \sqrt{2}\right) \leq \epsilon/(c_2+1)$. We thus obtain that $q \leq \epsilon/(c_2+1) + c_2 \cdot \epsilon/(c_2+1) = \epsilon$. □

D Proof of Proposition 11

Notations.

- Let Π_0 be the conflict-finding m-server PIR protocol described in Fig. 4.
- Let $\kappa \in \mathbb{N}$.
- A client C has an index $\tau \in [n]$ and each server S_i has the database $\boldsymbol{a} \in \mathcal{X}^n$.

Protocol.

1. C executes κ independent instances of Π_0 in parallel.
2. Let

$$(x^{(1)}, \star), \ldots, (x^{(\kappa_1)}, \star), (\mathsf{conflict}, (G_0^{(1)}, G_1^{(1)})), \ldots, (\mathsf{conflict}, (G_0^{(\kappa_2)}, G_1^{(\kappa_2)}))$$

   be the (rearranged) outputs of the κ instances of Π_0, where $\kappa_1, \kappa_2 \geq 0$, $\kappa_1 + \kappa_2 = \kappa$, $x^{(h)} \in \mathcal{X}$ and $(G_0^{(h)}, G_1^{(h)})$ is a partition of the set of m servers.
3. C does the following:
   (a) If $\kappa_2 \geq 1$, C chooses $h \in [\kappa_2]$ uniformly at random and outputs $(\mathsf{conflict}, (G_0, G_1))$, where $G_0 = G_0^{(h)}$ and $G_1 = G_1^{(h)}$.
   (b) If $\kappa_2 = 0$ and there exists $y \in \mathcal{X}$ such that $|\{h \in [\kappa] : x^{(h)} = y\}| > \kappa/2$, then C outputs $(y, \star)$.
   (c) Otherwise, C outputs $(\perp, \mathsf{failure})$.

Fig. 5. A construction of conflict-finding PIR from regular PIR

Let Π_0 be a t-private $(b; 1-\epsilon_0)$-conflict-finding PIR protocol for $\epsilon_0 = m/\lambda$ given by Proposition 10. Consider a PIR protocol Π_1 described in Fig. 5. The t-privacy follows from that of Π_0. The correctness easily follows since if all servers behave honestly, then $\kappa_1 = \kappa$, $\kappa_2 = 0$ and $x^{(h)} = a_\tau$ for all h.

To see that Π_1 satisfies the soundness property in Definition 8, first observe that a client outputs $(y, z) = (\perp, \mathsf{failure})$ only if he proceeds to Step 3(c), which means that $\kappa_1 = \kappa$ and $|\{h \in [\kappa] : x^{(h)} \neq a_\tau\}| \geq \kappa/2$. Since Π_0 is $(b; 1-\epsilon_0)$-conflict-finding, the probability that it occurs is at most

$$\epsilon_0^{\kappa/2} \cdot \binom{\kappa}{\kappa/2} \leq \left(\frac{m}{\lambda}\right)^{\kappa/2} \cdot 2^\kappa = \left(2\sqrt{\frac{m}{\lambda}}\right)^\kappa \leq \epsilon_1$$

if we set $\lambda = 16m$ and $\kappa = \log \epsilon_1^{-1}$. Similarly, we can see that $y \in \mathcal{X} \setminus \{a_\tau\}$ and $z = \star$ with probability at most ϵ_1 since the adversary now must make sure that $|\{h \in [\kappa] : x^{(h)} = y\}| > \kappa/2$ for some $y \neq a_\tau$. Next, if $y = \mathsf{conflict}$, we have that for any $h \in [\kappa_2]$, either $H \subseteq G_0^{(h)}$ or $H \subseteq G_1^{(h)}$ occurs with probability 1, where H is the set of all honest servers. In particular, neither G_0 nor G_1 is the empty set and either $H \subseteq G_0$ or $H \subseteq G_1$ occurs. Thus Π_1 satisfies the conflict finding property in Definition 8. Finally, the z-output of Π_1 is determined only by the z-outputs of Π_0, which are independent of a client's index τ. Hence Π_1 satisfies the z-independence property in Definition 8.

If we set $\lambda = 16m$ and $\kappa = \log \epsilon_1^{-1}$, the communication complexity of Π_1 is $O(\kappa \cdot \mathrm{Comm}(\Pi_0)) = O(m^2(\log \epsilon_1^{-1}) \cdot \mathrm{Comm}(\Pi))$. It can also be seen that $\text{s-Comp}(\Pi_1) = O(m^2(\log \epsilon_1^{-1}) \cdot \text{s-Comp}(\Pi))$. Observe that the client-side computational complexity of Step 1 is $\kappa \cdot \text{c-Comp}(\Pi_0) = O(m^3(\log \epsilon_1^{-1}) \cdot \text{c-Comp}(\Pi))$. The other steps can be done in time $O(m\kappa \log |\mathcal{X}|) = O(m(\log \epsilon_1^{-1}) \cdot \text{c-Comp}(\Pi))$. Note that it is possible to find the majority of a sequence $x^{(1)}, \ldots, x^{(\kappa_1)} \in \mathcal{X}$ in time $O(\kappa_1 \log |\mathcal{X}|)$ by the Boyer-Moore algorithm [8]. Thus we have that $\text{c-Comp}(\Pi_1) = O(m^3(\log \epsilon_1^{-1}) \cdot \text{c-Comp}(\Pi))$.

E Definition of Error-Detecting PIR

For completeness, we provide the formal definition of b-error-detecting PIR [15], which allows a client to detect up to b errors.

Definition 9 (Error-detecting PIR). *A PIR protocol Π is said to be $(b; 1-\epsilon)$-error-detecting if the following conditions hold:*

- *Π satisfies correctness in Definition 2;*
- *A client C is allowed to output a special symbol $\perp$ and for any $\boldsymbol{a} \in \mathcal{X}^n$, any $\tau \in [n]$ and any malicious adversary $\mathcal{B}$ who corrupts at most b servers, the probability that C outputs a_τ or $\perp$ at the end of the protocol is at least $1-\epsilon$.*