Multi-Message Private Information Retrieval using Product-Matrix Minimum Storage Regenerating Codes

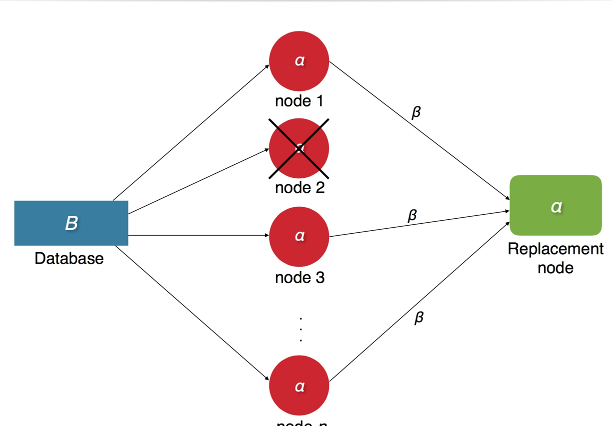
Chatdanai Dorkson and Siaw-Lynn Ng

Department of Mathematics, Royal Holloway, University of London

Abstract

Multi-message PIR scheme allows a user to download multiple messages from the database without revealing the identity of desired messages. Obviously, the user can repeatly use a single-message PIR scheme, but we wish for a more efficient way. In this work, we design a multi-message PIR using a product-matrix MSR codes from [1] achieving cPoP equals $\frac{p+2}{p}$ when p is the number of desired records. The use of regenerating codes beneficially reduces repair cost when a node failure occurs in the system. This work is the generalisation of our result on the single-message PIR [2].

Regenerating Codes



An $(n, k, r, \alpha, \beta, B)$ *regenerating code* stores B symbols among n nodes. Each node stores α symbols satisfying:

- The B symbols can be recovered from any k nodes
- If one node fails, we can connect to some set of r remaining nodes where $k \le r < n$, and gets β symbols from each of these r nodes to regenerate α symbols in the failed node.

Regenerating codes optimally trade repair bandwidth $(r\beta)$ with the amount of data stored per node (α). One interesting extremal point on the optimal trade-off curve is called the *minimum storage regeneration* (MSR) which minimises α first and then minimise $r\beta$.

Contact Information

Chatdanai.Dorkson.2016@rhul.ac.uk

References

[1] K. Rashmi, N. Shah, P. Kumar *Optimal Exact-Regenerating Codes for Distributed Storage at the MSR and MBR Points via a Product-Matrix Construction.* IEEE Trans. Inf. Theory 2011, pp.5227–5239.

[2] C. Dorkson and S. Ng *Private Information Retrieval using Product-Matrix Minimum Storage Regenerating Codes.* `http://arxiv.org/abs/1805.07190`. (Manuscript submitted for publication.)

System Model

- A database X consists of m records, each of length ℓ over $\mathbb{F}_q$, denoted by $X^1, X^2, \ldots, X^m$.
- Each record is encoded and distributed across n non-communicating nodes using the same product-matrix MSR code.
- A user who wants to download the record $X^{f_1}, X^{f_2}, \ldots, X^{f_p}$ submits a $d \times m\alpha$ query matrix Q_i over $GF(q)$ to node i
- Node i responds with an answer $A_i = Q_i S_i$ where S_i the column vector consisting of all symbols stored in node i.
- A scheme is *perfect information-theoretic* if for $i \in [n]$, Q_i gives no information about which records are being retrieved, and A_i ensures that the user can recover the desired records $X^{f_1}, \ldots, X^{f_p}$ with no errors.

An Example of Our Scheme

In our construction, we use the product-matrix MSR code from [1] with $n = (p+2)(k-2)$, over the finite field $\mathbb{F}_q$. The parameters of the MSR code are

$$(n, k, r, \alpha, \beta, B) = ((p+2)(k-2), k, 2k-2, k-1, 1, k(k-1)).$$

Assume $\ell = k(k-1)$ in this construction. We give an example to motivate our scheme.

Example: Suppose that we have 3 records over the finite field $\mathbb{F}_{13}$, each with size 6, so we can use an $(8, 3, 4, 2, 1, 6)$ product-matrix MSR code over $\mathbb{F}_{13}$ to encode each record. We write $X^i = \{x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5}, x_{i6}\}$. Choose the encoding matrix Ψ_8 to be the Vandermonde matrix, and the message matrix $\mathcal{M}_i$ for the record $i, i \in \{1, 2, 3\}$ as described in [1]:

$$\Psi_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 4 & 9 & 3 & 12 & 10 & 10 & 12 \\ 1 & 8 & 1 & 12 & 8 & 8 & 5 & 5 \end{bmatrix}^T, \quad \mathcal{M}_i = \begin{bmatrix} x_{i1} & x_{i2} \\ x_{i2} & x_{i3} \\ x_{i4} & x_{i5} \\ x_{i5} & x_{i6} \end{bmatrix}.$$

Hence, each node stores

node 1	node 2	node 3	node 4	node 5	node 6	node 7	node 8
$x_{11}+x_{12}+x_{14}+x_{15}$	$x_{11}+2x_{12}+4x_{14}+8x_{15}$	$x_{11}+3x_{12}+9x_{14}+x_{15}$	$x_{11}+4x_{12}+3x_{14}+12x_{15}$	$x_{11}+5x_{12}+12x_{14}+8x_{15}$	$x_{11}+6x_{12}+10x_{14}+8x_{15}$	$x_{11}+7x_{12}+10x_{14}+5x_{15}$	$x_{11}+8x_{12}+12x_{14}+5x_{15}$
$x_{12}+x_{13}+x_{15}+x_{16}$	$x_{12}+2x_{13}+4x_{15}+8x_{16}$	$x_{12}+3x_{13}+9x_{15}+x_{16}$	$x_{12}+4x_{13}+3x_{15}+12x_{16}$	$x_{12}+5x_{13}+12x_{15}+8x_{16}$	$x_{12}+6x_{13}+10x_{15}+8x_{16}$	$x_{12}+7x_{13}+10x_{15}+5x_{16}$	$x_{12}+8x_{13}+12x_{15}+5x_{16}$
$x_{21}+x_{22}+x_{24}+x_{25}$	$x_{21}+2x_{22}+4x_{24}+8x_{25}$	$x_{21}+3x_{22}+9x_{24}+x_{25}$	$x_{21}+4x_{22}+3x_{24}+12x_{25}$	$x_{21}+5x_{22}+12x_{24}+8x_{25}$	$x_{21}+6x_{22}+10x_{24}+8x_{25}$	$x_{21}+7x_{22}+10x_{24}+5x_{25}$	$x_{21}+8x_{22}+12x_{24}+5x_{25}$
$x_{22}+x_{23}+x_{25}+x_{26}$	$x_{22}+2x_{23}+4x_{25}+8x_{26}$	$x_{22}+3x_{23}+9x_{25}+x_{26}$	$x_{22}+4x_{23}+3x_{25}+12x_{26}$	$x_{22}+5x_{23}+12x_{25}+8x_{26}$	$x_{22}+6x_{23}+10x_{25}+8x_{26}$	$x_{22}+7x_{23}+10x_{25}+5x_{26}$	$x_{22}+8x_{23}+12x_{25}+5x_{26}$
$x_{31}+x_{32}+x_{34}+x_{35}$	$x_{31}+2x_{32}+4x_{34}+8x_{35}$	$x_{31}+3x_{32}+9x_{34}+x_{35}$	$x_{31}+4x_{32}+3x_{34}+12x_{35}$	$x_{31}+5x_{32}+12x_{34}+8x_{35}$	$x_{31}+6x_{32}+10x_{34}+8x_{35}$	$x_{31}+7x_{32}+10x_{34}+5x_{35}$	$x_{31}+8x_{32}+12x_{34}+5x_{35}$
$x_{32}+x_{33}+x_{35}+x_{36}$	$x_{32}+2x_{33}+4x_{35}+8x_{36}$	$x_{32}+3x_{33}+9x_{35}+x_{36}$	$x_{32}+4x_{33}+3x_{35}+12x_{36}$	$x_{32}+5x_{33}+12x_{35}+8x_{36}$	$x_{32}+6x_{33}+10x_{35}+8x_{36}$	$x_{32}+7x_{33}+10x_{35}+5x_{36}$	$x_{32}+8x_{33}+12x_{35}+5x_{36}$

Let Y^a_{ij} denote the j^{th} symbol stored in node i of the record a.

In the retrieval step, suppose the user wants to download X^1 and X^2. The query Q_i is a (3×6) matrix which we can interpret as 3 subqueries submitted to node i for each $i \in [8]$. To form the query matrices, the user generates a (3×6) random matrix $U = [u_{ij}]$ whose elements are chosen uniformly at a random from $\mathbb{F}_{13}$. Let

$$V_1 = \begin{bmatrix} 1&0&0&0&0&0 \\ 0&1&0&0&0&0 \\ 0&0&0&0&0&0 \end{bmatrix}, \quad V_2 = \begin{bmatrix} 0&0&0&0&0&0 \\ 1&0&0&0&0&0 \\ 0&1&0&0&0&0 \end{bmatrix}, \quad V_3 = \begin{bmatrix} 0&1&0&0&0&0 \\ 0&0&0&0&0&0 \\ 1&0&0&0&0&0 \end{bmatrix}, V_4 = \begin{bmatrix} 0&0&1&0&0&0 \\ 0&0&0&1&0&0 \\ 0&0&0&0&0&0 \end{bmatrix}, \quad V_5 = \begin{bmatrix} 0&0&0&0&0&0 \\ 0&0&1&0&0&0 \\ 0&0&0&1&0&0 \end{bmatrix}, \quad V_6 = \begin{bmatrix} 0&0&0&1&0&0 \\ 0&0&0&0&0&0 \\ 0&0&1&0&0&0 \end{bmatrix}.$$

For node $i \in [6]$, the query matrix is $Q_i = U + V_i$ and $Q_7 = Q_8 = U$. Then each node computes and returns the length-3 vector $A_i = Q_i S_i$ where S_i is a length-6 vector of symbols stored in node i. Write $A_i = (r_{i1}, r_{i2}, r_{i3})^T$. Let

$$w_1 = (x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32})^T, \quad w_2 = (x_{12}, x_{13}, x_{22}, x_{23}, x_{32}, x_{33})^T,$$
$$w_3 = (x_{14}, x_{15}, x_{24}, x_{25}, x_{34}, x_{35})^T, \quad w_4 = (x_{15}, x_{16}, x_{25}, x_{26}, x_{35}, x_{36})^T.$$

Consider first subquery 1, we obtain

$$Y^1_{11} + I_1 + I_2 + I_3 + I_4 = r_{11} \quad (1) \qquad I_1 + 5I_2 + 12I_3 + 8I_4 = r_{51} \quad (5)$$
$$I_1 + 2I_2 + 4I_3 + 8I_4 = r_{21} \quad (2) \qquad Y^2_{64} + I_1 + 6I_2 + 10I_3 + 8I_4 = r_{61} \quad (6)$$
$$Y^1_{32} + I_1 + 3I_2 + 9I_3 + I_4 = r_{31} \quad (3) \qquad I_1 + 7I_2 + 10I_3 + 5I_4 = r_{71} \quad (7)$$
$$Y^2_{43} + I_1 + 4I_2 + 3I_3 + 12I_4 = r_{41} \quad (4) \qquad I_1 + 8I_2 + 12I_3 + 5I_4 = r_{81} \quad (8)$$

where $I_l = u_1 w_l, l = 1, 2, 3, 4$, and u_1 is the first row of U. The user can solve for I_1, I_2, I_3, I_4 from (2), (5), (7), (8) as they form the equation with (4×4) submatrix of Ψ_8 which is invertible. Therefore, the user gets Y^1_{11}, Y^1_{32} for record 1 and Y^2_{43}, Y^2_{64} for record 2. Similarly, from subquery 2, the user obtains Y^1_{12}, Y^1_{21} for record 1 and Y^2_{44}, Y^2_{53} for record 2. Lastly, from subquery 3, the user obtains Y^1_{22}, Y^1_{31} for record 1 and Y^2_{54}, Y^2_{63} for record 2. Hence, the user has all the symbols of X^1 which are stored in the node 1, 2, 3 and all the symbols of X^2 which are stored in the node 4, 5, 6. From the property of regenerating codes, the user can reconstruct X^1 and X^2 as desired.

node 1	node 2	node 3	node 4	node 5	node 6	node 7	node 8
1	2	3					
2	3	1					
			1	2	3		
			2	3	1		

Table 1: Retrieval pattern for a (8,3,4,2,1,6) MSR code