A Fast Multi-Server, Multi-Block Private Information Retrieval Protocol

Luqin Wang*, Trishank Karthik Kuppusamy*, Yong Liu† and Justin Cappos*
*Department of Computer Science and Engineering
†Department of Electrical and Computer Engineering
New York University, New York, NY USA

***Abstract*—Private Information Retrieval (PIR) allows users to retrieve information from a database without revealing which information in the database was queried. The traditional information-theoretic PIR schemes utilize multiple servers to download a single data block, thus incurring high communication overhead and high computation burdens. In this paper, we develop an information-theoretic multi-block PIR scheme that significantly reduces client communication and computation overheads by downloading multiple data blocks at a time. The design of k-safe binary matrices insures the information will not be revealed even if up to k servers collude. Our scheme has much lower overhead than classic PIR schemes. The implementation of fast XOR operations benefits both servers and clients in reducing coding and decoding time. Our work demonstrates that multi-block PIR scheme can be optimized to simultaneously achieve low communication and computation overhead, comparable to even non-PIR systems, while maintaining a high level of privacy.**

I. Introduction

In many network applications, a client would like to retrieve information from a server without revealing to the server what it wants to download. For example, an inventor may want to query a patent database without revealing which patents she wants to retrieve [1], a trader may want to get specific stock quotes without revealing the investment she may have or want to make [2], or a client may want to download a security update without revealing which unpatched vulnerability the update addresses [3]. Private Information Retrieval (PIR) allows users to retrieve information from a database without revealing the queries to anyone, including the database server itself. Computational PIR [4]–[7], which leverages a single database server, is known to be impractical [8]. However, recent results have shown that information-theoretic PIR [9], which uses multiple mirrors that containing copies of the server data, can, in some cases, perform in a similar manner to non-PIR systems [3], [10].

In classic multi-server information-theoretic PIR schemes, each mirror keeps a replicated copy of the database. In order to retrieve one data block, a client needs to send multiple random binary coefficients as requests to different mirrors, and decode from those received mixed blocks [9]. While binary calculation can be fast on mirrors, privacy comes at the price of greatly increased communication overhead. More recently, Henry at el. [11] showed that if a client requests multiple data blocks, it is possible to *reuse randomly mixed data blocks across multiple requests*. Although this reduces communication overhead while maintaining the same level of information retrieval privacy, the usage of error-correcting code results in a constant communication overhead, which cannot be further reduced. Moreover, they leverage computationally expensive encoding and decoding operations that substantially decrease the throughput of the resulting systems.

In this paper, we present the first matrix-based information-theoretic PIR scheme that combines multiple block requests with the use of fast XOR operations instead of more computationally expensive operations. Using fast XOR operations allows us to reuse the same high performance PIR mirror infrastructure that has been shown to have similar goodput to FTP on realistic datasets and deployment environments [3]. However, using multiple block requests has benefits over this existing PIR scheme. By leveraging multiple block requests, our proposed scheme can further increase performance and significantly reduce communication overhead.

The rest of the paper is organized as follows. We briefly review the related work in Section II. The PIR system architecture and the threat model are presented in Section III. The single-block PIR scheme is introduced in Section IV. We develop our multi-block PIR scheme in Section V. The overhead and privacy of the proposed PIR scheme are evaluated in Section VI. Section VII discusses the robustness against Byzantine failures. We conclude the paper in Section VIII.

II. Related Work

In 1995, Chor et al. proposed PIR as a novel mechanism for allowing clients to obtain information from a database without disclosing to the server what was being retrieved [9]. Subsequently, other researchers pointed out two perceived weaknesses of the basic scheme: the need for multiple non-communicating servers (referred to as the replication problem) and the communication overhead. In 1997, Kushilevitz and Ostrovsky solved both problems by moving from the information-theoretic model to a model that admits computationally bounded adversaries [12]. This work proved that one could obtain provable privacy with sublinear asymptotic complexity using only a single server, which spurred the exploration of the CPIR (Computationally Private Information Retrieval) problem [5], [12]–[15]. Many of the CPIR studies focused on reducing communication cost at the expense of computational complexity.

Despite the rich literature on theoretical PIR schemes, very few efforts were made to implement those schemes. Recent studies by Sion and Carbunar [8], Yoshida et al. [16] and Sassaman et al. [17] revealed the impracticality of the existing computational PIR schemes and pointed out that many are

presently impractical and, given hardware trends, unlikely to improve [8]. These critics argued that it would be faster to transfer the entire database than to use most of the proposed PIR schemes.

Olumofin and Goldberg [10] refuted such claims and proved the feasibility of PIRs by publishing performance results for a single-server lattice-based PIR system [19] and two multi-server information-theoretic PIR systems [9] [20] which do not use the primitives mentioned as impractical in Sion and Carbunar's prior work. Olumofin showed that Goldberg's system [20] can be one to three orders of magnitude faster than transferring the entire database. Cappos [3] also demonstrated that Chor's PIR scheme can be implemented with high computational efficiency and that its performance is similar to non-private protocols such as HTTP and FTP in practice.

Henry et al. [11] proposed a multi-block scheme for [20] Goldberg's design by encoding multiple data block requests into a single PIR query. However, in order to tackle Byzantine failure and enhance robustness, their scheme applies both computationally expensive operations over finite field and error-correcting codes. The resulting system incurs high communication overhead and slow client decoding time. Demmler et al. [18] developed a multi-block scheme which further reduced the communication overhead. We propose a novel multi-block scheme that is more flexible in the retrieval of all data block requests than Demmler et al. demonstrated. In addition, our scheme was able to substantially reduce communication overhead even further than all previous schemes. Lastly, leveraging fast XOR operations significantly decreases the encoding and decoding time of the system.

III. Architecture and Threat Model of Private Information Retrieval

A. Architecture

A PIR system typically has three components.

- **Vendor:** A vendor produces the database which contains blocks of data desired by clients. This database is public and can be read by any party. The vendor builds a manifest for this database that describes the secure hashes of each block. The vendor is also responsible for maintaining a list of correctly-operating mirrors.
- **Client:** A client requests one or more blocks of data from the database. In order to retrieve data, a client first contacts the vendor to get the list of mirrors and the manifest. The client then makes requests to the mirrors to retrieve content.
- **Mirror:** A mirror obtains a copy of the database and provides blocks of data to client. When a mirror gets a request from a client, it generates a response according to the request (described below) and sends back a signed response to the client.

B. Threat model

To understand the scope of issues that our work will address, we use the following threat model which comes from prior work [3], [9].

- The vendor is trusted to produce a valid database that the client wishes to retrieve. The vendor is largely trusted but wishes to reduce its bandwidth consumption by offloading client download requests to mirrors.
- Non-malicious mirrors may fail at any time, and will not respond to client queries.
- A malicious party may operate one or more mirrors. Therefore, the adversary may see all communications and decode any encrypted messages for their mirrors. Furthermore, these mirrors may share or publicize any information they receive. However, for the majority of this paper (until Section VII), we assume a malicious mirror is honest-but-curious. In other words, it will not corrupt or modify content, but it may collude with others and reveal information that could pose a threat.
- In Section VII, we relax the prior assumption and assume a mirror may act in a Byzantine manner, including modifying content.

To retrieve information privately from potential honest-but-curious mirrors, a client sends multiple requests to multiple mirrors. Some requests are for randomly mixed blocks to "confuse" the malicious mirrors, and the rest of the requests are carefully crafted so that after collecting all the responses, the client is able to decode and get all the data blocks she wants. In this paper, we assume all the working mirrors generate correct responses and the client receives the responses correctly. A PIR scheme protects client queries against collusion by malicious mirrors. We measure the privacy of a PIR scheme using a threshold model called k-safe PIR.

Definition 1: k-Safe PIR: a client information retrieval scheme is k-safe if it does not reveal any information about the client's query as long as the number of malicious mirrors is no greater than k.

IV. Single-Block PIR Scheme

In this section, we formally develop our multi-block PIR scheme that retrieves multiple data blocks with low communication overhead. We focus on PIR schemes that use only the binary XOR operation. We show that Chor-PIR, a k-safe PIR scheme for downloading a single block, can be extended to download multiple blocks with significantly reduced communication overhead, while maintaining the same information retrieval privacy against k malicious mirrors.

A. Notations

For clarity of presentation, we will use the following notations throughout the paper:

- D is the database containing N equal-sized data blocks, $D = [B_1 \quad B_2 \quad \dots \quad B_N]$, with each data block being a bit string with S bits.
- $e_l = [0 \quad 0 \quad \dots \quad 1 \quad \dots \quad 0]^{\mathrm{T}}$ is the position vector with $|e_l| = N$ and only the l-th bit is one.
- C_i is the block encoding the coefficient vector to be sent to the i-th mirror. C_i is a column vector with dimension N.

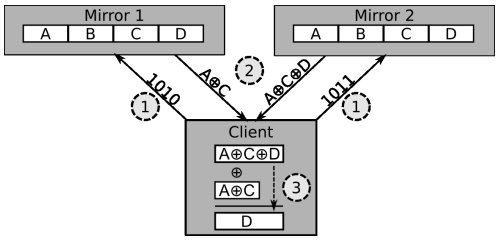

Fig. 1: Diagram of a Chor-PIR scheme to retrieve one data block

- E_i is the linearly encoded data block sent back by the i-th mirror to the client: $E_i = D \times C_i$, where addition and multiplication are defined in a finite field.
- k is the number of colluders: the client's query will not be revealed as long as no more than k mirrors collude.

B. Single-Block PIR

PIR of a single block has been studied extensively in previous works [1], [5], [6], [14], [21]. With the Chor-PIR protocol, which achieves privacy with up to k colluding mirrors, the client has to first download k randomly mixed blocks from k different mirrors, and then download the desired block (mixed with the k randomly mixed blocks) from the mirror $k+1$. More precisely:

1) The client generates k random bit strings $\{\xi_i, 1 \leq i \leq k\}$, with $|\xi_i| = N$, and sends ξ_i to mirror i as the block encoding coefficients:
$$C_i = \{\xi_{ij}, 1 \leq j \leq N\}, 1 \leq i \leq k.$$
2) Mirror i returns to the client the encoded block:
$$E_i = D \times C_i \triangleq \oplus_{j=1}^{N} \xi_{ij} B_j,$$
where the string operation is bit-wise, with XOR $\oplus$ addition and binary multiplication. Equivalently, the mirror works in the two-element finite field $GF(2)$.
3) The client sends to mirror $k+1$ the encoding coefficient vector:
$$C_{k+1} = \oplus_{i=1}^{k} \xi_i \oplus e_l.$$
4) Mirror $k+1$ returns the encoded block:
$$E_{k+1} = D \times (\oplus_{i=1}^{k} \xi_i \oplus e_l).$$
5) Finally, the client decodes the data block l:
$$\begin{aligned} \oplus_{i=1}^{k+1} E_i &= D\xi_1 \oplus \cdots \oplus D\xi_k \oplus D(\oplus_{i=1}^{k} \xi_i \oplus e_l) \\ &= D \times \big(\xi_1 \oplus \cdots \oplus \xi_k \oplus (\oplus_{i=1}^{k} \xi_i \oplus e_l)\big) \\ &= D \times e_l = B_l. \end{aligned}$$

Figure 1 shows a simple example of using Chor-PIR to privately retrieve a data block from two mirrors with $k = 1$. Due to the random bit strings $\{\xi_i, 1 \leq i \leq k\}$, the privacy of the client's query is preserved unless $k+1$ mirrors collude. The privacy comes at the price of downloading $k+1$ mixed data blocks to decode one original data block. If we assume that each mirror is malicious independently with probability $\hat{p}$, then the probability that the data query will never be revealed is:

$$Privacy\_Chor(\hat{p}, k) = 1 - \hat{p}^{(k+1)}. \quad (1)$$

The communication overhead, or the number of extra blocks needed to retrieve one data block, is:

$$Overhead\_Chor(\hat{p}, k) = k. \quad (2)$$

According to (1), given a malicious mirror probability $\hat{p}$, one has to choose a large k to achieve a high level of privacy, leading to high communication overhead.

V. Binary Multi-Block PIR (BMB-PIR)

When a client needs to download multiple blocks privately, one naïve way is to download each block independently using Chor-PIR. Then each block incurs a communication overhead of k. To reduce the communication overhead, one can try to reuse the randomly mixed data blocks from the first k mirrors, and then download another data block (say t) from mirror $k+2$ by sending a bit string $C_{k+2} = \oplus_{i=1}^{k} \xi_i \oplus e_t$. Unfortunately, if mirror $k+1$ and mirror $k+2$ collude, then they only need to add the encoding coefficient vectors from the client,

$$C_{k+1} \oplus C_{k+2} = (\oplus_{i=1}^{k} \xi_i \oplus e_l) \oplus (\oplus_{i=1}^{k} \xi_i \oplus e_t) = e_l \oplus e_t,$$

to eliminate all the random strings, and discover that the client wants to download blocks l and t. A more refined download scheme is needed to reuse the randomly mixed blocks and reduce the communication overhead.

A. Multi-Block Download Scheme

We propose a scheme for multi-block PIR based on the binary XOR operation to achieve privacy when up to k mirrors collude.

Definition 2: k-Safe Binary Matrix: we call a binary matrix R k-safe if any k columns of R are linearly independent under XOR addition.

If we can generate a k-safe binary matrix of the form:

$$R_{n\times m}^{\langle k\rangle} = \left[\begin{array}{cccc|ccc} 1 & * & \dots & * & v_{1,n+1} & \dots & v_{1,m} \\ 0 & 1 & \ddots & \vdots & v_{2,n+1} & \dots & v_{2,m} \\ \vdots & \ddots & \ddots & * & \vdots & \dots & \vdots \\ 0 & \dots & 0 & 1 & v_{n,n+1} & \dots & v_{n,m} \end{array}\right]_{n\times m} \quad (3)$$

$$= [RL_{n\times n}^{\langle k\rangle} | RR_{n\times(m-n)}^{\langle k\rangle}], \quad (4)$$

then we can download $m-n$ data blocks by reusing n randomly mixed data blocks and achieve privacy when up to k mirrors collude. Here is the client downloading strategy:

1) The client generates an $N \times n$ random binary matrix:
$$F \triangleq [\xi_1, \xi_2, \cdots, \xi_n],$$
where ξ_i is the i-th column, corresponding to a random binary string with length N.
2) The client sends mirror i the encoding vector:
$$C_i = F \times RL_{n\times n}^{\langle k\rangle}(i), \quad 1 \leq i \leq n,$$
where $RL_{n\times n}^{\langle k\rangle}(i)$ is the i-th column of matrix $RL_{n\times n}^{\langle k\rangle}$.

3) Mirror i sends the client back an encoded block:

$$E_i = D \times C_i.$$

4) The client decodes the n randomly mixed blocks by computing $D\xi_i$ as:

$$D\xi_i = [E_1, \cdots, E_n] \times RL^{-1}(i), \quad 1 \le i \le n,$$

where RL^{-1} is the inverse matrix of $RL^{\langle k \rangle}_{n\times n}$, and $RL^{-1}(i)$ is its i-th column.

5) The client sends to mirror $n+j$ the encoding vector:

$$C_{n+j} = F \times RR^{\langle k \rangle}_{n\times(m-n)}(j) \oplus e_{p_j}, \quad 1 \le j \le m-n,$$

where p_j is the index of the j-th data block the client wants to download.

6) Mirror $n+j$ returns to the client the encoded block:

$$\begin{aligned} E_{n+j} &= D \times \left(F \times RR^{\langle k \rangle}_{n\times(m-n)}(j) \oplus e_{p_j}\right) \\ &= [D\xi_1, \cdots, D\xi_n] \times RR^{\langle k \rangle}_{n\times(m-n)}(j) \oplus De_{p_j} \\ &= \oplus_{i=1}^{n} v_{i,n+j} D\xi_i \oplus B_{p_j}. \end{aligned}$$

7) The client decodes block p_j as:

$$B_{p_j} = E_{n+j} \oplus_{i=1}^{n} v_{i,n+j} D\xi_i, 1 \le j \le m-n.$$

Since $R^{\langle k \rangle}_{n\times m}$ is k-safe, then by definition, any subset of up to k mirrors cannot cancel out the random strings $\{\xi_1, \xi_2, \cdots, \xi_n\}$ by manipulating their received coding strings C_i from the client. So the client can download $m-n$ data blocks by first downloading n randomly mixed data blocks, and achieve privacy with up to k colluding mirrors. If the client wants to download more than $m-n$ blocks, it has to repeat the process.

B. Construction of K-safe Binary Matrix

The dimensions of a k-safe binary matrix determines the number of randomly mixed blocks the client needs to download and the number of data blocks it can thereafter retrieve. Now the challenge is to construct $R^{\langle k \rangle}_{n\times m}$ with small download overhead $\frac{n}{m-n}$. The single-block Chor-PIR is a special case of multi-block scheme with

$$R^{\langle k \rangle}_{k\times(k+1)} = \begin{bmatrix} 1 & 0 & \dots & 0 & 1 \\ 0 & 1 & \dots & 0 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \dots & 1 & 1 \end{bmatrix}_{k\times(k+1)},$$

which we call the basis k-safe matrix. It is easy to check that any of the k columns are linearly independent. Unfortunately, the communication overhead introduced by the basis k-safe matrix is k, which is too high. Now we propose an iterative algorithm to grow the basis k-safe matrix to reduce the communication overhead. By duplicating matrix $R^{\langle k \rangle}_{k\times(k+1)}$ from left to right, adding a $\lfloor k/2 \rfloor$-safe matrix to the bottom right and filling zeros in the bottom left, we have:

$$R^{\langle k \rangle}_{d(k,2k+2)\times(2k+2)} \triangleq \begin{bmatrix} R^{\langle k \rangle}_{k\times(k+1)} & R^{\langle k \rangle}_{k\times(k+1)} \\ 0 & R^{\langle \lfloor k/2 \rfloor \rangle}_{d(\lfloor k/2 \rfloor, k+1)\times(k+1)} \end{bmatrix}, \tag{5}$$

where $R^{\langle \lfloor k/2 \rfloor \rangle}_{d(\lfloor k/2 \rfloor, k+1)\times(k+1)}$ is a $\lfloor k/2 \rfloor$-safe matrix with $k+1$ columns. The number of rows is determined by a function $d(k,m)$. For the basis k-safe matrix, we have $d(k, k+1) = k$.

Proposition 5.1: The binary matrix $R^{\langle k \rangle}_{d(k,2k+2)\times(2k+2)}$ constructed in (5) is k-safe.

Proof: Since any k columns of $R^{\langle k \rangle}_{k\times(k+1)}$ are linearly independent, if we apply Gaussian column elimination with XOR addition to any k columns of the expanded matrix $R^{\langle k \rangle}_{d(k,2k+2)\times(2k+2)}$, the only way to cancel out the upper portion of k columns of $R^{\langle k \rangle}_{d(k,2k+2)\times(2k+2)}$ is to take exactly the same $\lfloor k/2 \rfloor$ vectors from the left and right half. However, since the bottom-right matrix is $\lfloor k/2 \rfloor$-safe, the bottom portion of those k columns will never be canceled out. So $R^{\langle k \rangle}_{d(k,2k+2)\times(2k+2)}$ is indeed k-safe. ■

According to the expansion process, we have

$$d(k, 2k+2) = d(k, k+1) + d(\lfloor k/2 \rfloor, k+1).$$

By switching columns, we can convert $R^{\langle k \rangle}_{d(k,2k+2)\times(2k+2)}$ into a k-safe matrix of the form in (3), with $m = 2k+2$, $n = d(k, 2k+2)$. The left upper triangular property ensures that the left side square matrix is invertible.

More generally, given a k-safe matrix $R^{\langle k \rangle}_{d(k,m)\times m}$ with m columns and $d(k,m)$ rows, one can expand it into a k-safe matrix with $2m$ columns using a similar process:

$$R^{\langle k \rangle}_{d(k,2m)\times 2m} = \begin{bmatrix} R^{\langle k \rangle}_{d(k,m)\times m} & R^{\langle k \rangle}_{d(k,m)\times m} \\ 0 & R^{\langle \lfloor k/2 \rfloor \rangle}_{d(\lfloor k/2 \rfloor, m)\times m} \end{bmatrix},$$

with $d(k, 2m) = d(k, m) + d(\lfloor k/2 \rfloor, m)$. The proof is a straightforward extension of Proposition 5.1.

For example, if we set $k = 4$, then the 4-safe matrix (without expansion) is:

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}_{4\times 5}.$$

After one expansion, we see that:

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}_{7\times 10}.$$

VI. Overhead and Privacy Evaluation

To determine whether our proposed Multi-Block PIR scheme has better performance than the single-block Chor-PIR scheme, we compare their communication overhead and privacy under the same threat model.

A. Analysis

The overhead and privacy of Chor-PIR are analyzed in (1) and (2) respectively. For BMB-PIR based on a k-safe binary matrix $R^{\langle k \rangle}_{d(k,m)\times m}$, we use m mirrors to download $m - d(k,m)$ data blocks in a k-safe manner. If each mirror is malicious with probability $\hat{p}$, then the query privacy is preserved if no

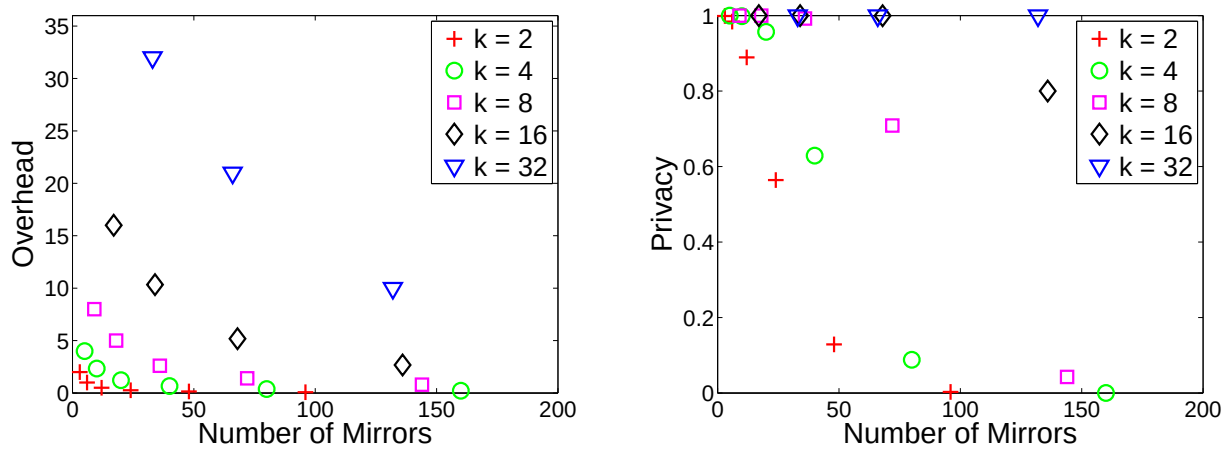

Fig. 2: In BMB-PIR, as the number of employed mirrors increases, the communication overhead decreases while the privacy level also decreases.

more than k mirrors are malicious. Therefore, the privacy of BMB-PIR can be calculated as:

$$Privacy\_BMB(k,m,\hat{p}) = \sum_{i=0}^{k} \binom{m}{i} \hat{p}^i \left(1-\hat{p}\right)^{m-i}. \quad (6)$$

And the communication overhead is:

$$Overhead\_BMB(k,m) = \frac{d\left(k,m\right)}{m - d\left(k,m\right)}. \quad (7)$$

For the 4-safe example considered at the end of the previous section, we can see that the overhead is 4 without expansion. After one round of expansion, the overhead becomes 2.3333, a reduction of nearly 40%.

B. Numerical Examples

Figure 2 illustrates the way that privacy level and communication overhead change as the number of mirrors m increases in BMB-PIR. The data points in the figure are calculated with the assumption that all the mirrors have the same probability of being malicious, $\hat{p} = 0.1$. We see that as m grows, communication overhead decreases dramatically. For example, to achieve 8-safe, the communication overhead reduces from 8 with 9 mirrors (essential Chor-PIR) to 2.6 with 36 mirrors and to 1 with 128 mirrors. However, as more mirrors are employed to download more data blocks, the probability that more than k mirrors are malicious also increases; consequently, the privacy level decreases substantially.

TABLE I: Lowest Overhead Achieved by Chor-PIR at Different Target Privacy Levels

Privacy	$\langle k, m\rangle$	Overhead
0.9	$\langle 1, 2\rangle$	1
0.99	$\langle 1, 2\rangle$	1
0.999	$\langle 2, 3\rangle$	2
0.9999	$\langle 3, 4\rangle$	3
0.99999	$\langle 4, 5\rangle$	4

Given a probability $\hat{p} = 0.1$ that a mirror is honest-but-curious, Table I and II report for the Chor-PIR and BMB-PIR protocols the lowest communication overhead that each can achieve at target privacy levels $\bar{p}$ ranging from 0.9 to 0.99999, and the corresponding best $\langle k, m\rangle$ settings. We can see that to achieve the same target privacy of 0.9, BMB-PIR reduces the overhead by a factor of 66.7%. As the target privacy level increases, BMB-PIR has to employ more mirrors. The overhead reductions at privacy levels 0.9999 and 0.99999 are 26.7% and 14.7% respectively.

VII. Robustness against Byzantine Failures

Previous analysis assumes the mirrors work correctly. However, if some mirrors act in a Byzantine manner or fail, the client can still correctly retrieve the data using algorithm 1. The idea is to repeatedly download $k+1$ blocks (k mixed blocks plus 1 data block) until the data block is decoded successfully. For the remaining $m-k-1$ data blocks, the client downloads and decodes them one by one. If the decoding fails, the client simply switches to another random mirror and downloads the data block until it is successfully decoded. We show the communication overhead of BMB-PIR as follow. Assuming that a mirror fails with probability $\bar{p}$. In order to successfully decode the first data block, the number of blocks the client needs to download:

$$\begin{aligned}&(k+1) \times \sum_{L=1}^{\infty} L[1-(1-\bar{p})^{k+1}]^{(L-1)}(1-\bar{p})^{k+1} \\ &= \frac{k+1}{(1-\bar{p})^{k+1}}\end{aligned} \quad (8)$$

To decode the rest $m-k-1$ data blocks, the number of blocks the client needs to download is:

$$(m-k-1) \times \sum_{L=1}^{\infty} L\bar{p}^{(L-1)}(1-\bar{p}) = \frac{m-k-1}{1-\bar{p}} \quad (9)$$

TABLE II: Lowest Overhead Achieved by BMB-PIR at Different Target Privacy Levels

Privacy	$m \leq 8$		$m \leq 16$		$m \leq 32$		$m \leq 64$		$m \leq 128$		$m \leq 256$	
	overhead	$\langle k, m \rangle$	**overhead**	$\langle k, m \rangle$	**overhead**	$\langle k, m \rangle$	**overhead**	$\langle k, m \rangle$	**overhead**	$\langle k, m \rangle$	**overhead**	$\langle k, m \rangle$
0.9	0.3333	$\langle 1, 4 \rangle$	0.3333	$\langle 1, 4 \rangle$	0.3333	$\langle 1, 4 \rangle$	0.3333	$\langle 1, 4 \rangle$	0.3333	$\langle 1, 4 \rangle$	0.3333	$\langle 1, 4 \rangle$
0.99	1	$\langle 1, 2 \rangle$	1	$\langle 1, 2 \rangle$	1	$\langle 1, 2 \rangle$	1	$\langle 1, 2 \rangle$	1	$\langle 1, 2 \rangle$	1	$\langle 1, 2 \rangle$
0.999	2	$\langle 2, 3 \rangle$	2	$\langle 2, 3 \rangle$	2	$\langle 2, 3 \rangle$	1.9091	$\langle 15, 64 \rangle$	1.9091	$\langle 15, 64 \rangle$	1.9091	$\langle 15, 64 \rangle$
0.9999	3	$\langle 3, 4 \rangle$	2.2	$\langle 7, 16 \rangle$	2.2	$\langle 7, 16 \rangle$	2.2	$\langle 7, 16 \rangle$	2.2	$\langle 7, 16 \rangle$	2.2	$\langle 7, 16 \rangle$
0.99999	4	$\langle 4, 5 \rangle$	4	$\langle 4, 5 \rangle$	3.8	$\langle 11, 24 \rangle$	3.8	$\langle 11, 24 \rangle$	3.4138	$\langle 31, 128 \rangle$	3.4138	$\langle 31, 128 \rangle$

Input: k, m, and $\{C_1, C_2, \ldots, C_{m-k}\}$
Output: $\{B_{p_1}, B_{p_2}, \ldots, B_{p_{(m-k)}}\}$

Client Initialization:
for $i = 1$ **to** k **do**
  client randomly chooses a mirror, sends request vector C_i and gets encoded data block E_i
end
client randomly chooses a mirror, sends request vector $C_{k+j}, j = 1$ and gets encoded data block E_{k+j}
client decodes B_{p_1}
if *client fails to decode* B_{p_1} **then**
  go back to step 2;
end
for $j = 2$ **to** $m - k$ **do**
  client randomly chooses a mirror, sends request vector C_{k+j} and gets encoded data block E_{k+j}
  client decodes B_{p_j}
  if *client fails to decode* B_{p_j} **then**
    go back to step 11;
  end
end

Algorithm 1: Detect and Retransmit Protocol

Hence, the overall communication overhead with a mirror Byzantine failure probability $\bar{p}$ is:

$$\frac{k + 1 + (m - k - 1)(1 - \bar{p})^k}{(m - k)(1 - \bar{p})^{k+1}} - 1 \tag{10}$$

VIII. Conclusion

In this paper, we described a fast multi-block PIR scheme that is capable of efficiently and privately downloading multiple data blocks. Specifically, we showed that Chor's PIR scheme can be extended to download multiple data blocks at a time by a recursive construction to produce larger k-safe matrices from smaller ones. This significantly reduces the communication overhead of multi-block PIR retrieval. The XOR operations provide extremely fast coding and decoding time against other multi-block PIR schemes. As a result, our multiple-block PIR protocol can be used in practice to retrieve both small files, such as security updates, and large files, such as disk images, privately and efficiently.

References

[1] D. Asonov, "Private Information Retrieval - An Overview And Current Trends," in *GI Jahrestagung (2)*, pp. 889–894, 2001.

[2] E. Yang, J. Xu, and K. Bennett, "A fault-tolerant approach to secure information retrieval," in *Reliable Distributed Systems, 2002. Proceedings. 21st IEEE Symposium on*, pp. 12 – 21, 2002.

[3] J. Cappos, "Avoiding theoretical optimality to efficiently and privately retrieve security updates," in *Financial Cryptography and Data Security 2013 (FC 2013)*, 2013.

[4] A. Beimel, Y. Ishai, E. Kushilevitz, and J. franois Raymond, "Breaking the $O(n^{1/(2k1)})$ Barrier for Information-Theoretic Private Information Retrieval," in *In Proc. of the 43rd IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 261–270, 2002.

[5] C. Cachin, S. Micali, and M. Stadler, "Computationally private information retrieval with polylogarithmic communication," in *Proceedings of the 17th international conference on Theory and application of cryptographic techniques*, EUROCRYPT'99, (Berlin, Heidelberg), pp. 402–414, Springer-Verlag, 1999.

[6] D. Asonov and J.-C. Freytag, "Almost optimal private information retrieval," in *Privacy Enhancing Technologies*, pp. 209–223, Springer, 2003.

[7] A. Ambainis, "Upper bound on the communication complexity of private information retrieval," in *Automata, Languages and Programming*, pp. 401–407, Springer, 1997.

[8] R. Sion and B. Carbunar, "On the Computational Practicality of Private Information Retrieval," in *In Proceedings of the Network and Distributed Systems Security Symposium (NDSS).*, 2007.

[9] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private information retrieval," *Journal of the ACM (JACM)*, vol. 45, no. 6, pp. 965–981, 1998.

[10] F. G. Olumofin and I. Goldberg, "Revisiting the computational practicality of private information retrieval," in *Financial Cryptography*, pp. 158–172, 2011.

[11] R. Henry, Y. Huang, and I. Goldberg, "One (Block) Size Fits All: PIR and SPIR Over Arbitrary-Length Records via Multi-block PIR Queries," in *In Proceedings of the Network and Distributed Systems Security Symposium (NDSS).*, 2013.

[12] E. Kushilevitz and R. Ostrovsky, "Replication is not needed: single database, computationally-private information retrieval," in *Foundations of Computer Science, 1997. Proceedings., 38th Annual Symposium on*, pp. 364 –373, oct 1997.

[13] C. Blundo, P. D'Arco, and A. De Santis, "A t-private k-database information retrieval scheme," *International Journal of Information Security*, vol. 1, no. 1, pp. 64–68, 2001.

[14] K. S. Narayanam, "A Novel Scheme for Single Database Symmetric Private Information Retrieval," in *Financial Cryptography*, 2006.

[15] S. K. Mishra and P. Sarkar, "Symmetrically private information retrieval," in *INDOCRYPT*, pp. 225–236, 2000.

[16] R. Yoshida, Y. Cui, R. Shigetomi, and H. Imai, "The practicality of the keyword search using PIR," in *Information Theory and Its Applications, 2008. ISITA 2008. International Symposium on*, pp. 1 –6, dec. 2008.

[17] L. Sassaman, B. Preneel, and K. U. L. Esat-cosic, "The Byzantine Postman Problem: A Trivial Attack Against PIR-based Nym Servers," tech. rep., ESAT-COSIC 2007-001, 2007.

[18] D. Demmler, A. Herzberg, and T. Schneider, "Raid-pir: Practical multi-server pir," in *Proceedings of the 6th edition of the ACM Workshop on Cloud Computing Security*, pp. 45–56, ACM, 2014.

[19] C. Aguilar-Melchor and P. Gaborit, "A lattice-based computationally-efficient private information retrieval protocol," *Cryptol. ePrint Arch., Report*, vol. 446, 2007.

[20] I. Goldberg, "Improving the robustness of private information retrieval," in *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, SP '07, (Washington, DC, USA), pp. 131–148, IEEE Computer Society, 2007.

[21] M. Layouni, "Accredited symmetrically private information retrieval," in *IWSEC 2007. LNCS*, pp. 262–277, Springer, 2007.