PIR in Distributed Storage Systems: **My** Recent Results

Salim El Rouayheb

ECE Department
Illinois Institute of Technology

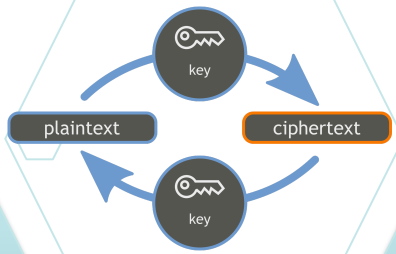
Google Maps is so concerned about privacy that it accidentally blurred out a cow's face

INSIDER

Jacob Shamsian, INSIDER

Sep. 17, 2016, 1:11 PM 4,755

The Many Facets of Privacy

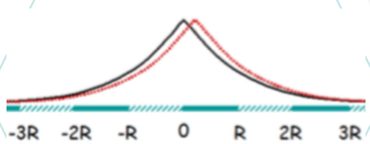


Crypto-
graphy

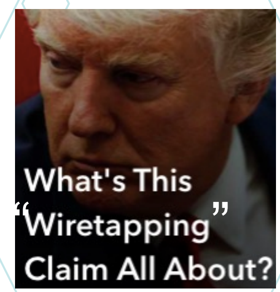
Anonymity

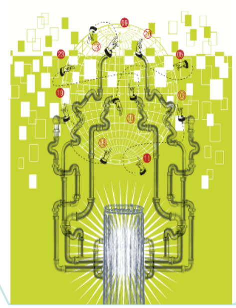
Physical
Layer
Security

Private
Information
Retrieval



Differential
Privacy



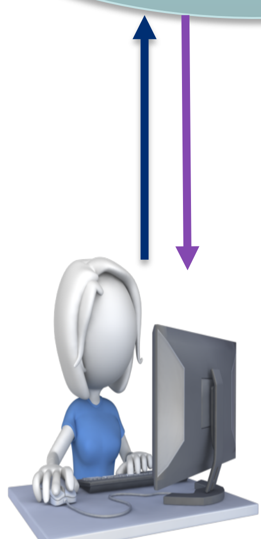


PIR IN DISTRIUTED SOTRAGE SYS.

Cloud

Anonymity: hide Alice's identity

PIR: Hide Alice's queries



Alice

Mr. Supreme Leader

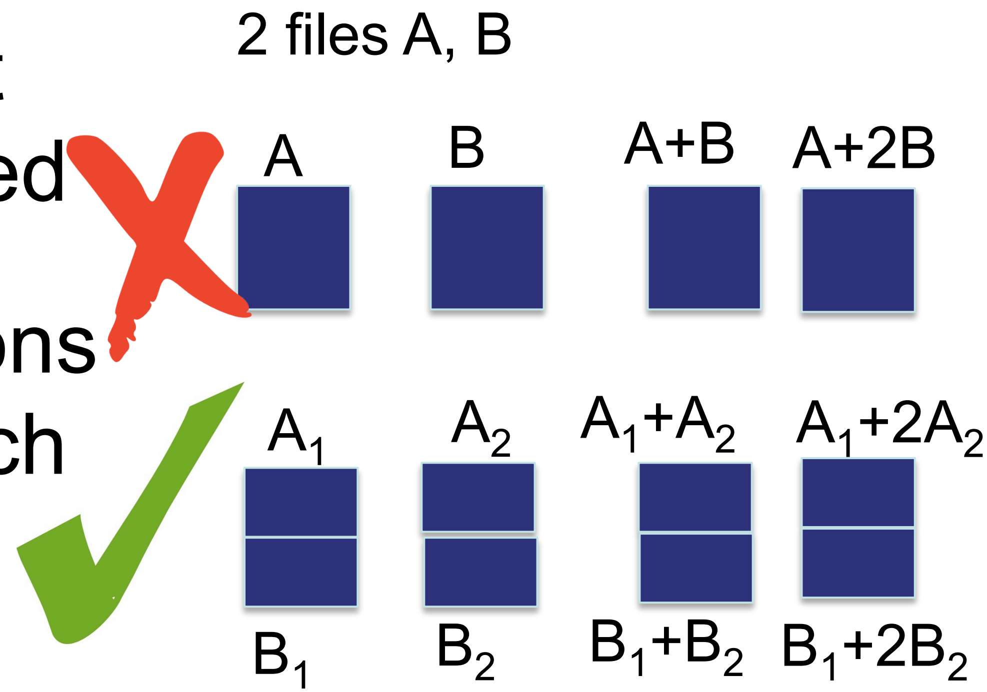
Coding for Reliable Distributed Storage

- Lots of research on codes for reliability in distributed storage systems
  - Regenerating codes, Locally Recoverable codes etc. [Dimakis et al.], [Tamo & Barg], [Yekhanin et al.] etc.
  - Many best paper awards
  - Microsoft says codes saved them millions of dollars

How are theoretical challenges of requiring privacy in PIR sense in addition to reliability in DSS?

Coding for **Privacy** and Reliability

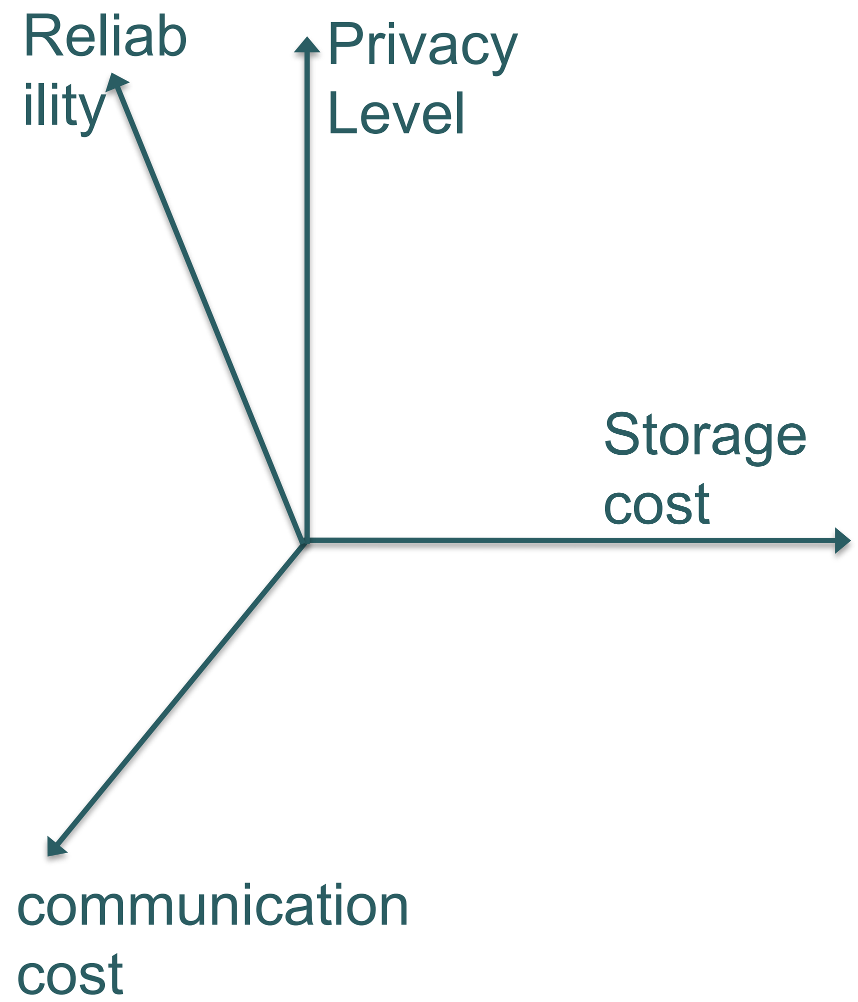
- Typically, coding different files together is not allowed
- Have to deal with collusions (nodes have to talk to each other for repair etc.)
- Locality, repair BW, etc.
- Many system overhead of PIR:
  1. Communication cost
  2. Storage cost
  3. Computational overhead
  4. Latency



Coding for **Privacy** and Reliability

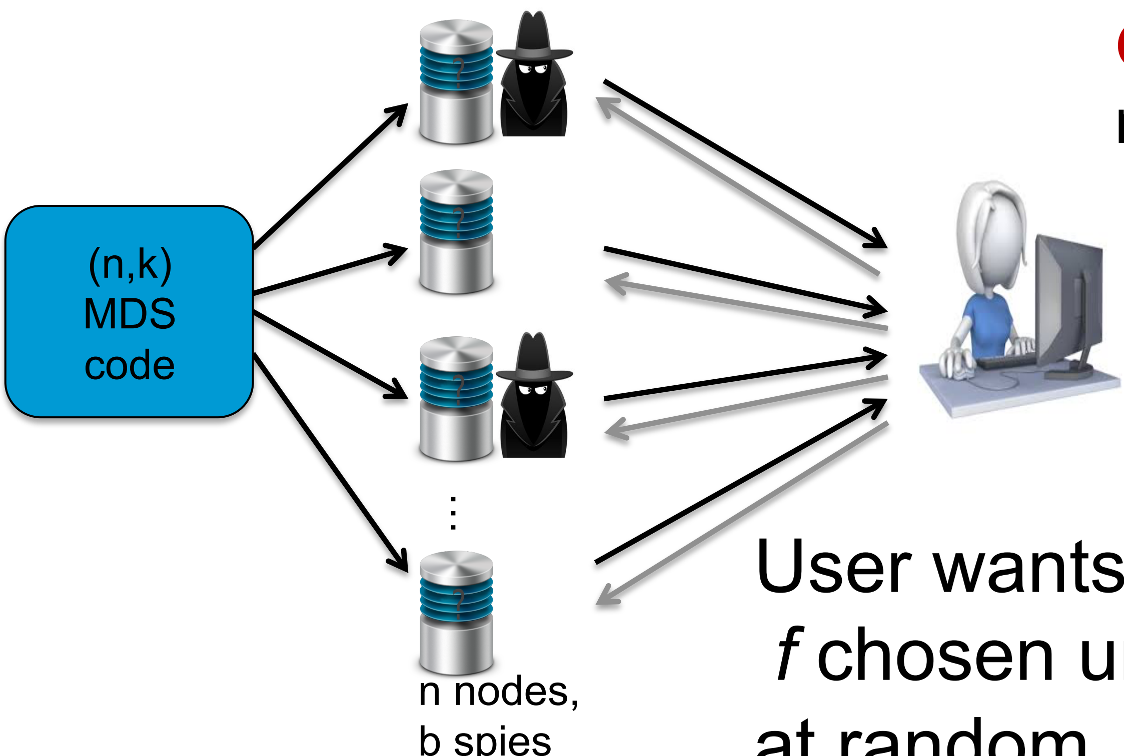
- What are the fundamental limits on the **possible tradeoffs** in this multi-dimensional space ?

- How to construct codes that achieve these fundamental limits?



System Model: Separation Approach

- A distributed system with n servers storing files $X_1, \ldots, X_m$
- b passive spy nodes
- Use "best" codes that minimize storage overhead for reliability, then optimize for privacy
- (n,k) MDS code is given and not design parameter.



Goal: Design PIR scheme with min download cost

Perfect privacy:

$$H(f|\text{queries to } b \text{ servers}) = H(f)$$

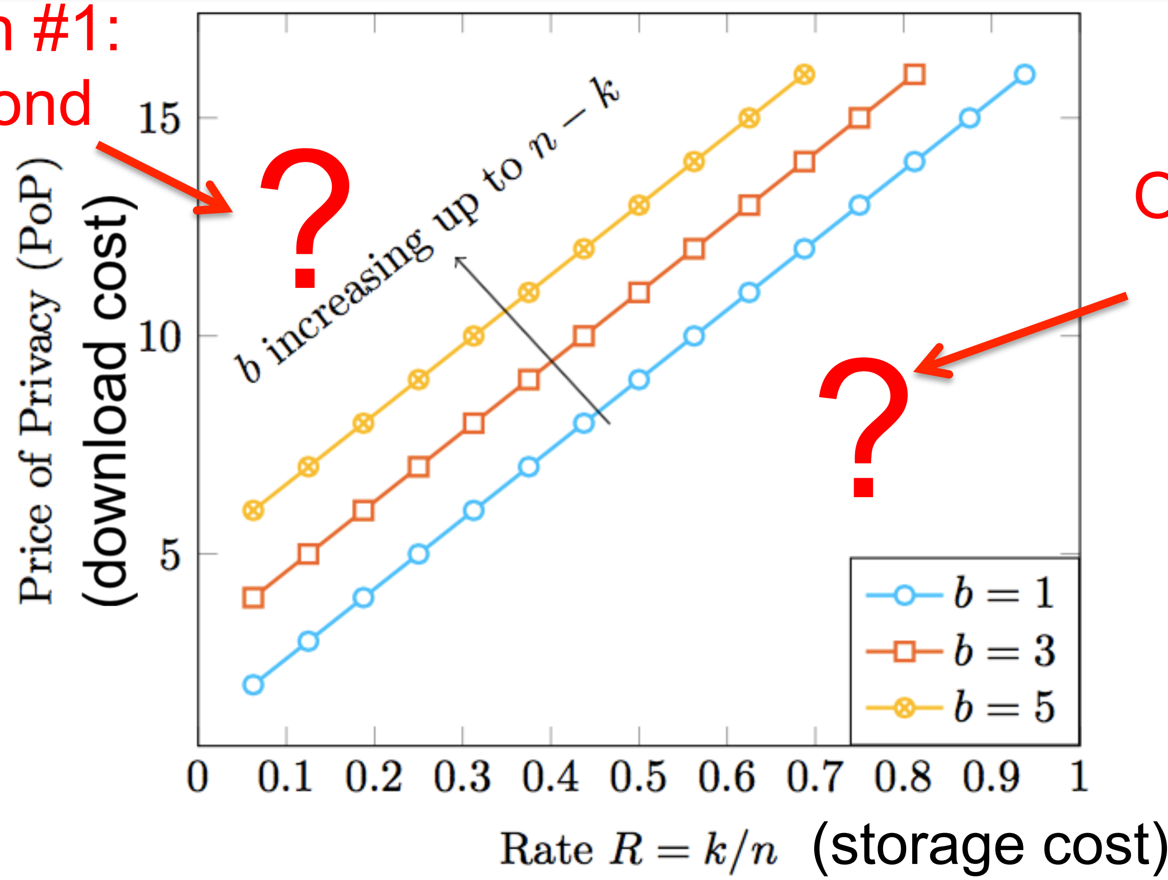
User wants X_f
f chosen uniformly at random

R. Tajeddine, S.E.R., "Private Information Retrieval from MDS Coded Data", ISIT 2016

Our Results: PIR on Coded Data

Theorem 1:[Tajeddine & S.E.R. ISIT'16] Consider a DSS using an *(n,k)* MDS code with *b≤n-k* spy nodes. Then, there exist an explicit linear PIR scheme with communication Price of Privacy ***PoP=b+k***

Open question #1: Schemes beyond b>n-k spies?



Open question #2: Fundamental limits?

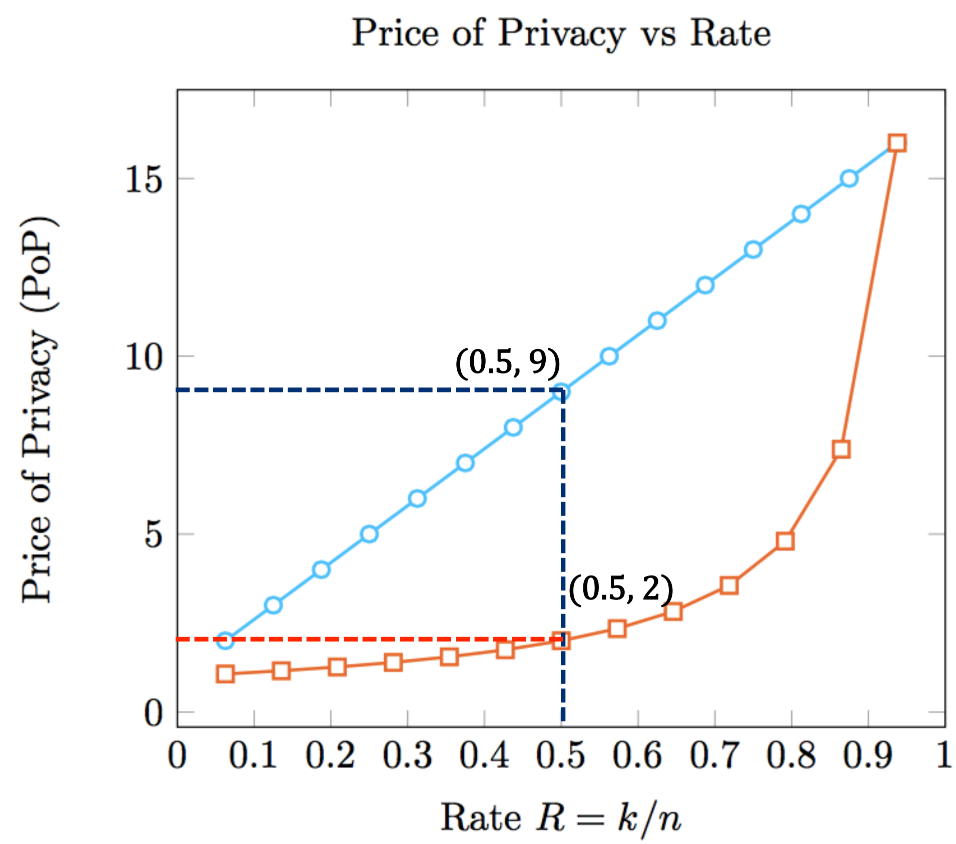
R. Tajeddine, S. E. R., "Private Information Retrieval from MDS Coded Data", ISIT 2016

IMPROVED PIR FOR SINGLE SPY

Theorem 2:[Tajeddine & S.E.R. ISIT'16] Consider a DSS using an *(n,k)* MDS code with *b=1* spy node. Then, there exist an explicit linear PIR scheme with communication Price of Privacy

$$PoP = \frac{n}{n-k} = \frac{1}{1-R}.$$

- Achieves the information theoretic bound given in [Chan et al, ISIT '15], [Banawan & Ulukus, Arxiv'16]
- Achieve the bound given in [Sun et. al, ISIT '16] when applying replication.*
- The PIR scheme is universal, i.e. does not depend on the MDS code.



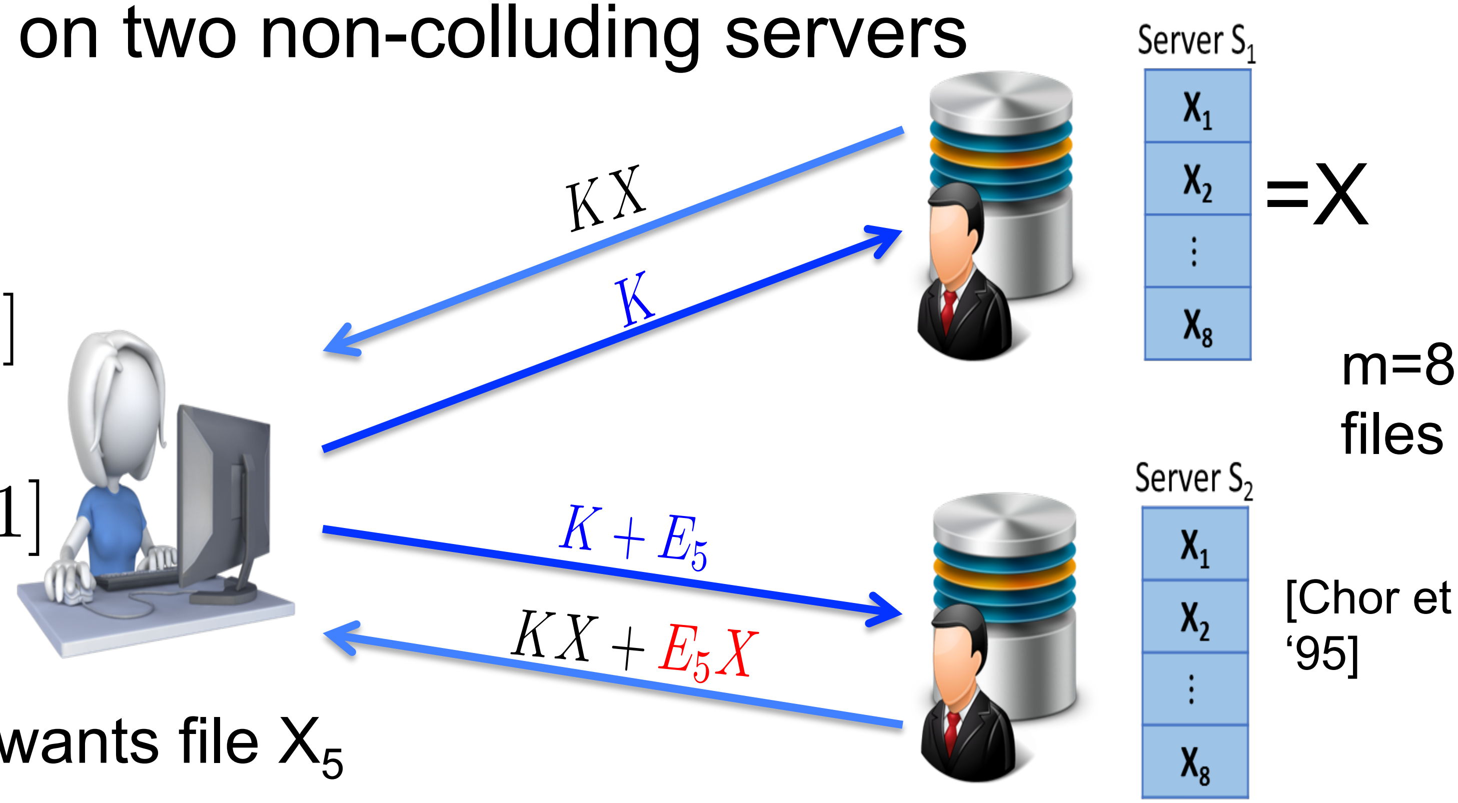
Efficient PIR: Toy Example

- Data replicated on two non-colluding servers

$$E_5 = [0\,0\,0\,0\,1\,0\,0\,0]$$

$$K = [1\,1\,0\,1\,0\,1\,0\,1]$$

random key



[Chor et al. '95]

Alice wants file X_5

- Downloads twice the file size. Price of Privacy PoP=2
- Perfect privacy in information theoretic sense
- *How about total upload + download cost?* 2m+2*FileSize

"Classical" Replication-Based PIR

- Focus has been on upload+download communication cost
- Early results, $O(m^{1/2n-1})$, m files replicated on n servers
  [chor et al. '95], [Ambainis, '97],…
- Holy grail: subpolynomial com.cost. [Yekhanin '08], [Efremenko '12], [Beimel et al. '06]
- $m^{O(\sqrt{\log\log m / \log m})}$ [Dvir and Gopi '14]

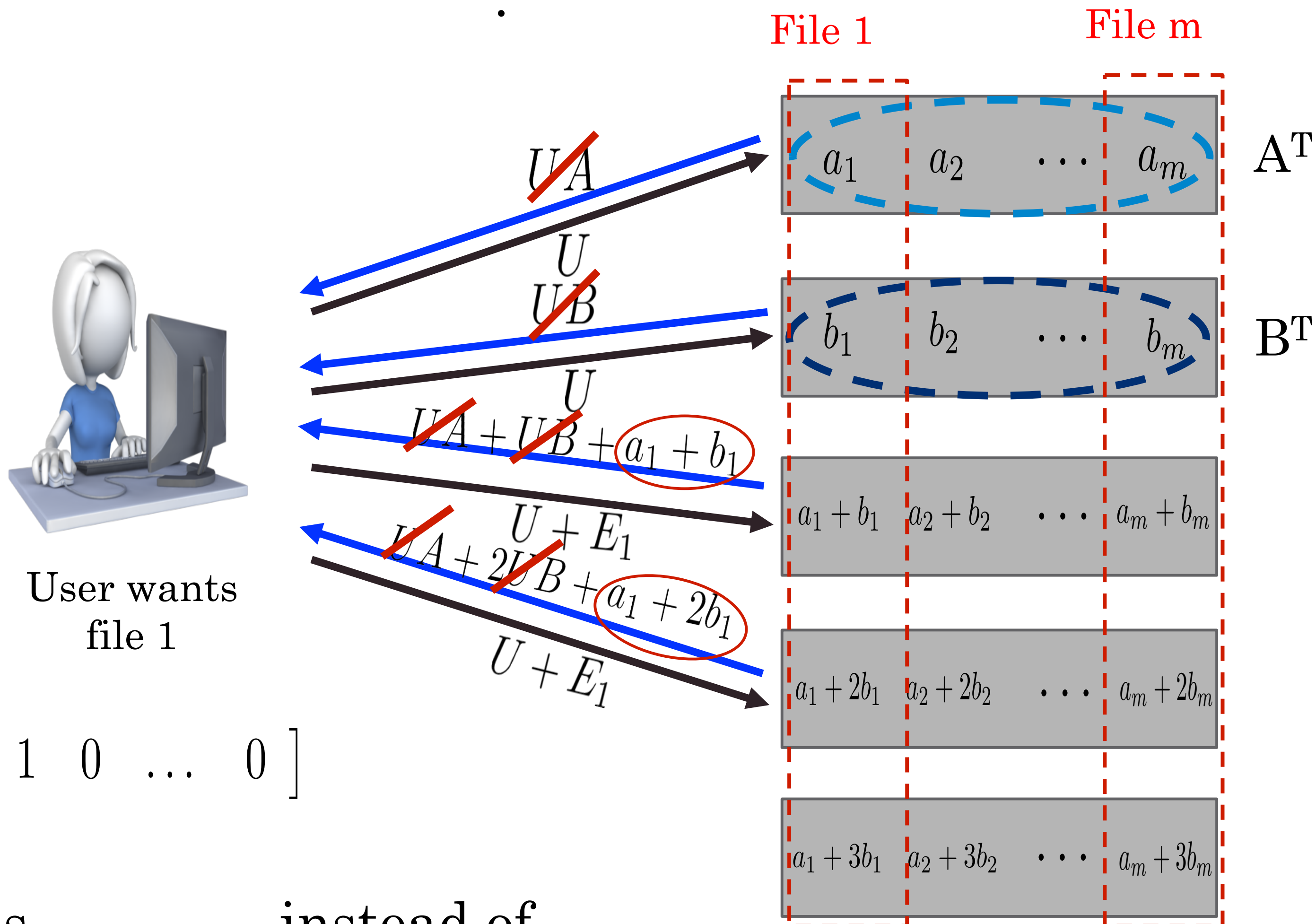
Requires replication ➔ high storage cost

Computational PIR

- No replication. Single server.
- [Kushilevitz and Ostrovsky, '97], [Chor and Gilboa, '97], [Cachin, Micali, and Stadler, '99], …
- High computational complexity [Sion and Carbunar, '07]

Theorem 2: b=1 spy, optimal scheme

- Generate an iid random vector $U = \begin{bmatrix} u_1 & u_2 & \dots & u_m \end{bmatrix}$.

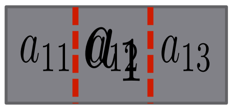


$$E_1 = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}$$

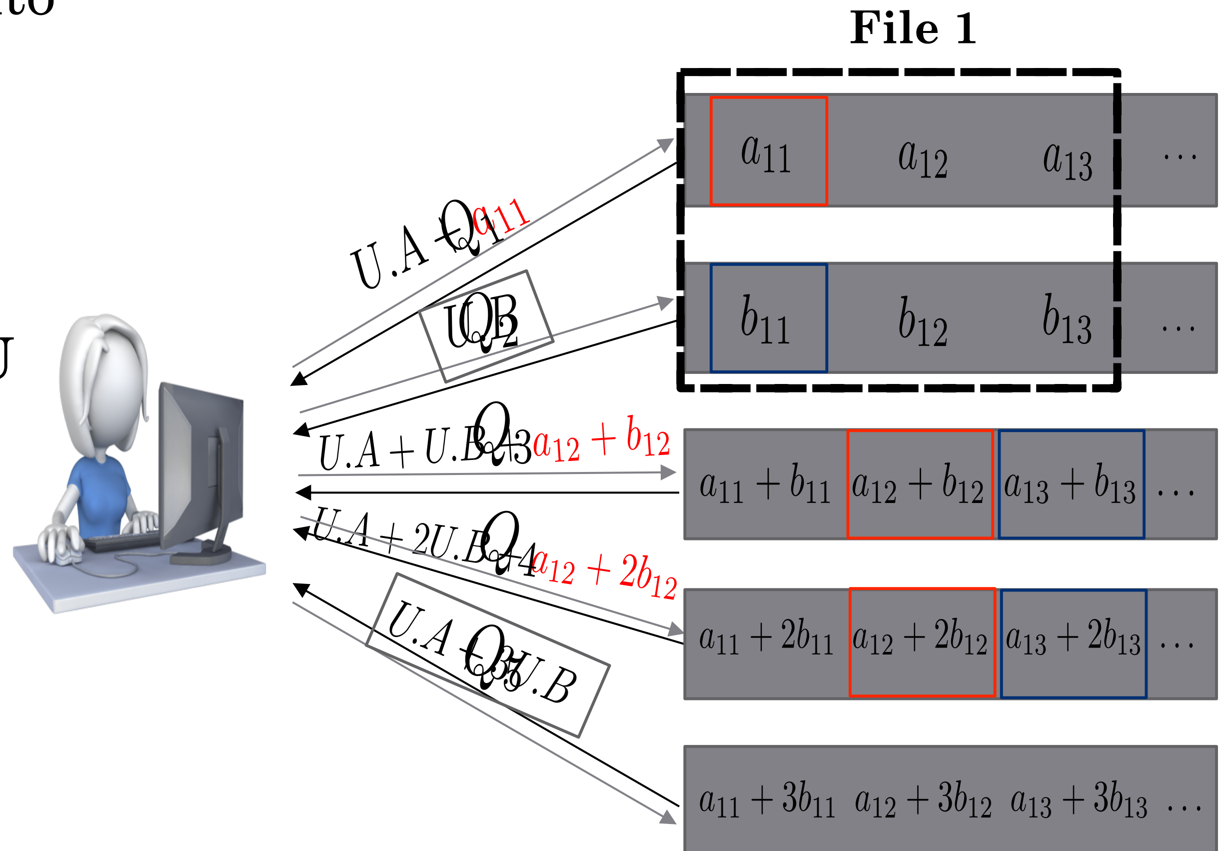
- This achieves $PoP = 2$ instead of $PoP = \frac{5}{3}$.

Theorem 2: b=1 spy, optimal scheme

- Divide each part into 3,



- 2 subqueries.
- 2 random vectors U and V



$$Q_1 = \begin{bmatrix} u_1+1 & u_2 & u_3 & \dots \\ v_1 & v_2 & v_3 & \dots \end{bmatrix}$$

$$Q_2 = \begin{bmatrix} u_1 & u_2 & u_3 & \dots \\ v_1+1 & v_2 & v_3 & \dots \end{bmatrix}$$

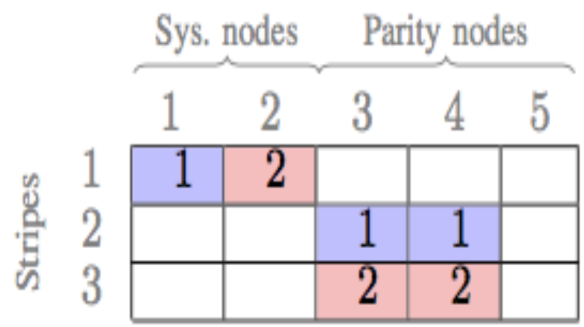
$$Q_3 = Q_4 = \begin{bmatrix} u_1 & u_2+1 & u_3 & \dots \\ v_1 & v_2 & v_3+1 & \dots \end{bmatrix}$$

$$Q_5 = \begin{bmatrix} u_1 & u_2 & u_3 & \dots \\ v_1 & v_2 & v_3 & \dots \end{bmatrix}$$

Remark for later: Alice needs the responses of all the servers.

Proof of Theorem 1

- Scheme:
  - We divide each file into $n-k$ stripes.
  - k sub-queries are made to each node (dimension of code is d).
  - We write $n-k=\beta k+r$.
- Conditions:
  - Decode $n-k$ parts in each sub-query.
  - Parts not on same node.
  - Different parts in each sub-query

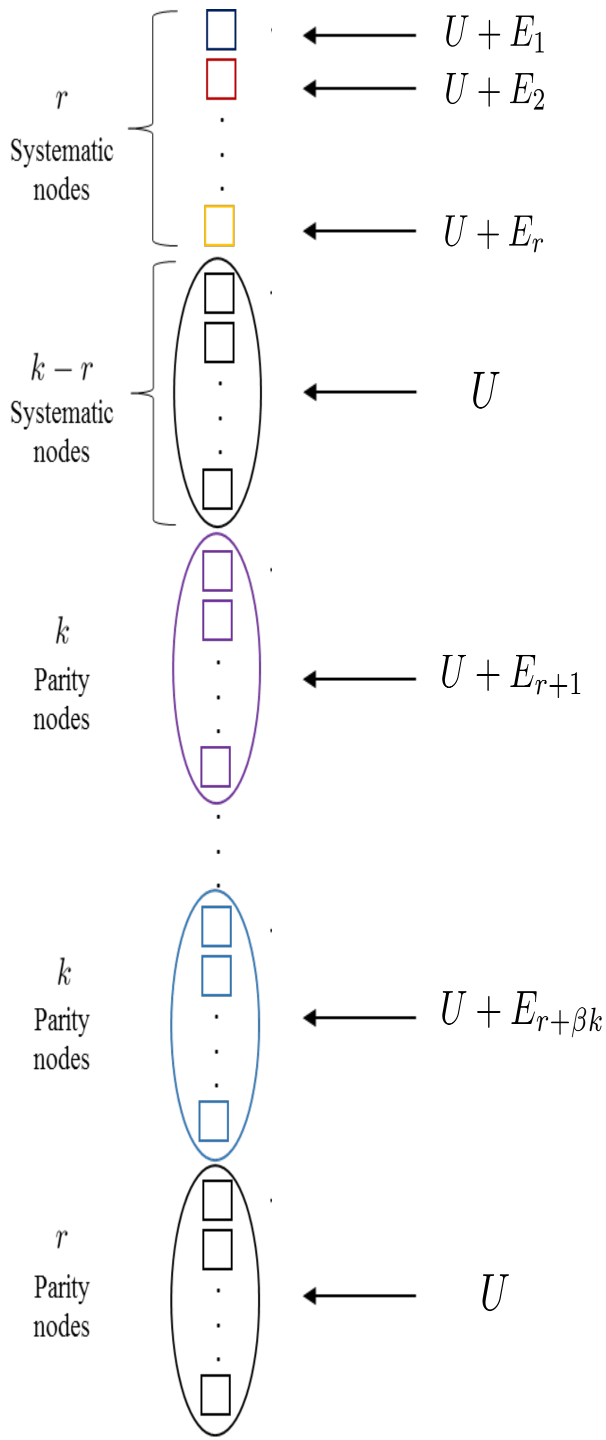


Retrieval pattern

Retrieval pattern for (15,4) MDS

		Sys. nodes				Parity nodes										
Stripes		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
r	1	1	2	3	4											
	2	2	3	4	1											
	3	3	4	1	2											
k	4					1	1	1	1							
	5					2	2	2	2							
	6					3	3	3	3							
	7					4	4	4	4							
k	8									1	1	1	1			
	9									2	2	2	2			
	10									3	3	3	3			
	11									4	4	4	4			
						k				k						

Querying



Where the E_i s are matrices with 1s at the positions we want to decode.

- k equations to decode interference.
- r equations from systematic nodes to decode parts of the first r stripes.
- βk equations from βk parity nodes to decode complete stripes.
- In total, $\beta k + r$ parts decoded.

Properties of the No-Collusion Scheme

- Universal: Does not depend on the MDS code
- Random vector can be just 0/1,i.e., projections are just XORS
- Instantaneous decoding in each sub-query
- Partial PIR of parts of the file
- Does not depend on number of files m
- Can be made Robust to non-responsive nodes (Reliability & Privacy) [Tajeddine & E.R. ISIT'17]

Theorem 1: Consider a DSS with n non-colluding nodes and using an (n, k) MDS code over $GF(q)$. Then, the linear PIR scheme over $GF(q)$ described in Section III is a universal ν-robust PIR scheme, i.e., it achieves perfect privacy and and has optimal $cPoP = \frac{n_i}{n_i - k}$, where $n_i = n - i$, for all number of unresponsive nodes $i, 0 \leq i \leq \nu$.

Example on Robust PIR scheme

	Layer 1	Layer 2			
		Node 1 is unresponsive	Node 2 is unresponsive	Node 3 is unresponsive	Node 4 is unresponsive
Node 1	$\mathbf{u}$	$\varnothing$	$\mathbf{v}$	$\mathbf{v}+\mathbf{e}_f$	$\mathbf{v}+\mathbf{e}_f$
Node 2	$\mathbf{u}$	$\mathbf{v}$	$\varnothing$	$\mathbf{v}$	$\mathbf{v}$
Node 3	$\mathbf{u}+\mathbf{e}_f$	$\mathbf{v}+\mathbf{u}$	$\mathbf{v}+\mathbf{u}$	$\varnothing$	$\mathbf{v}$
Node 4	$\mathbf{u}+\mathbf{e}_f$	$\mathbf{v}$	$\mathbf{v}$	$\mathbf{v}$	$\varnothing$

TABLE I: An example of our proposed 1-universal and adaptive robust PIR scheme. The scheme has two layers, with $\varnothing$ indicating the unresponsive node.

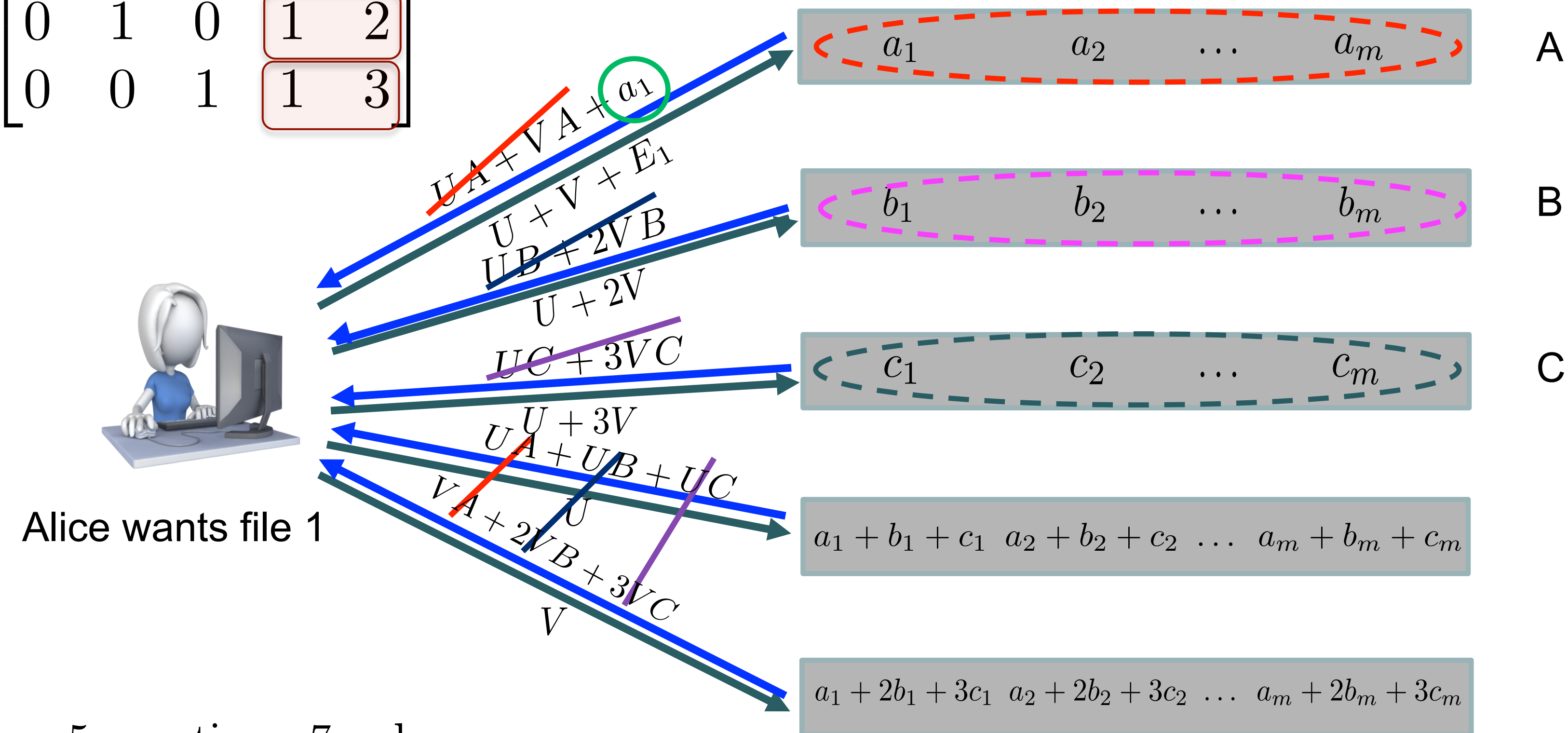
- The scheme is adaptive and universally optimal (achieves min PoP)

- **Open problem: non-adaptive robust PIR for coded data?**
- Later: non-adaptive robust PIR for uncoded data?

PIR Scheme for more than 1 spy

- User generates 2 iid random vectors U and V of length m (m number of files)

$$\Lambda = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 2 \\ 0 & 0 & 1 & 1 & 3 \end{bmatrix}$$



5 equations, 7 unknowns:
UA, VA, UB, VB, UC, VC, a_1

(5,3) code, b = 2 spies

- b=2 spy nodes
- Theorem 1 ➔PoP= b+k=5

Taste of the Proof Theorem 1

Theorem 1:[Tajeddine & E.R. ISIT'16] *b≤n-k* spies ➔ ***PoP=b+k***

- Generator matrix of the (n,k) MDS code

$$\Lambda = \left[\begin{array}{c|ccc} & & P & \\ I_{k\times k} & \lambda_{1,k+1} & \cdots & \lambda_{1,n} \\ & \vdots & \vdots & \vdots \\ & \lambda_{k,k+1} & \cdots & \lambda_{k,n} \end{array} \right]$$

- Parity check matrix generates the null space of the code

$$H = \left(\begin{array}{c|c} P^T & I \end{array} \right) \text{ with } \Lambda H = \mathbf{0}$$

- Step 1: Generate the random matrix

$$U = \begin{pmatrix} | & | & & | \\ U_1 & U_2 & \ldots & U_b \\ | & | & & | \end{pmatrix}$$

- Proof is in an extended version (with O. Gnilke) available on Arxiv Generalized constructions by [Freij-Hollanti et al. '16]
- More in Dave's talk
- Open problems: Fundamental bounds for coded & collusion?
- PIR schemes ind. of the code?

Taste of the Proof Theorem 1

- Step 2: Query phase

$$\begin{bmatrix} \text{---- Query to server 1-----} \\ \vdots \\ \text{---- Query to server n-----} \end{bmatrix} = Q = UH + E_f$$

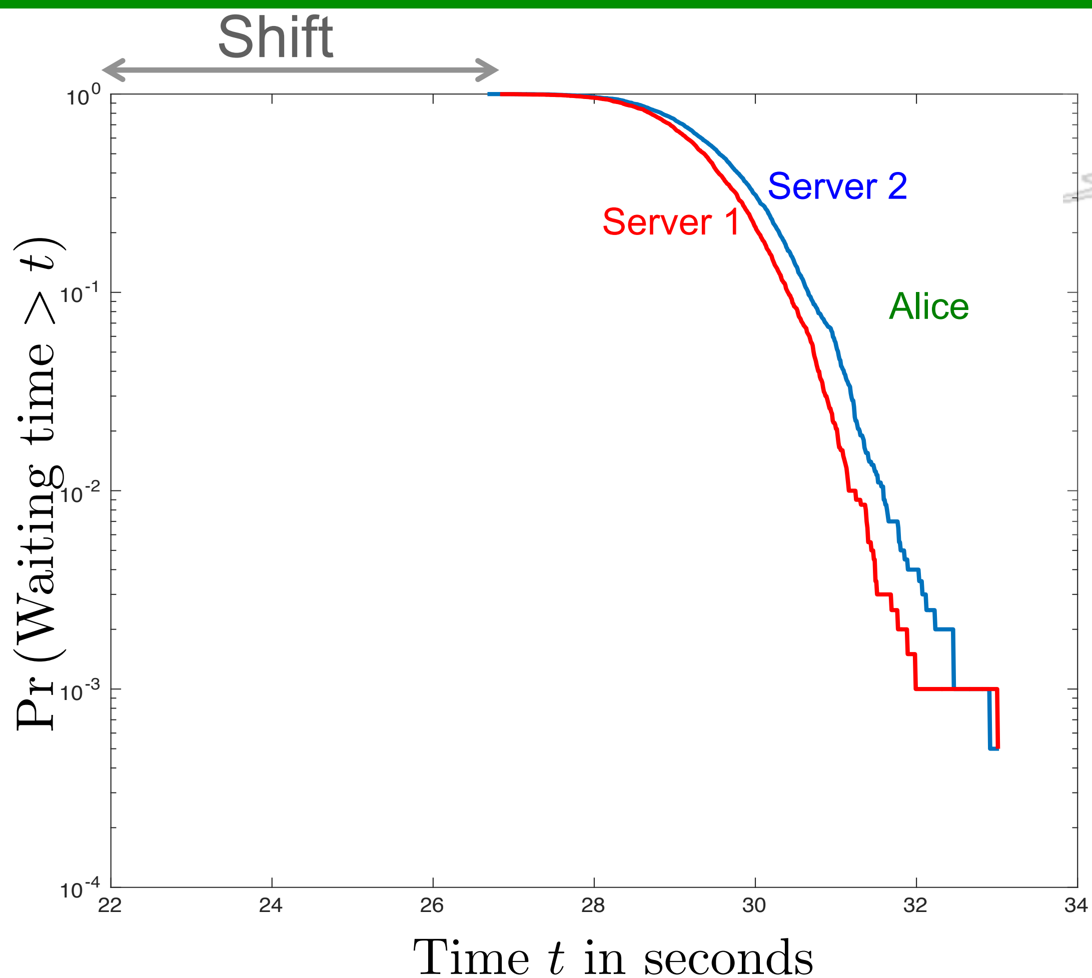
what Alice wants

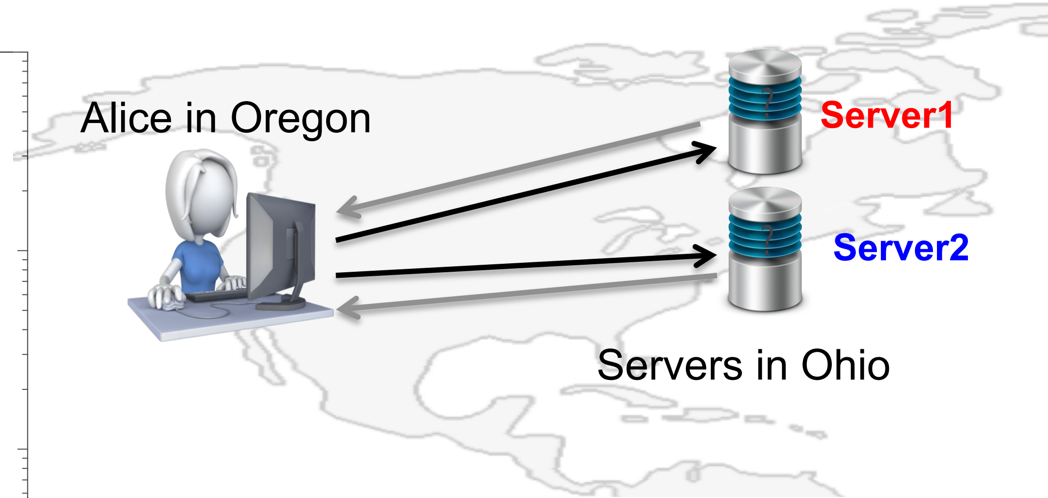
- Step 3: Response phase
- Each servers projects the query vector on its data and sends the result back to Alice
- Thus the response of all the nodes is:

$$R = \overbrace{UH\Lambda}^{=0} \mathcal{X} + E_f \Lambda \ \mathcal{X}$$

$$= E_f \Lambda \ \mathcal{X}$$

IMPLEMENTATION ON AMAZON CLOUD





File of 1 MB
Privacy level: one-out-1000
Average waiting time ≈ 29 s

Servers' response time shifted exponential $F(t) = 1 - e^{-\lambda(t-s)}$

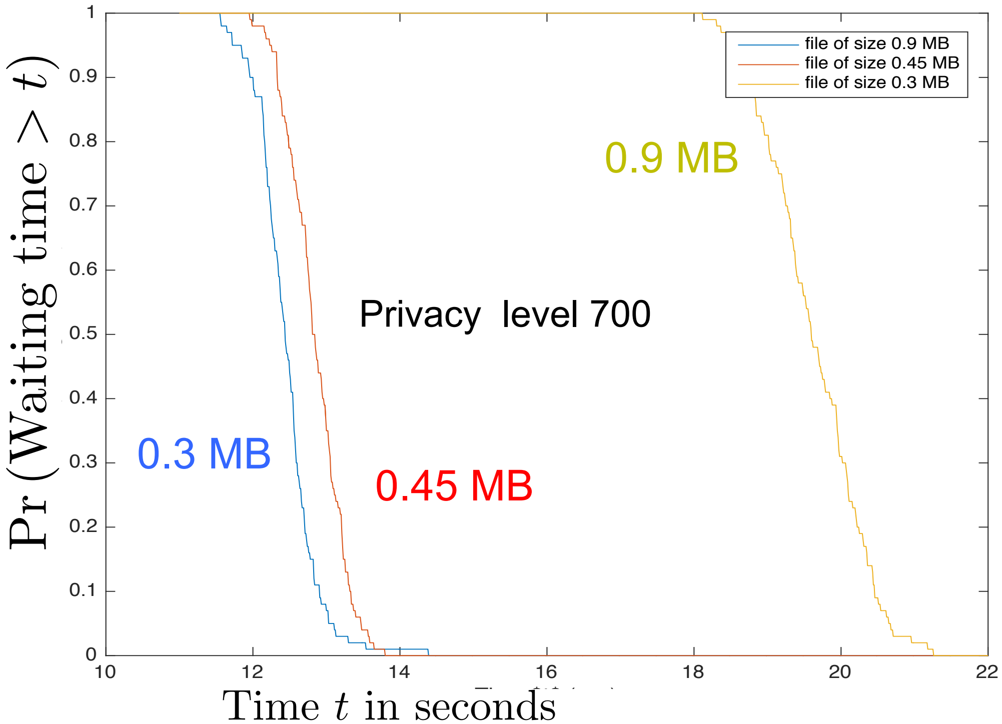
[Liang and Kozat, INFOCOM '14]

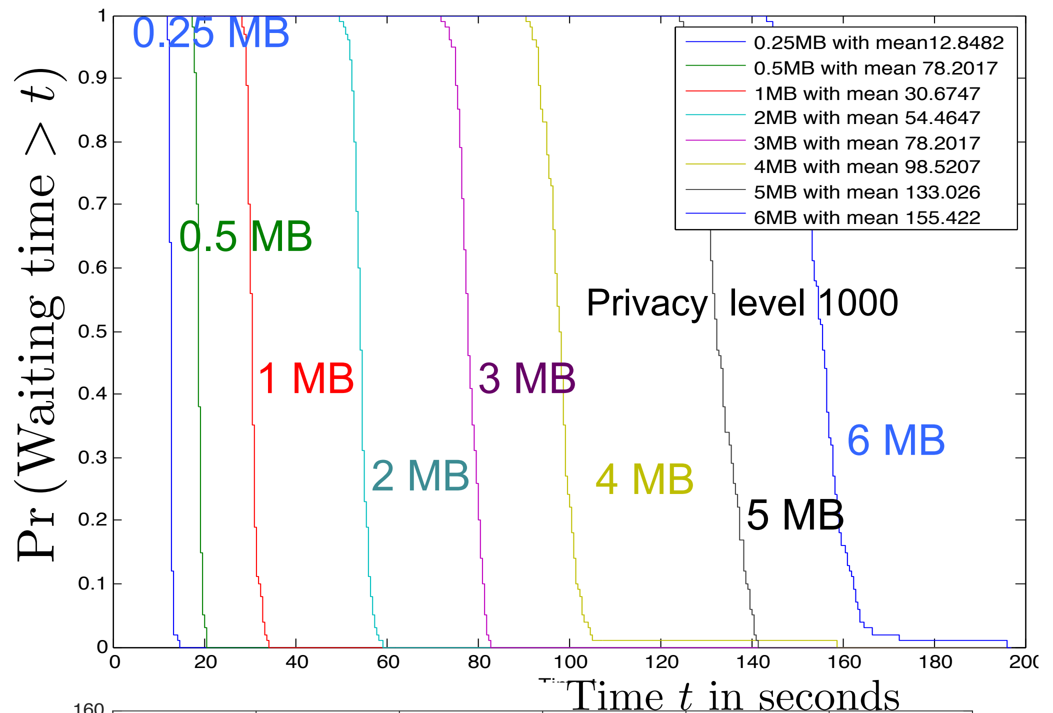
Two challenges:

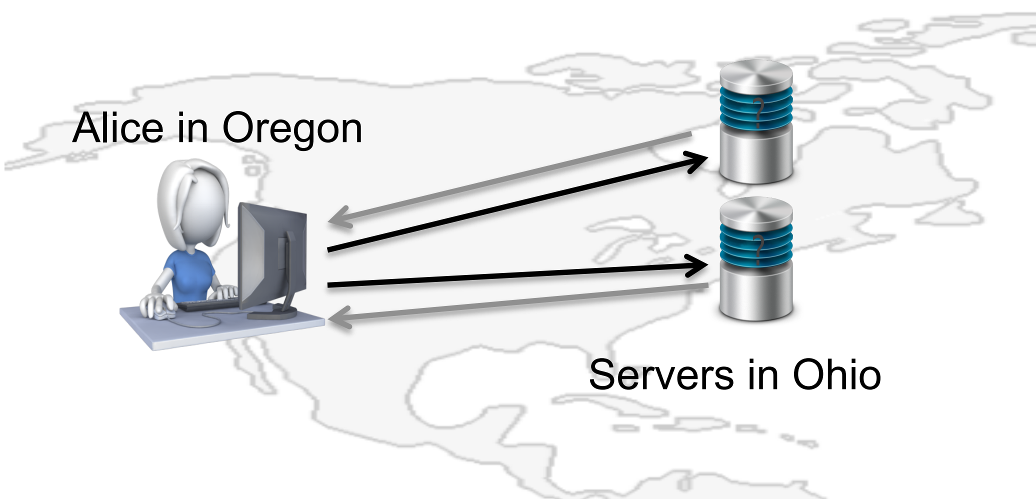
1. Straggler problem: Even one slow (straggler) server will delay Alice
   - The tail at scale effect [Dean and Barroso, ACM '13].
2. Computation overhead of PIR

Precomputations [Beimel et al. '00], Batch codes [Ishai et al, '04]

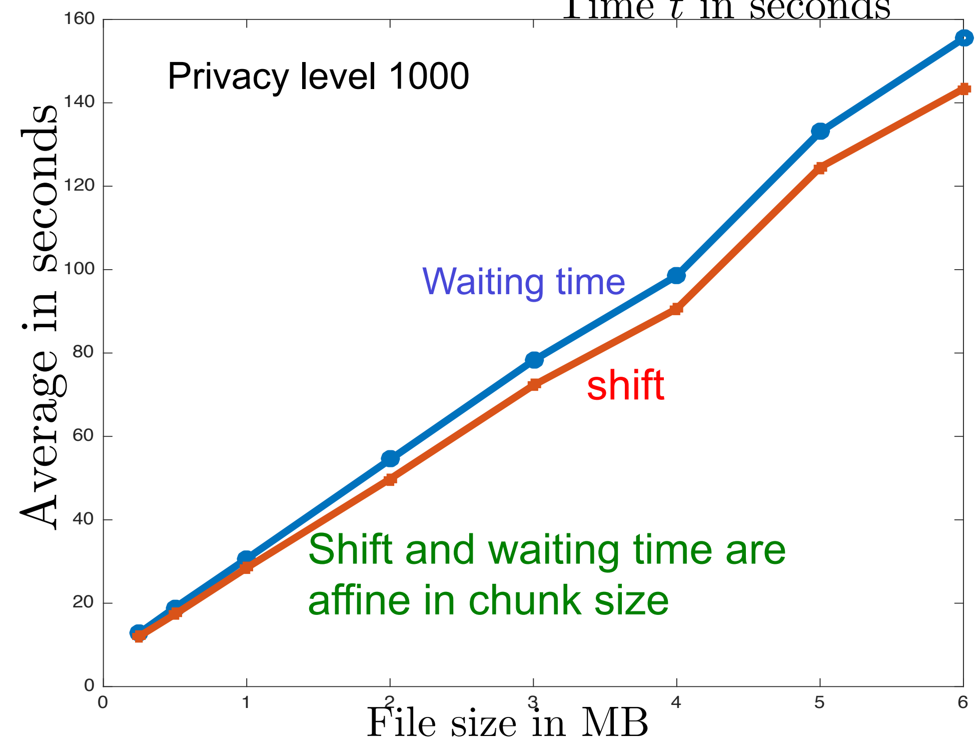
Effect of the file size



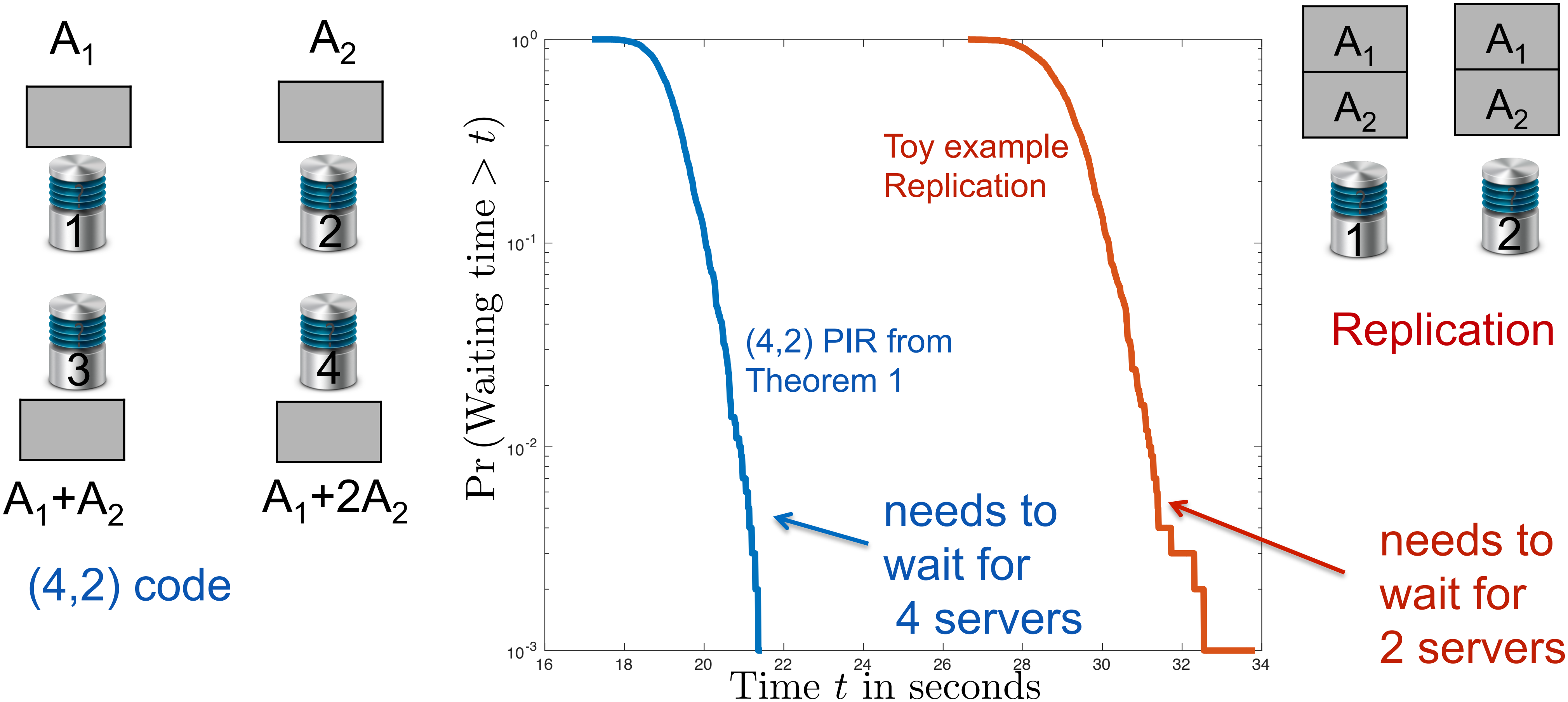




Emulations on AWS – EC2



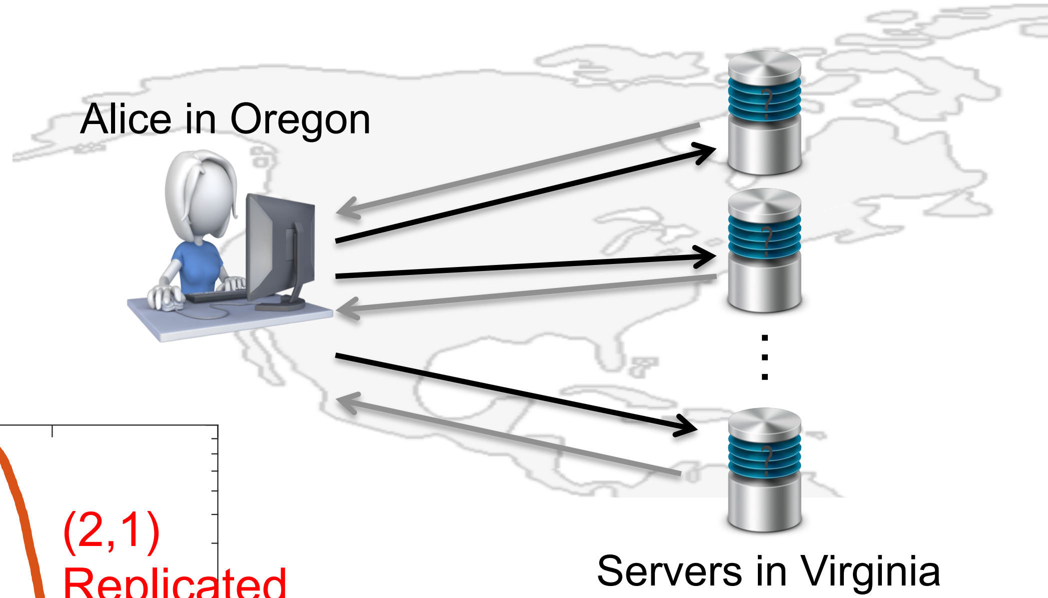
Codes Enable Parallelism

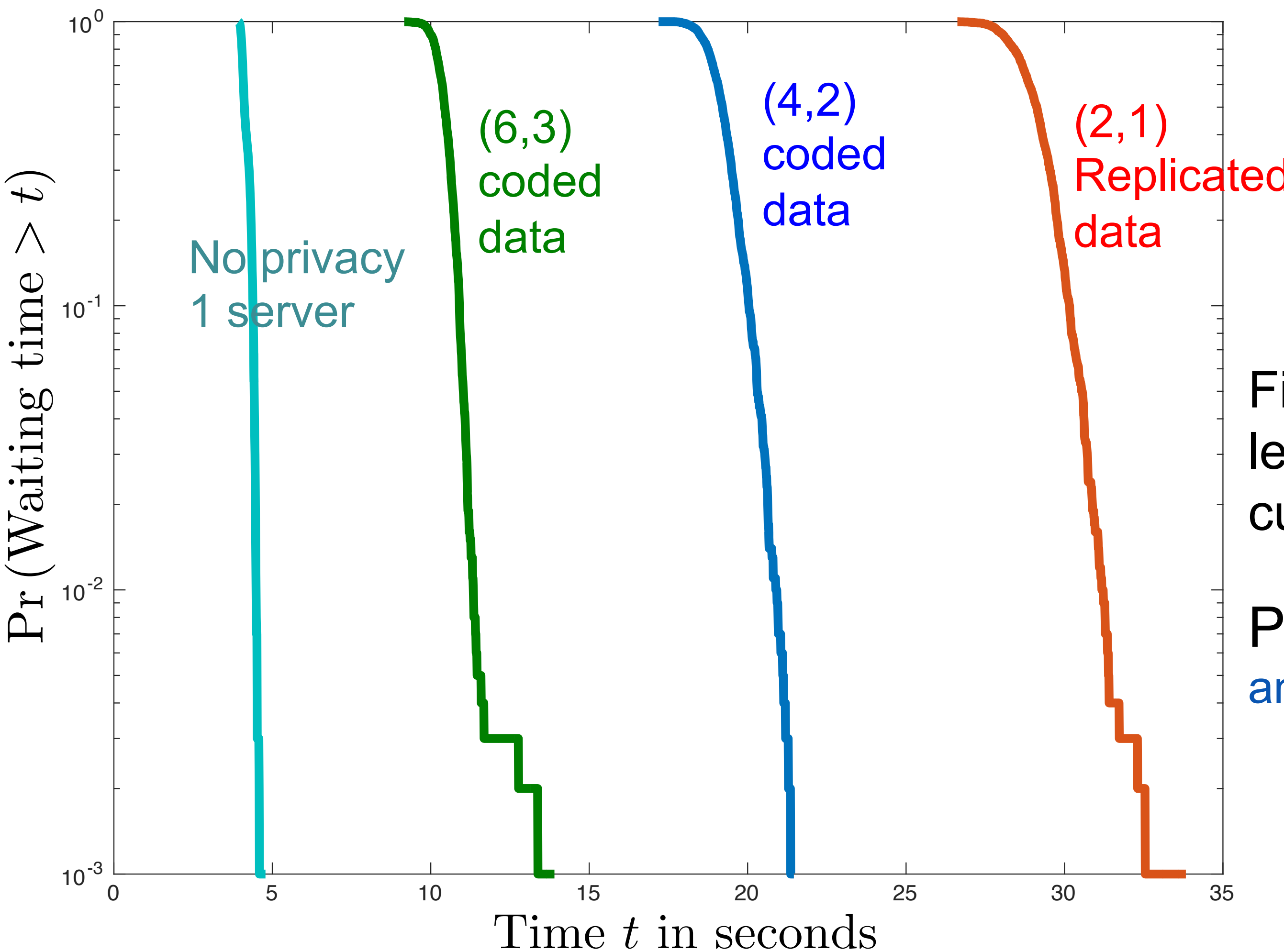


- Codes➔ chunking ➔ smaller waiting time per server
- But, more servers. Stragglers problem again

CODES ENABLE PARALLELISM

simulations on AWS – EC2

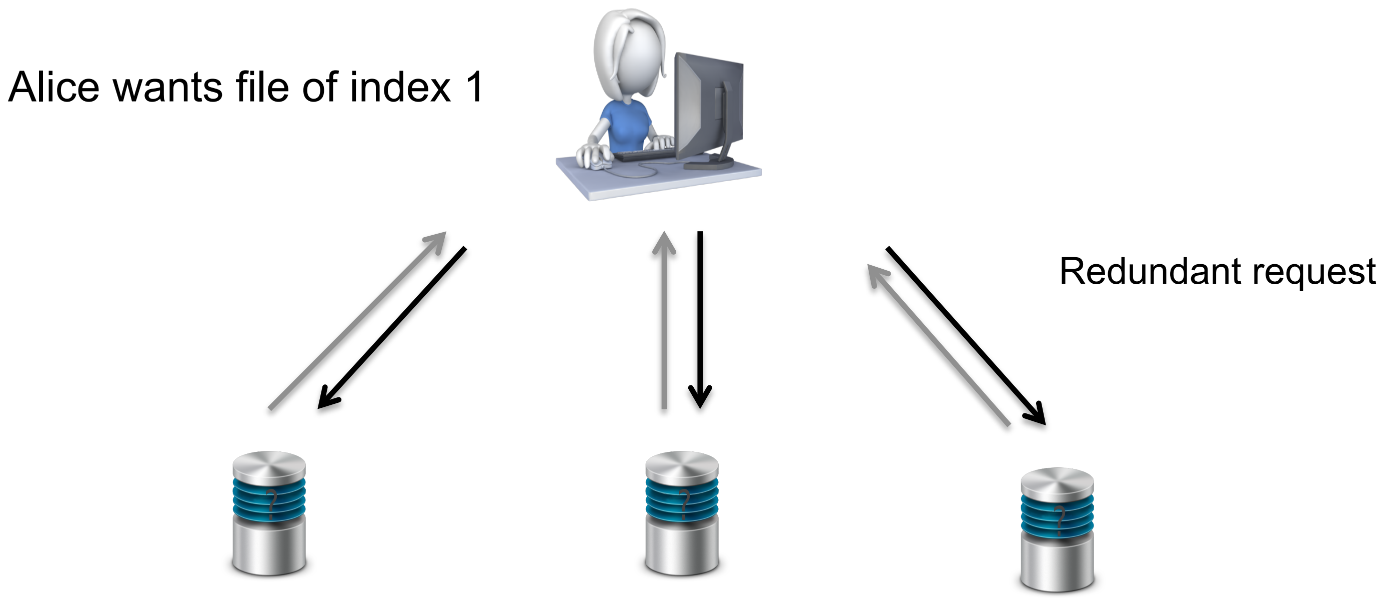




File of size 1 MB and privacy level of 1000 (for last three curves)

PIR schemes from [R. Tajeddine and **S.E.R.**, ISIT 2016]

PIR FOR STRAGGLERS

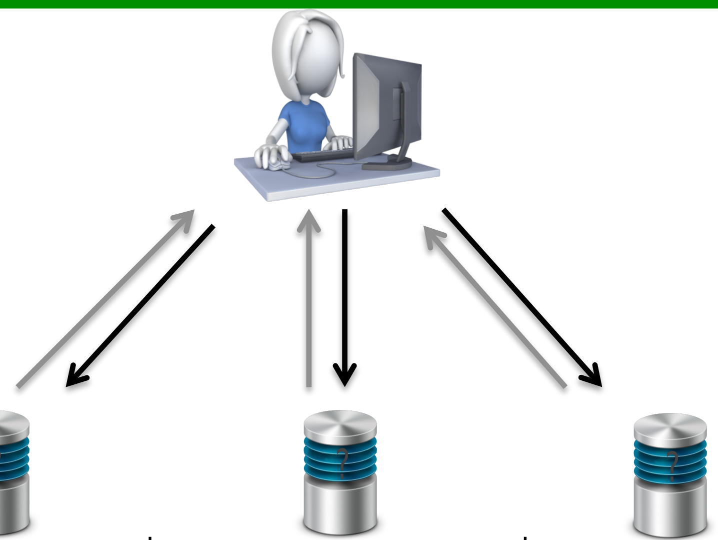


Server	1	2
Query	K	E_1+K

- Add redundancy to fight stragglers
- Idea used in the "non-privacy world" to speed up downloads
- [Joshi, Liu and Soljanin, JASC '14], [Lee, Lam, Pedarsani, Papailiopoulos and Ramchandran '16], [Shah, Lee and Ramchandran, '16], [Joshi, Soljanin and Wornell '15], ….

- What if no stragglers. We are wasting one server!

Universal PIR for any Number of Stragglers



- Files divided into 2 parts
- Queries divided into 2 subqueries
  - Avoid designing for worst case

Server	1	2	3
Sub-query 1	K_1	E_1+K_1	E_2+K_1
Sub-query 2	K_2	E_2+K_2	E_1+2K_2

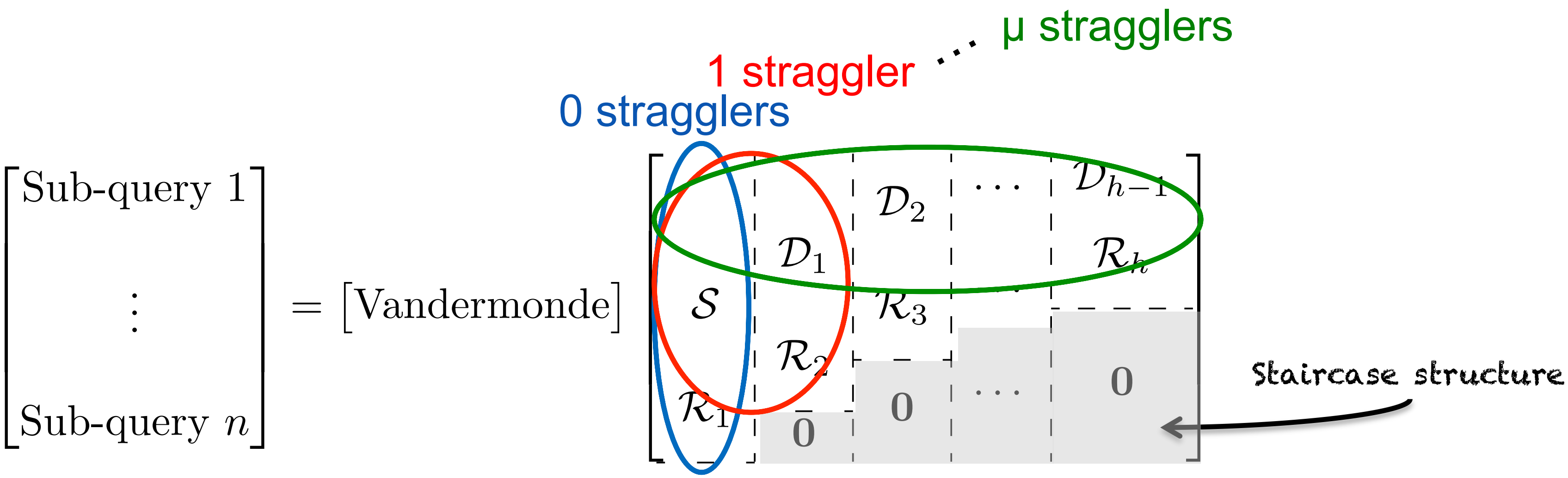
(3,2) Staircase PIR

- No straggler: Need responses of subquery 1. Achieves min PoP=3/2
- 1 straggler: Need full responses of any 2 servers. Achieves min PoP=2
- Connection to communication-efficient secret sharing

[R. Bitar and **S.E.R.**, "Staircase Codes for Secret Sharing with Optimal Communication and Read Overheads", ISIT 2016]

STAIRCASE PIR: GENERAL CONSTRUCTIONS

Theorem: [R. Bitar and **S.E.R.**, 2016] The μ-Universal Staircase PIR scheme constructed as follows in GF(q), q≥n, achieves minimum download cost for all number of responsive servers d, n-μ≤d≤n.

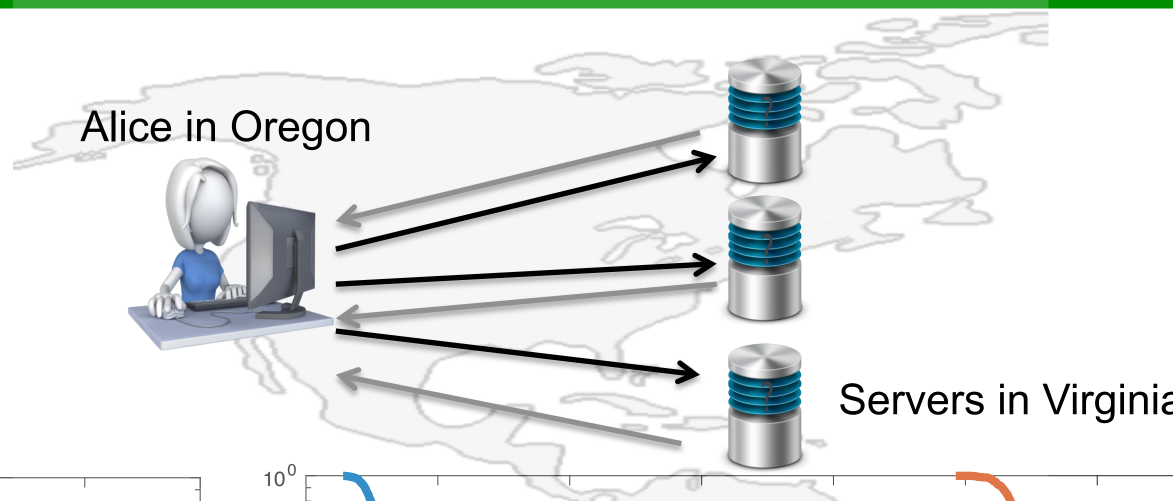


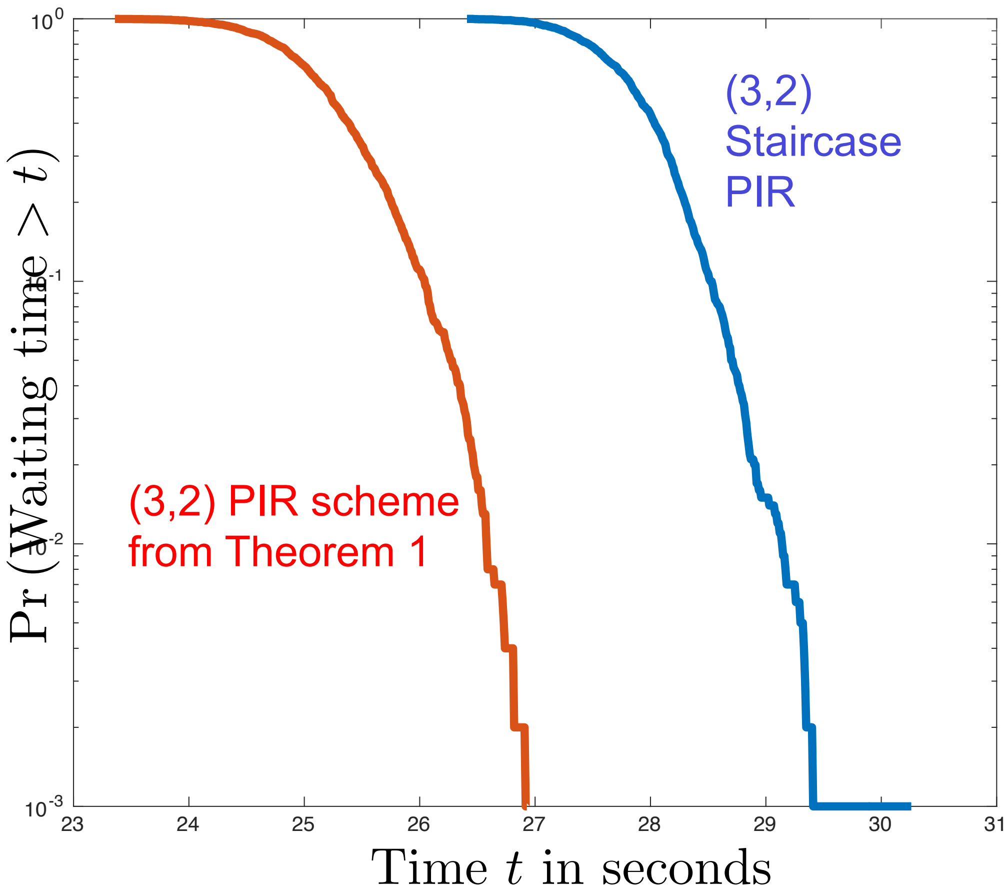
Encoding of the universal staircase code

[R. Bitar and **S. El Rouayheb**, "Staircase Codes for Secret Sharing with Optimal Communication and Read Overheads", ISIT 2016]

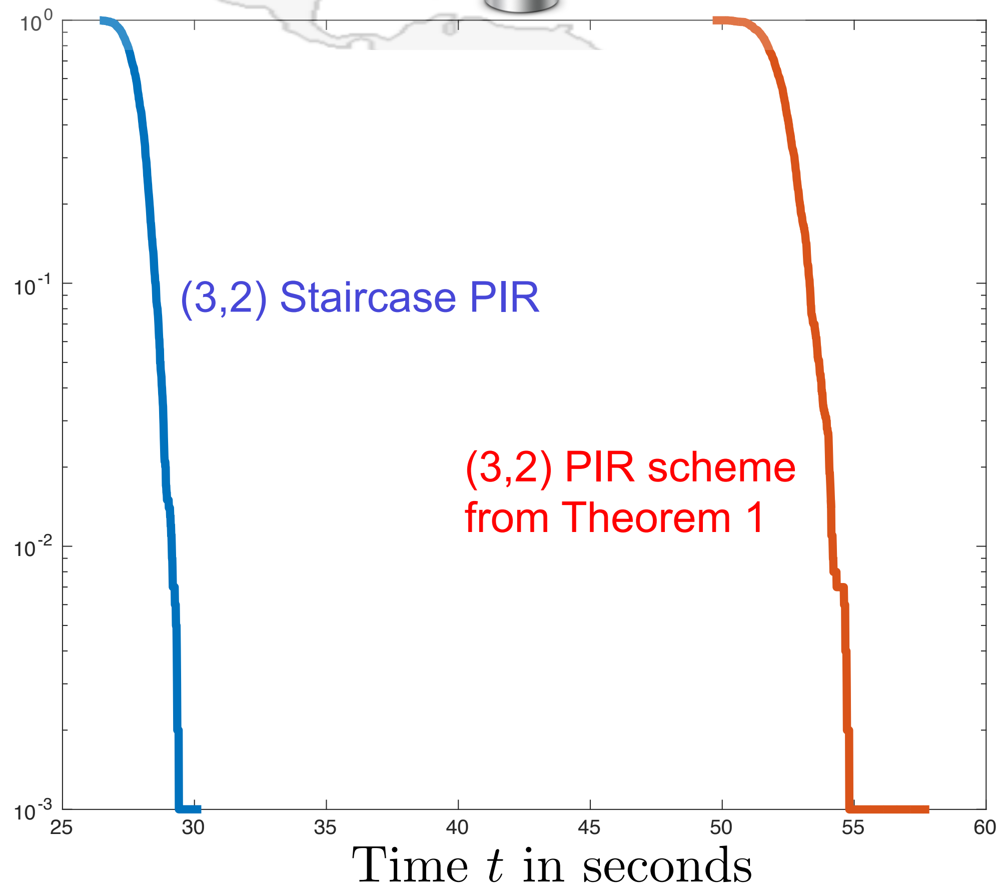
Latency Improvement by Staircase PIR

Emulations on AWS – EC2



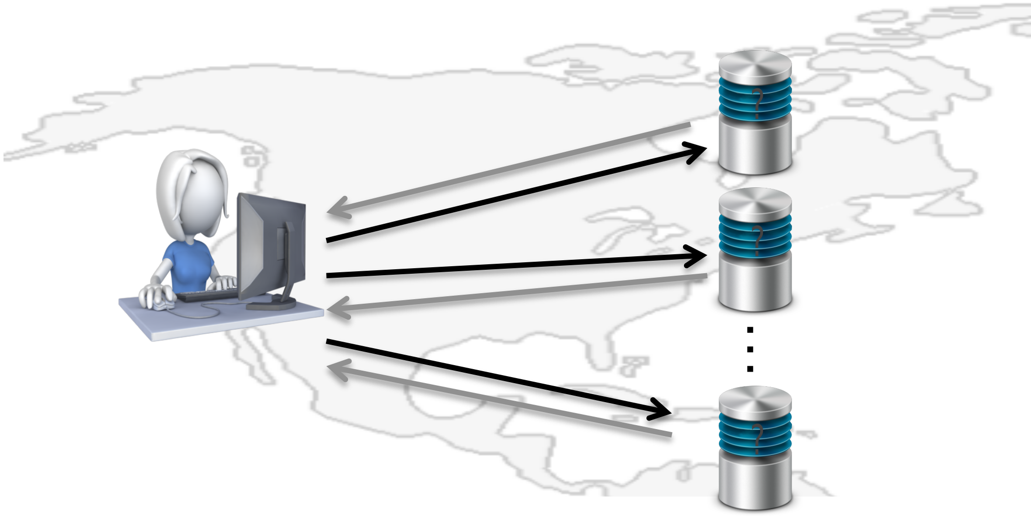


Requesting 1 file of size 1MB



Requesting 2 files of size 1MB

Decoding Options of Staircase PIR



n servers

Replicated data on n servers

Coded requests using (n,k) Staircase PIR [R. Bitar and **S.E.R.**, ISIT 2016]

System with b spies

d: number of fast servers

fast → slow

Decoding options:								Waiting time
No stragglers	1	2	3	4	5	$\cdots$	n	$\frac{k-b}{n-b}$
1 straggler	1	2	3	4	$\cdots$	d	n	$\frac{k-b}{d-b}$
2 stragglers	1	2	3	$\cdots$	d	n-1	n	$\frac{k-b}{d-b}$
$\vdots$								
n-k stragglers	1	$\cdots$	k	K+1	$\cdots$	n-1	n	1

The user waits until ***any*** decoding option is available

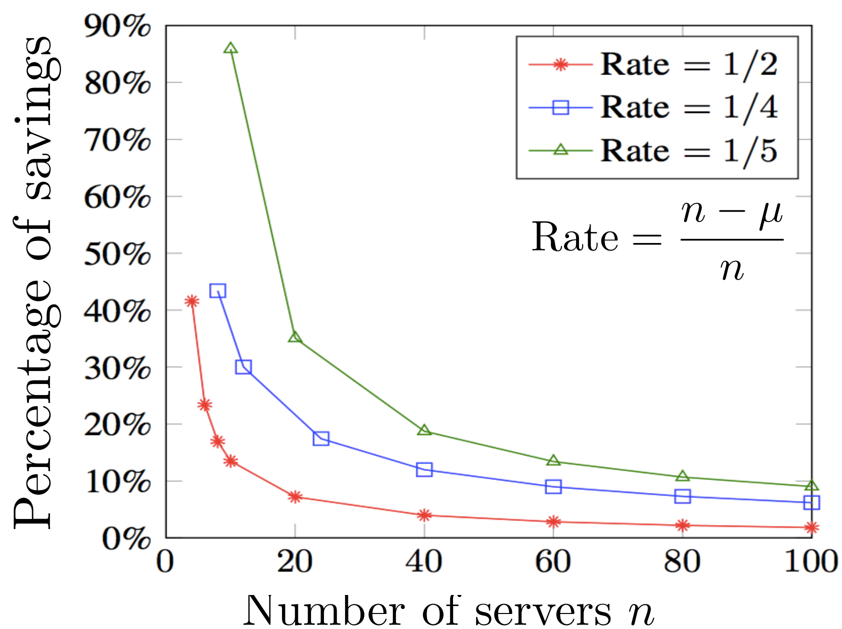
Efficiency of Coded Requests Over Replicated DBs

Theorem: [Bitar, Parag and **E.R.**, ISIT'17] Under an exponential distribution of the serving time assumed equally divided between subtasks, the mean waiting time $\mathbb{E}[T_{\text{SC}}]$ of an $(n, n-\mu)$ system using Staircase codes is upper bounded by

$$\mathbb{E}[T_{\text{SC}}] \leq \min_{d \in \{n-\mu,\ldots,n\}} \left(\frac{H_n - H_{n-d}}{\lambda(d-1)} \right), \qquad (1)$$

where H_n is the n^{th} harmonic sum defined as $H_n \triangleq \sum_{i=1}^{n} \frac{1}{i}$, and $H_0 \triangleq 0$.
The mean waiting time is lower bounded by

$$\mathbb{E}[T_{\text{SC}}] \geq \max_{d \in \{n-\mu,\ldots,n\}} \sum_{i=0}^{n-\mu-1} \binom{n}{i} \sum_{j=0}^{i} \binom{i}{j} \frac{2(-1)^j}{\lambda\left(n(n-1) + d(d-1) - 2(i-j)(d-1)\right)} \qquad (2)$$



Open question: Generalize to coded data.

[R. Bitar, P. Parag and **S.E.R.**, "Minimizing Latency for Secure Distributed Computing", ISIT 2017] (arxiv)

QUESTIONS?