Private Information Retrieval using Regenerating Codes

Chatdanai Dorkson and Siaw-Lynn Ng
Department of Mathematics, Royal Holloway, University of London

Objectives

Private information retrieval (PIR) schemes allow a user to download a file from the server without revealing any information on which record is retrieved. In the classical setting, the database is replicated among non-colluding nodes which results in high storage cost. This motivates the use of erasure codes. In our work, we construct PIR schemes where the database is stored using Minimum Storage Regenerating (MSR) codes which is a class of optimal regenerating codes provided efficient repair of a failed node. Hence, the resulting scheme has low storage cost with high efficiency of repair.

System Model of Replication-based PIR

- A database X consists of m records, each of length ℓ over $\mathbb{F}_q$, denoted by $X^1, X^2, \ldots, X^m$.
- The entire database is distributed across n non-colluding nodes.
- A user who wants to download the record X^f submits a query Q_i to node i, and then node i responds with an answer A_i for each $i \in [n]$
- A scheme is *perfect information-theoretic* if for $i \in [n]$, Q_i gives no information about which file is retrieved, and A_i ensures that the user can recover the desired record X^f with no errors.

Basic Example

- Suppose we have records $X^1, X^2, X^3 \in \mathbf{F}_2$ replicated and distributed across 2 nodes.
- Assume that a user wants the record X^1.
- The user generates a 3-bit random vector (u_1, u_2, u_3) uniformly at random, and chooses $Q_1 = (u_1, u_2, u_3)$ and $Q_2 = (u_1 + 1, u_2, u_3)$.
- Node 1 and node 2 are requested to return $A_1 = u_1X^1 \oplus u_2X^2 \oplus u_3X^3$. $A_2 = (u_1 + 1)X^1 \oplus u_2X^2 \oplus u_3X^3$.
- The user computes $A_1 \oplus A_2$ to get the record X^1.

Code-based PIR Schemes

Since the replication-based PIR results in high storage cost, this motivates the use of erasure codes which means that each node stores only a fraction of the entire database, hence reducing the storage cost while achieving reliability. We call this *Code-based PIR schemes*.

Measurements

Two measurements are usually used to analyse the efficiency of the PIR.

- *Storage overhead* is the ratio of the storage used in the scheme to the size of the database.
- *communication Price of Privacy (cPoP)* is the ratio of the amount of download data to the size of the desired record.

Regenerating Codes

An $(n, k, r, \alpha, \beta, B)$ *regenerating code* stores the database of size B among n nodes where each node stores α symbols satisfying two properties:

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

Regenerating codes optimally trade repair cost ($r\beta$) 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 the repair bandwidth.

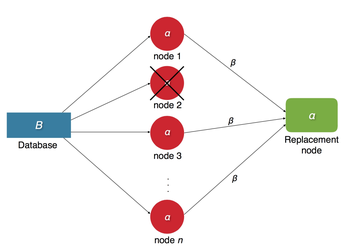

Figure 1: The repair process when one node fails

Our PIR Scheme

In our construction, we use the product-matrix MSR code from [1] with $n = 3k - 2$, over the finite field $\mathbb{F}_q$. The parameters of the MSR code are

$$(3k - 3, k, 2k - 2, k - 1, 1, k(k - 1)).$$

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

Example: Choose $k = 2$, so we use the $(4, 2, 2, 1, 1, 2)$ MSR code over $\mathbb{F}_{13}$. Suppose that the number of record is $m = 3$ each with size $\ell = 2$, so we can write $X^i = \{x_{i1}, x_{i2}\}$. We choose the encoding matrix Ψ_4 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].

$$\Psi_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \end{bmatrix}^T, \quad \mathcal{M}_i = \begin{bmatrix} x_{i1} \\ x_{i2} \end{bmatrix}.$$

Hence, each node stores

	node 1	node 2	node 3	node 4
record 1	$x_{11} + x_{12}$	$x_{11} + 2x_{12}$	$x_{11} + 3x_{12}$	$x_{11} + 4x_{12}$
record 2	$x_{21} + x_{22}$	$x_{21} + 2x_{22}$	$x_{21} + 3x_{22}$	$x_{21} + 4x_{22}$
record 3	$x_{31} + x_{32}$	$x_{31} + 2x_{32}$	$x_{31} + 3x_{32}$	$x_{31} + 4x_{32}$

Suppose the user wants record X^1. The query Q_i is a (1×3) matrix submitted to node i for each $i \in [4]$. The user generates a (1×3) random matrix $U = [u_{ij}]$ whose elements are chosen uniformly at random from $\mathbb{F}_{13}$. Let

$$V_1 = V_2 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}.$$

Choose $Q_1 = U + V_1, Q_2 = U + V_2$ and $Q_3 = Q_4 = U$. Each node computes and returns $A_i = Q_iS_i$ where S_i is a length-3 column vector of symbols stored in node i. Let

$$w_1 = (x_{11}, x_{21}, x_{31})^T, w_2 = (x_{12}, x_{22}, x_{32})^T.$$

We obtain

$$x_{11} + x_{12} + I_1 + I_2 = A_1 \quad (1)$$
$$x_{11} + 2x_{12} + I_1 + 2I_2 = A_2 \quad (2)$$
$$I_1 + 3I_2 = A_3 \quad (3)$$
$$I_1 + 4I_2 = A_4 \quad (4)$$

where $I_1 = U \cdot w_1$, and $I_2 = U \cdot w_2$. The user can solve for I_1, I_2 from (3), (4) as they form the system of equation with the submatrix of Ψ_4. Therefore, the user gets $x_{11} + x_{12}, x_{11} + 2x_{12}$ which are the symbols of X^1 stored in the first $k = 2$ nodes. From the property of the regenerating code, the user can reconstruct X^1 as desired.

node 1	node 2	node 3	node 4
1	1		

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

References

[1] K. V. Rashmi, N. B. Shah and P. V. 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.

Contact Information

- Email: Chatdanai.Dorkson.2016@live.rhul.ac.uk
- Twitter: @chat_coloc