

Parallelizable and Updatable Private Information Retrieval

By Boyan Litchov

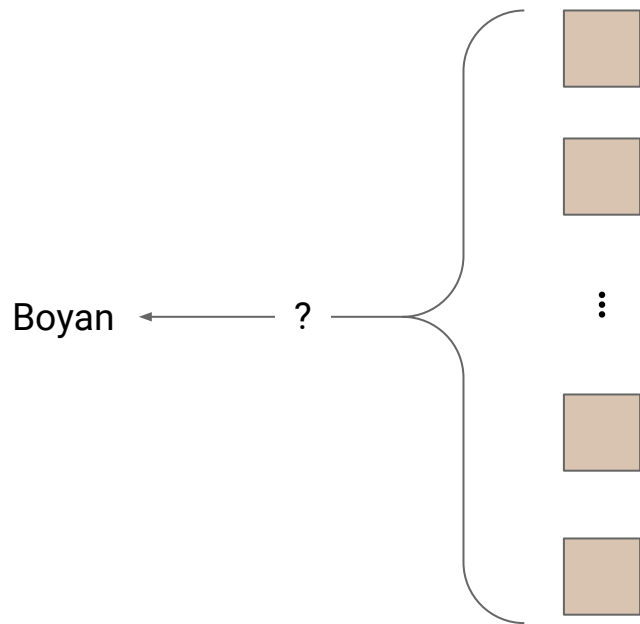
Mentored by Simon Langowski

A dark blue diagonal gradient bar that starts from the bottom left corner and extends towards the top right corner, covering the lower half of the slide.

Private Information Retrieval (PIR)



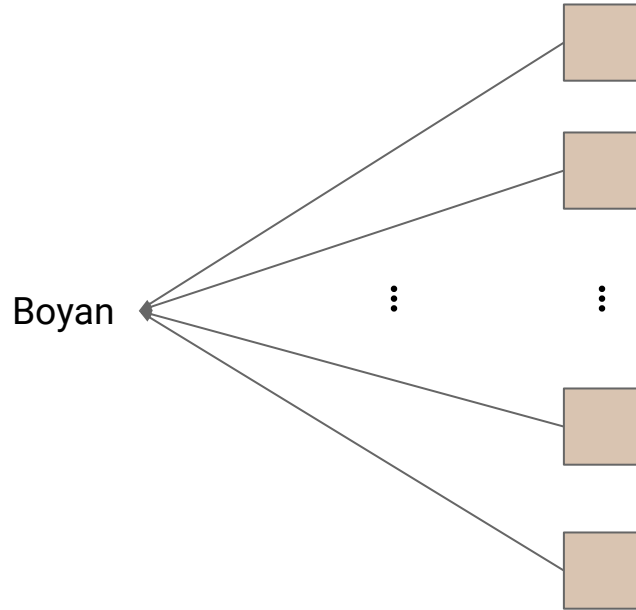
The Problem



Use Cases

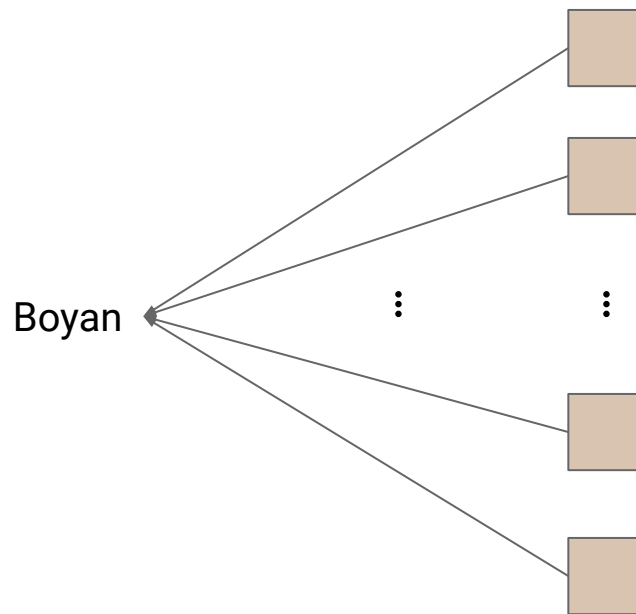
- Private Browsing
- Private Streaming
- Anonymous Messaging

A Simple Solution (1/2)



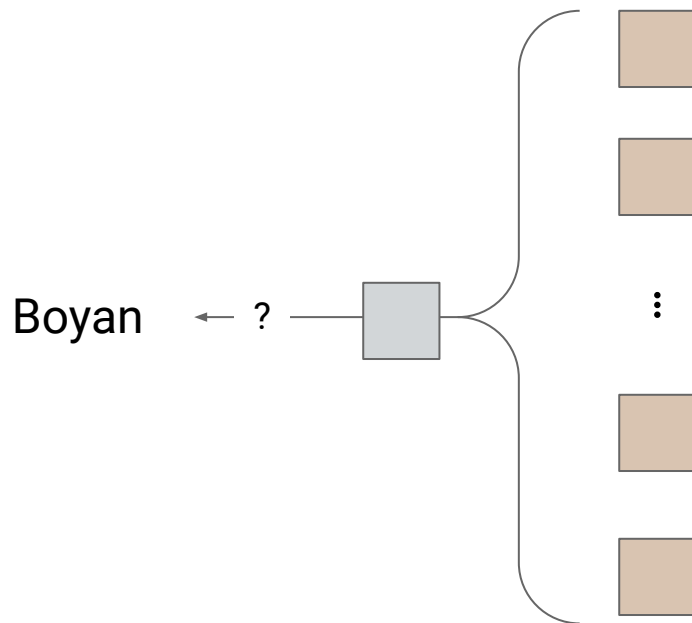
A Simple Solution (2/2)

- Network Costs are the entire database
 - Too high




The Goal

- Compress the database into one element
 - Minimizes network costs



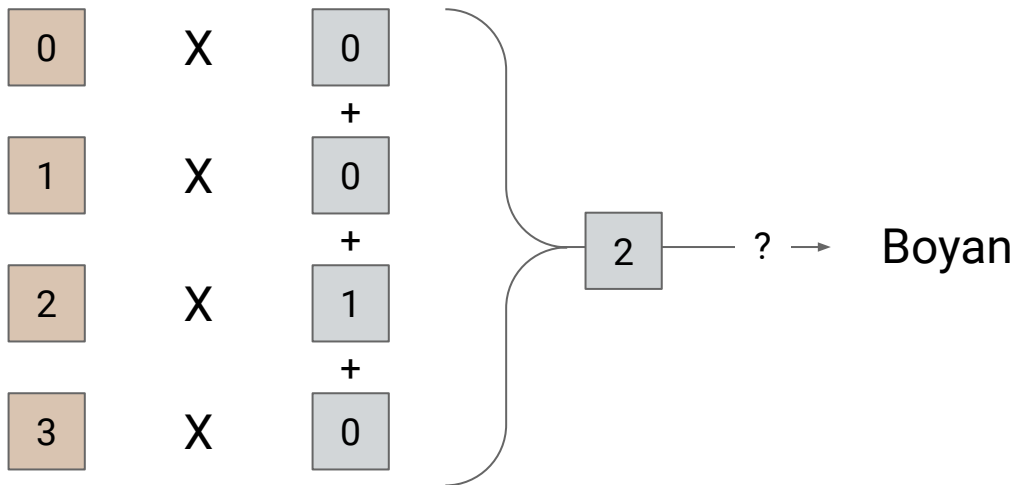
 Visible

 Encrypted

The Approach

Database

Query



Costs

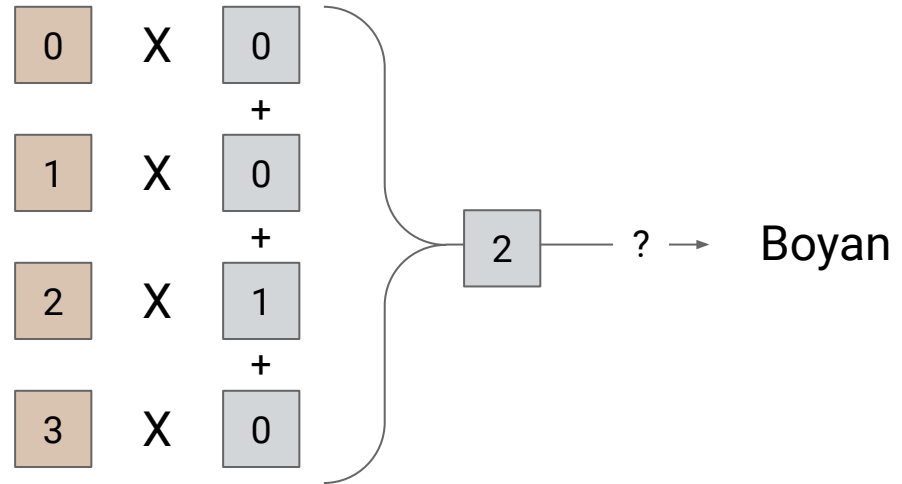
Network

- Query is as big as the database
- Response is 1 element

Computational

- n multiplications
- $n-1$ additions

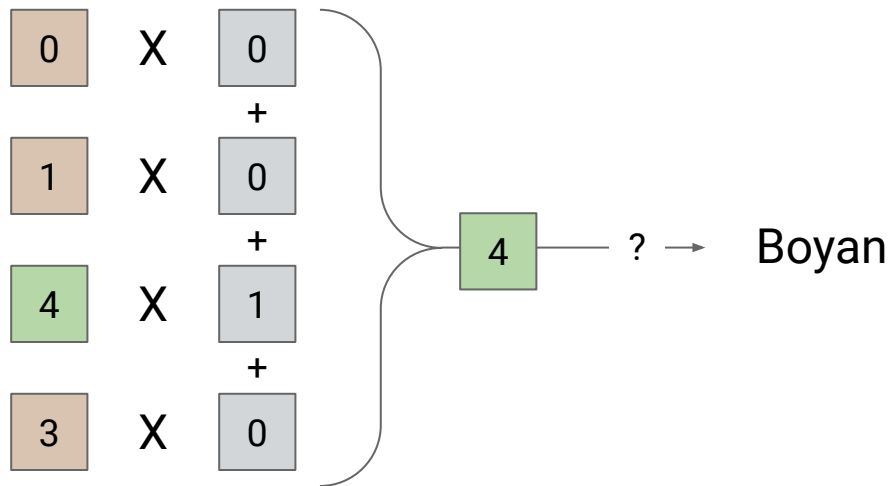
Database Query



Updatability (1/2)

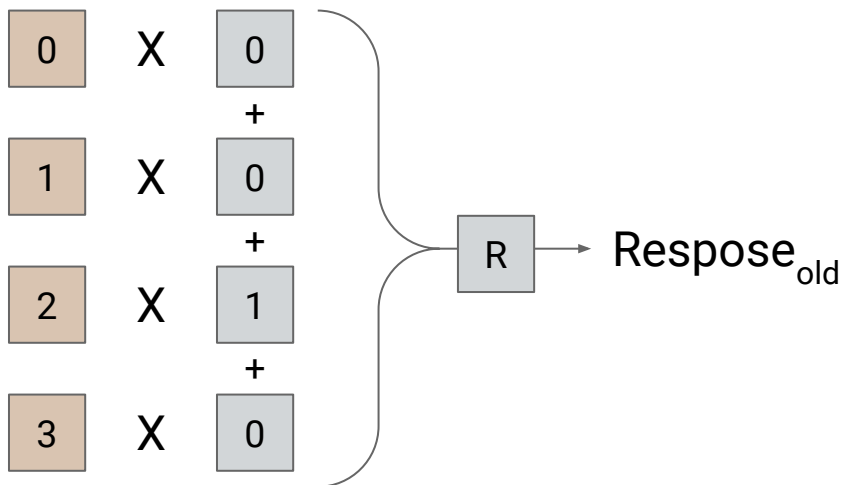
- If the database changes, the old response can be updated without computing on a large part of the database

Database Query

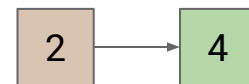


Updatability (2/2)

Database Query



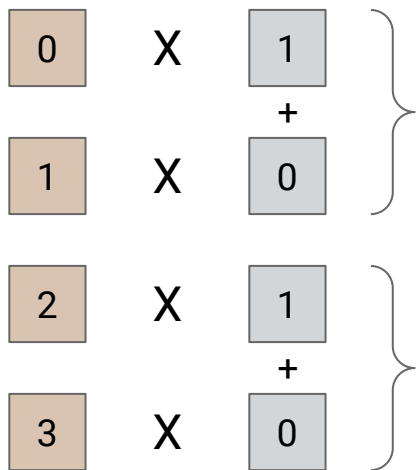
Update



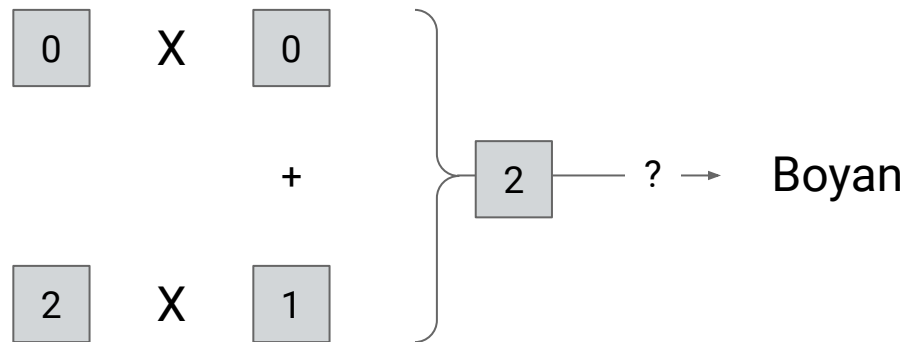
$$\begin{array}{r} \text{Response}_{\text{old}} - \begin{array}{c} \boxed{2} \quad \text{X} \quad \boxed{1} \\ \boxed{4} \quad \text{X} \quad \boxed{1} \end{array} \\ + \\ \hline = \text{Response}_{\text{new}} \end{array}$$

Folding (1/2)

Database



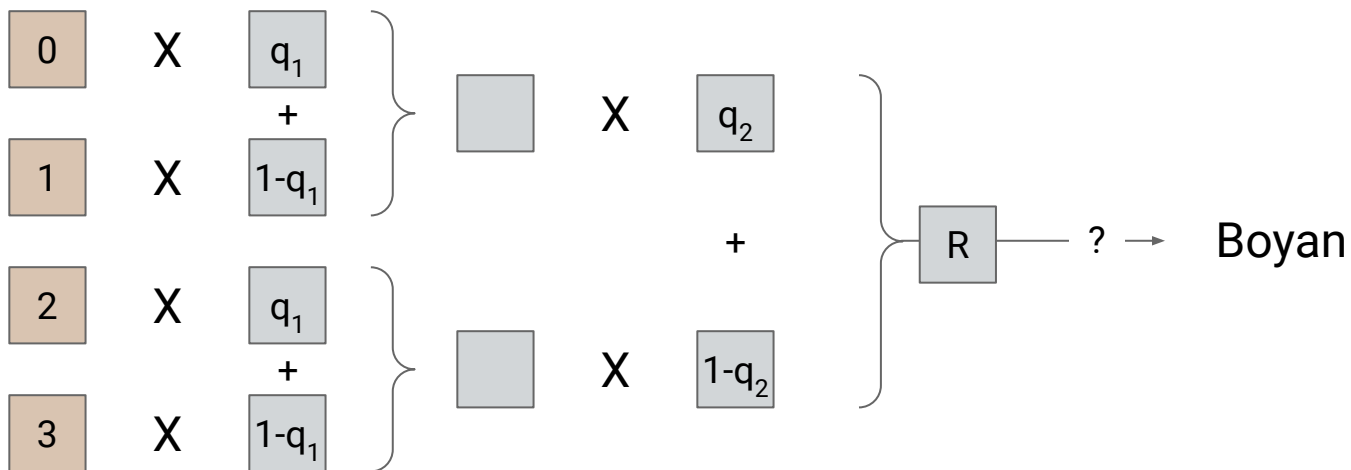
Query



Folding (2/2)

Database

Query



Costs

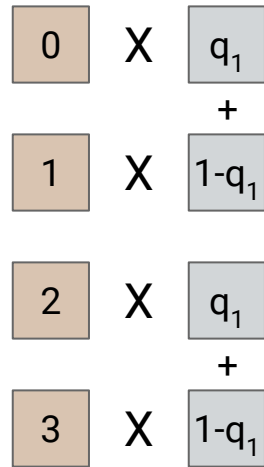
Network

- Query is $\log_2(n)$ elements
- Response is 1 element

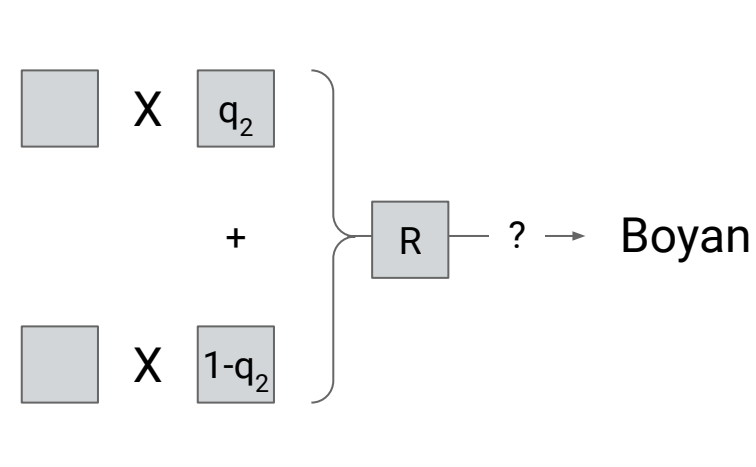
Computational

- $2n-1$ multiplications
- $n-1$ additions

Database



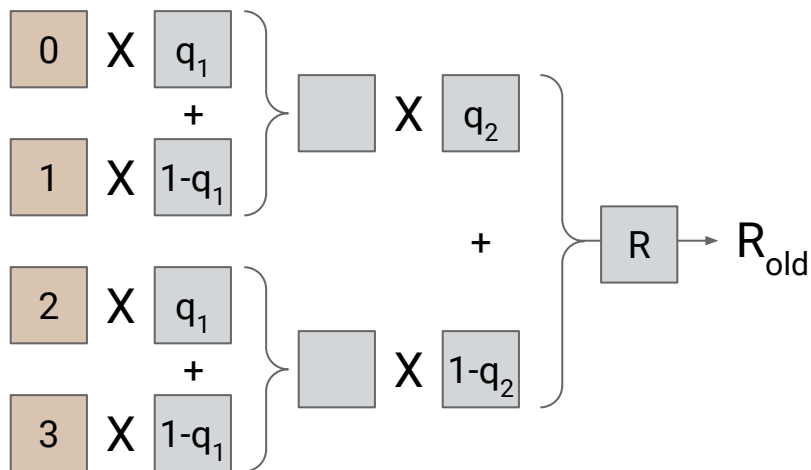
Query



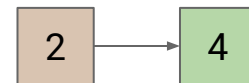
Updatability

Database

Query



Update



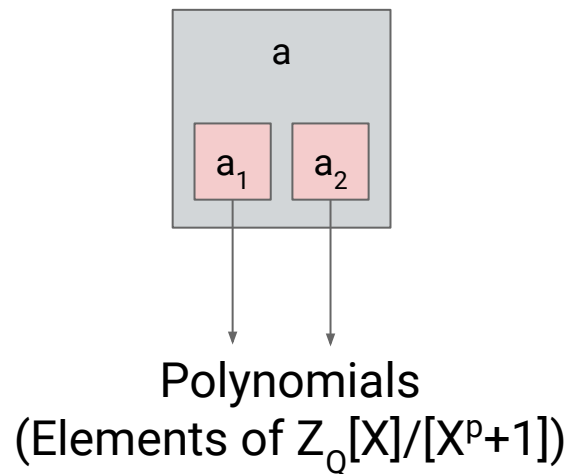
$$\begin{aligned} R_{old} & - 2 \times q_1 \times (1-q_2) \\ & + 4 \times q_1 \times (1-q_2) \\ \hline & = R_{new} \end{aligned}$$

Homomorphic Encryption & Number-Theoretic Transforms (NTTs)

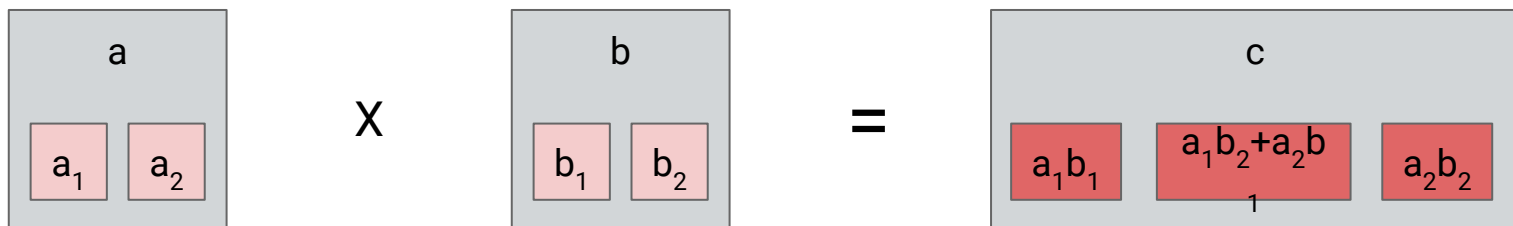



Fully Homomorphic Encryption Ciphertext

- Ciphertexts are noisy
- "Fresh" ciphertexts consist of two polynomials
 - Polynomial length of p



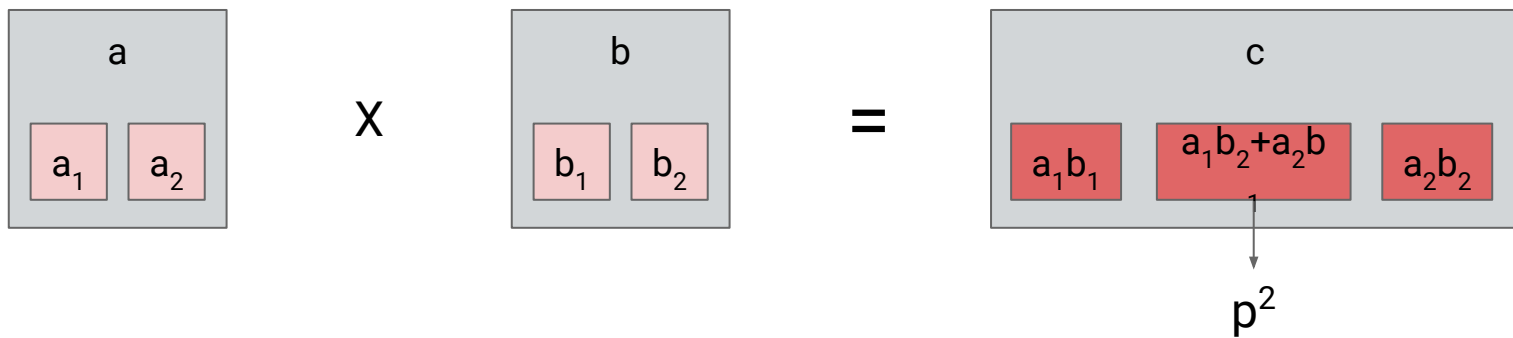
Multiplication





 Some Error

 More Error

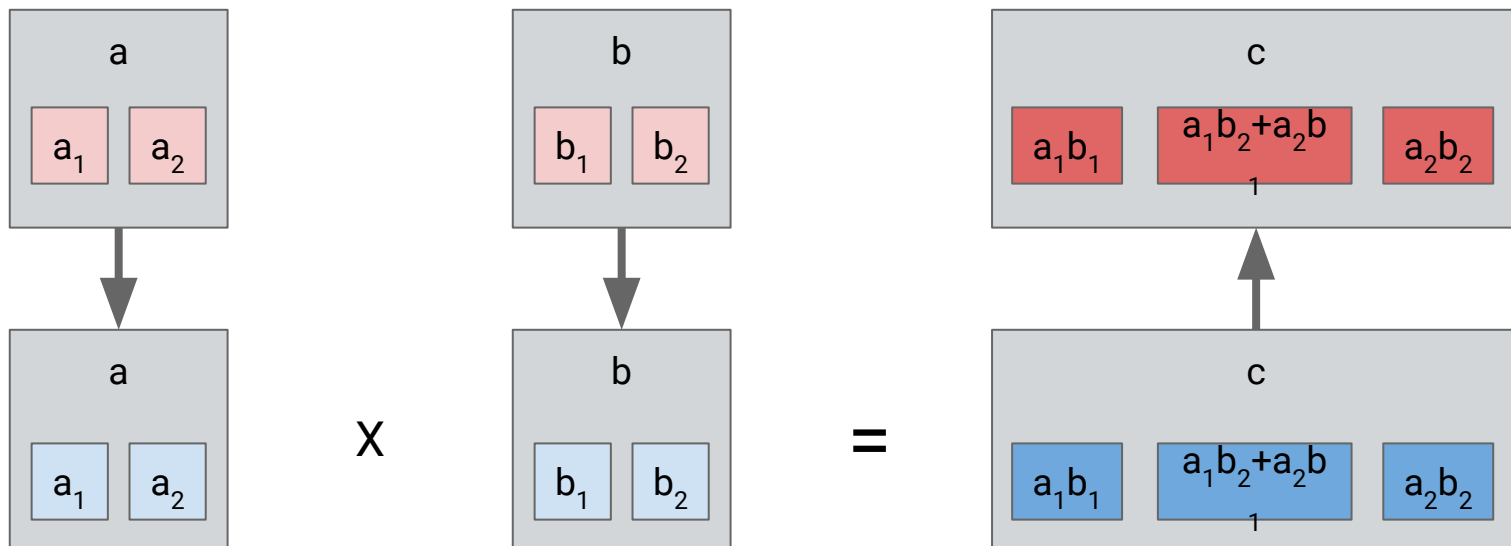
Multiplication (Time Complexity)



 Some Error

 More Error

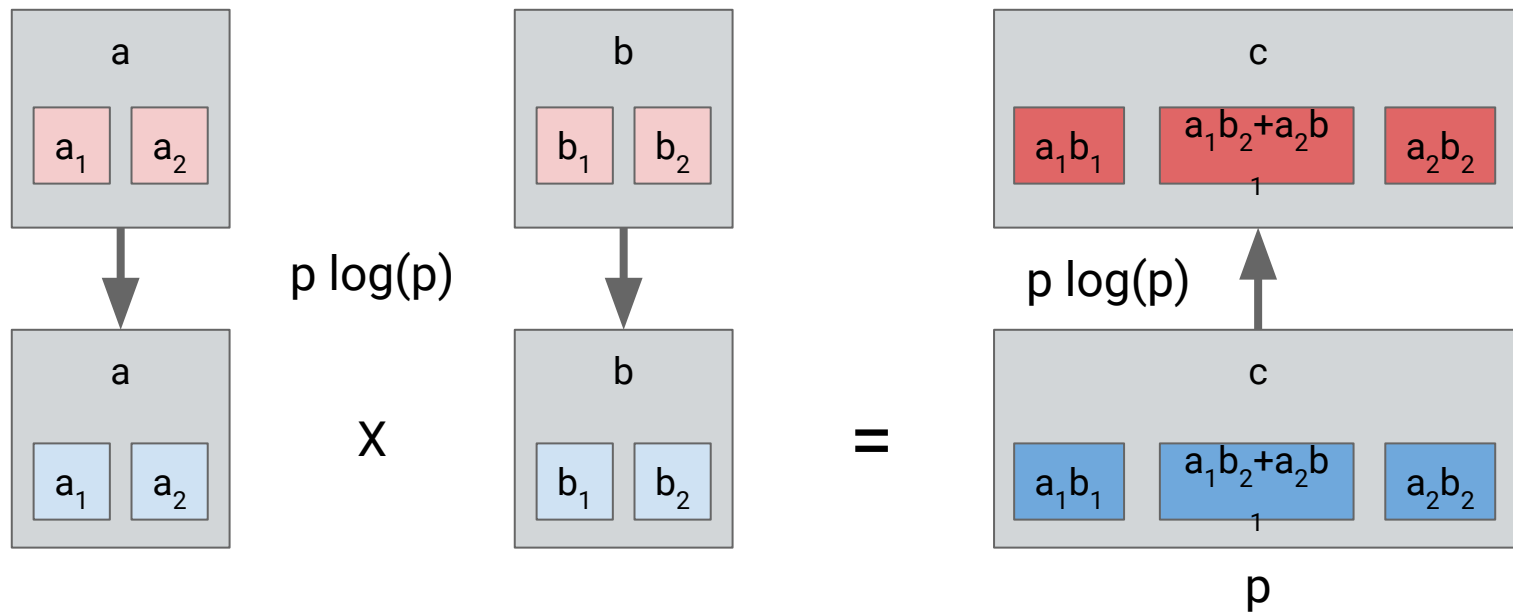
Number-Theoretic Transforms (Point Form)



 Coefficient Form

 Point/NTT Form

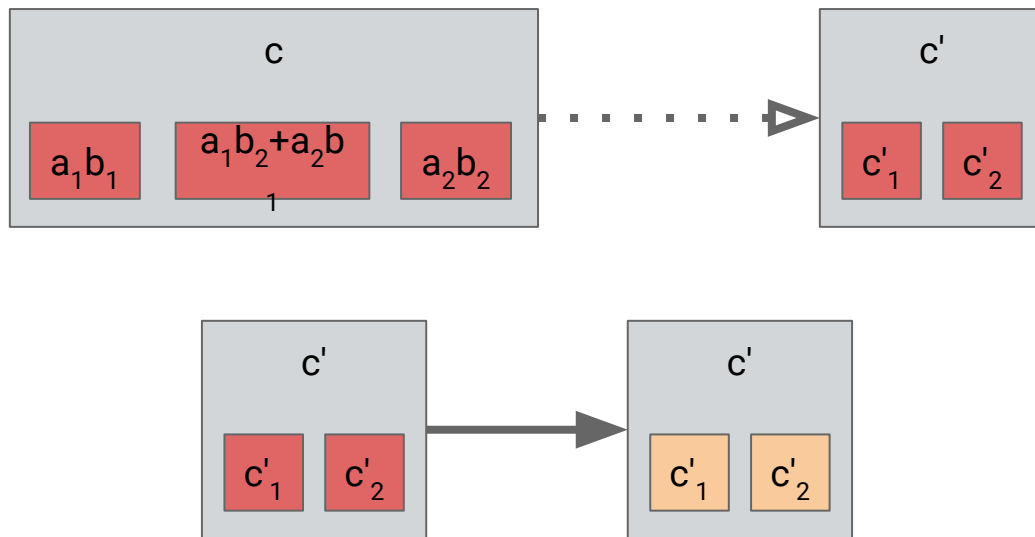
Number-Theoretic Transforms (Point Form)



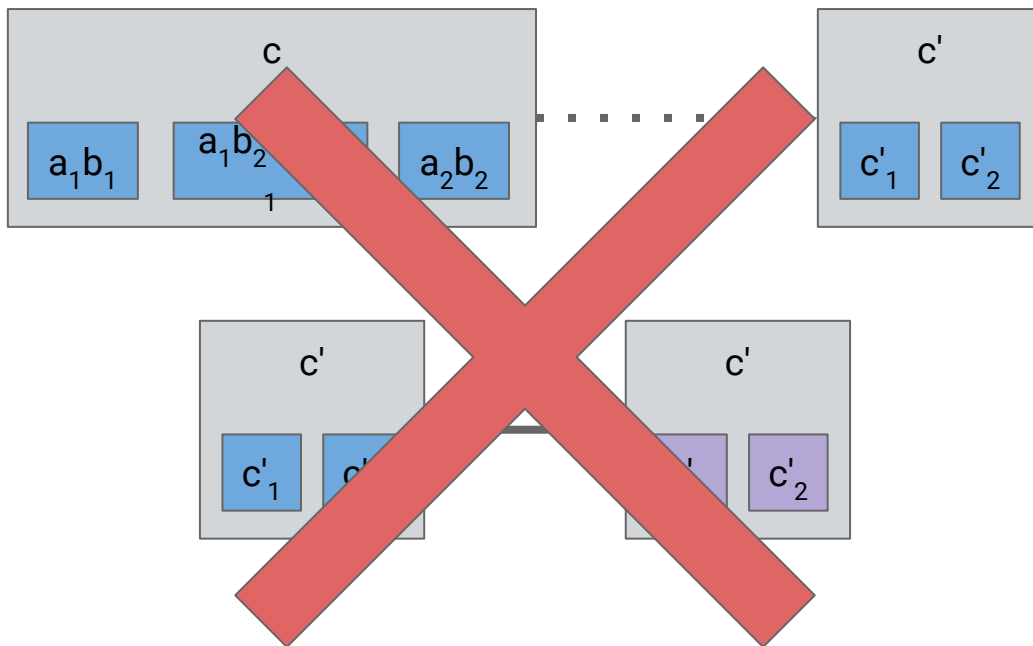
 Coefficient Form

 Point/NTT Form

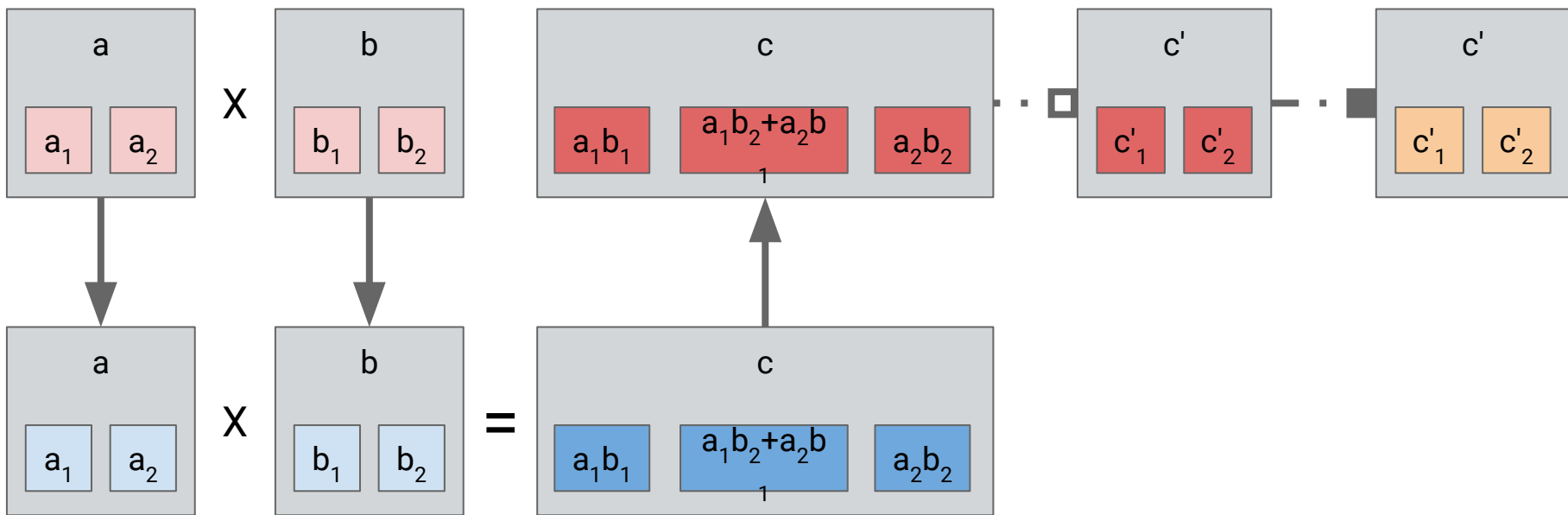
Key & Mod Switching



The Cost



Current PIR (1/2)



Current PIR (2/2)

Database

0

X

q_1

1

X

$1-q_1$

2

X

q_1

3

X

$1-q_1$

Size 1

Size 2

Query

X

q_2

X

$1-q_2$

Size 2

Size 2

Size 2

R

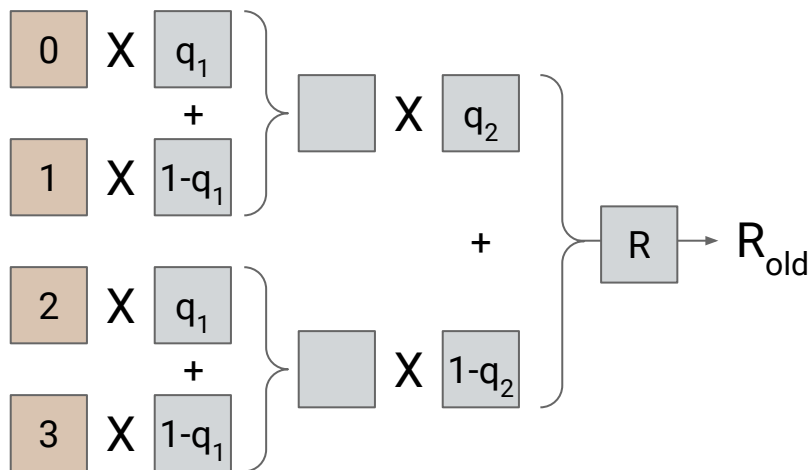
?

Boyan

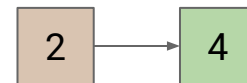
Folding & Updatability

Database

Query



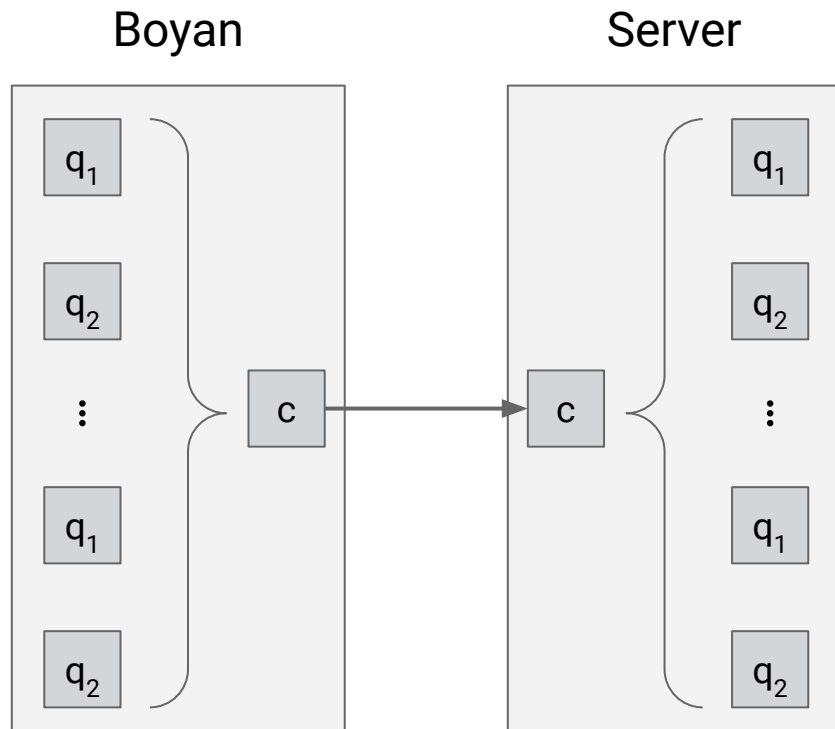
Update



$$\begin{array}{r}
 R_{old} \quad - \quad \left[\begin{array}{c} 2 \end{array} \right] \times \left[\begin{array}{c} q_1 \end{array} \right] \times \left[\begin{array}{c} 1-q_2 \end{array} \right] \\
 + \quad \left[\begin{array}{c} 4 \end{array} \right] \times \left[\begin{array}{c} q_1 \end{array} \right] \times \left[\begin{array}{c} 1-q_2 \end{array} \right] \\
 \hline
 = \quad R_{new}
 \end{array}$$

Side Note: Query Packing

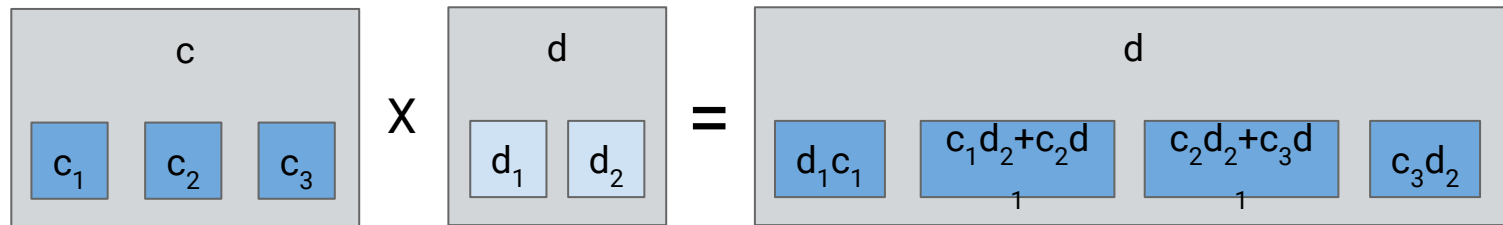
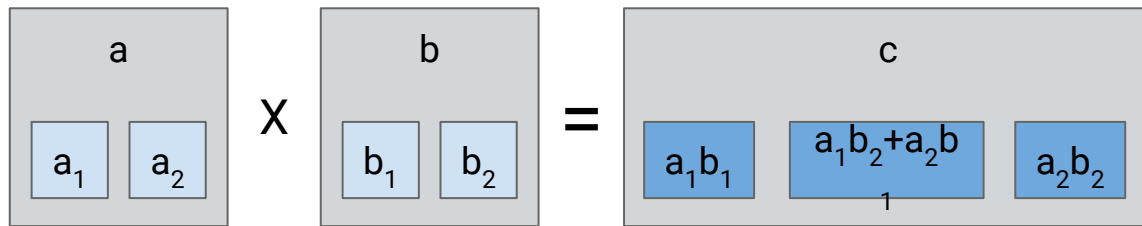
- Since queries only store 0 or 1, multiple "query ciphertexts" can be stored in one ciphertext
- Unpacking time is proportional to the size of the query



Our Scheme



Only NTT Form



Scheme Diagram

Database

0

X

q_1

1

X

$1-q_1$

2

X

q_1

3

X

$1-q_1$

Size 1

Size 2

Query

X

q_2

X

$1-q_2$

Size 2

Size 2

Size 3

R

?

Boyan

Cost

- More noise growth
 - Smaller multiplicative depth
- After each multiplication, ciphertext size increases
 - Subsequent multiplications take longer

Benefits

- Ciphertexts are always in NTT form, so computations on each point can be done independently
 - Easy parallelization
- Updatable

Preliminary Results



Database Size	PRIMES_PIR v1				SpiralStreamPack		
	Query Size (KB)	Response Size (KB)	Answer Time (s)		Query Size (KB)	Resp. Size (KB)	Answer Time (s)
			1 thread	12 Threads			
2 ¹	1573	1573	0.002	-	3785	71	0.104
2 ⁷	11010	6291	0.238	0.043 (5.52x)	3785	71	0.104
2 ⁸	12583	7078	0.440	0.083 (5.83x)	3785	71	0.105
2 ⁹	14156	7684	0.885	0.150 (6.57x)	3785	71	0.104
2 ¹⁰	15729	8651	1.820	0.286 (6.23x)	3785	71	0.107
2 ¹¹	17302	9437	3.961	0.601 (5.97x)	7455	71	0.134
2 ¹²	18874	10224	8.010	1.299 (5.50x)	14795	71	0.203

Preliminary Results (vs Non-Updatable Scheme)

Future Work

- Different Folding Schemes
- Protocol for Sparse Databases

Acknowledgments

Thanks to MIT PRIMES and Simon Langowski for making this project possible!

Questions?