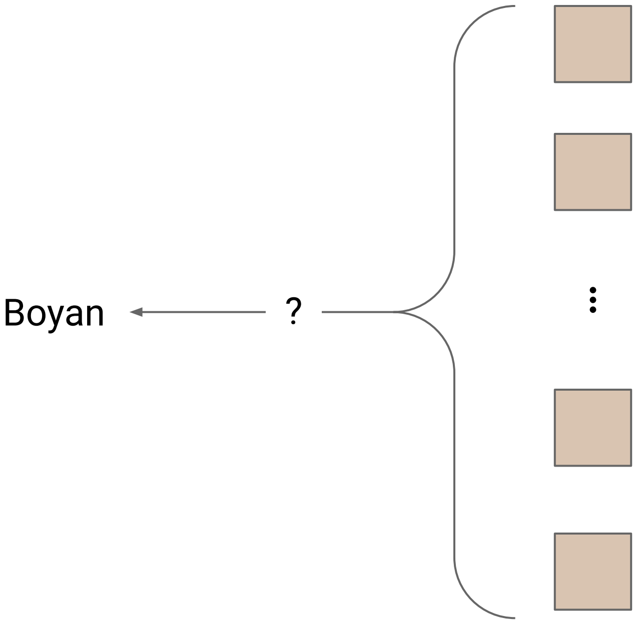
Parallelizable and Updatable Private Information Retrieval

By Boyan Litchev
Mentored by Simon Langowski

Private Information Retrieval (PIR)

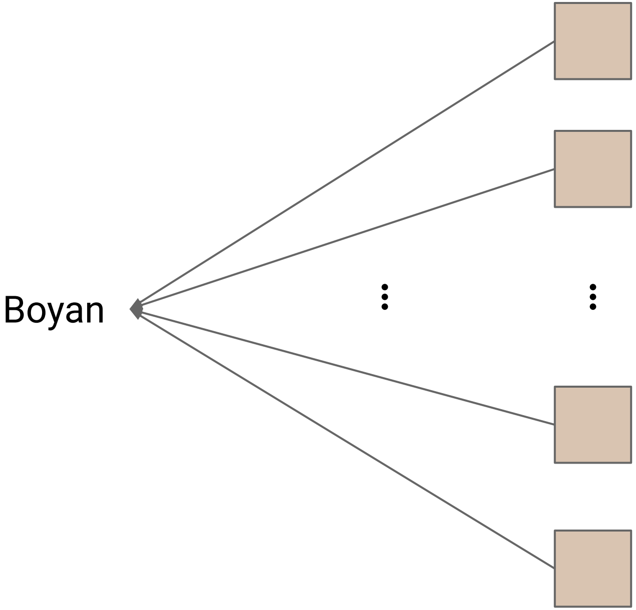
The Problem



Use Cases

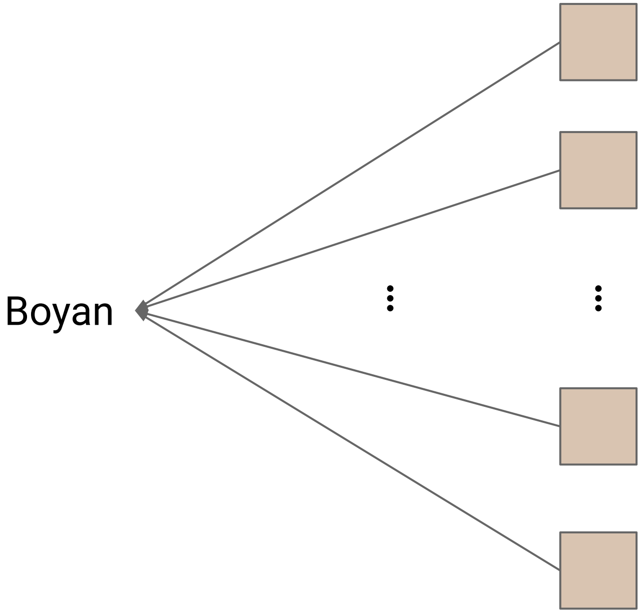
- Private Browsing
- Private Streaming
- Anonymous Messaging

A Simple Solution (1/2)



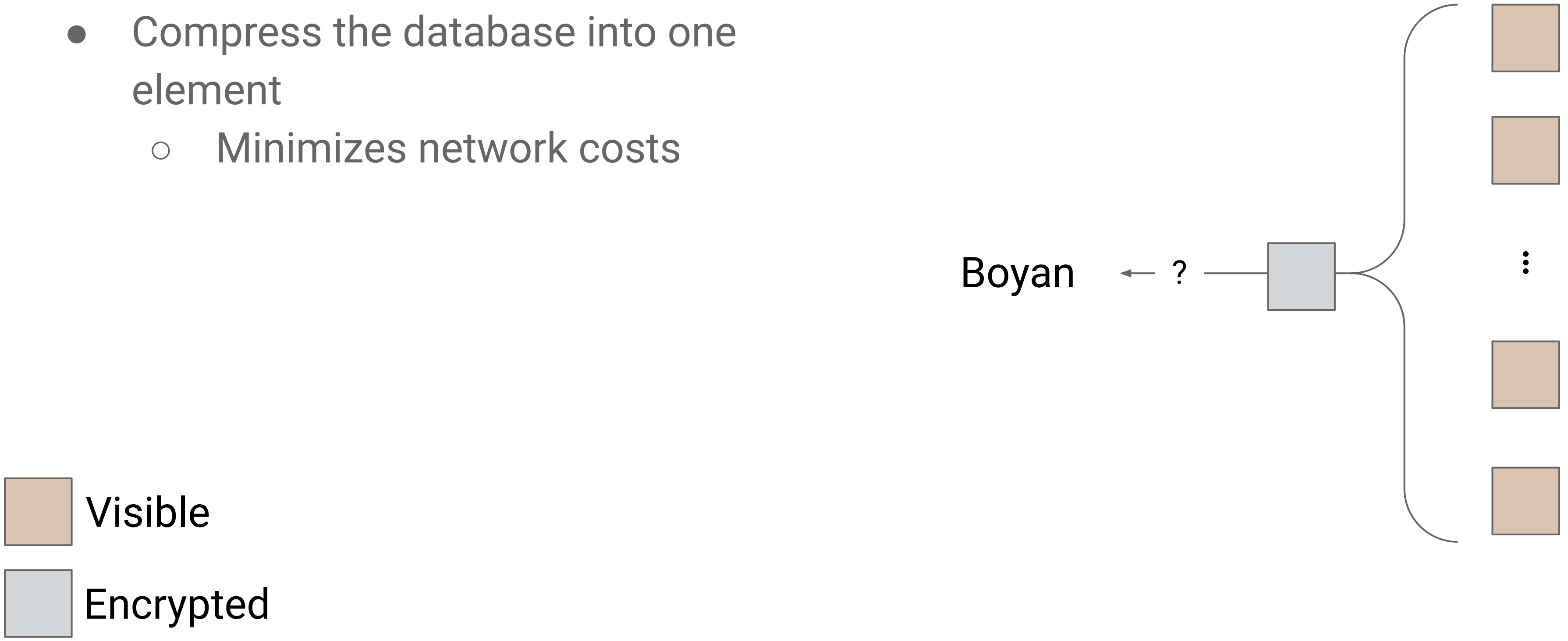
A Simple Solution (2/2)

- Network Costs are the entire database
  - Too high

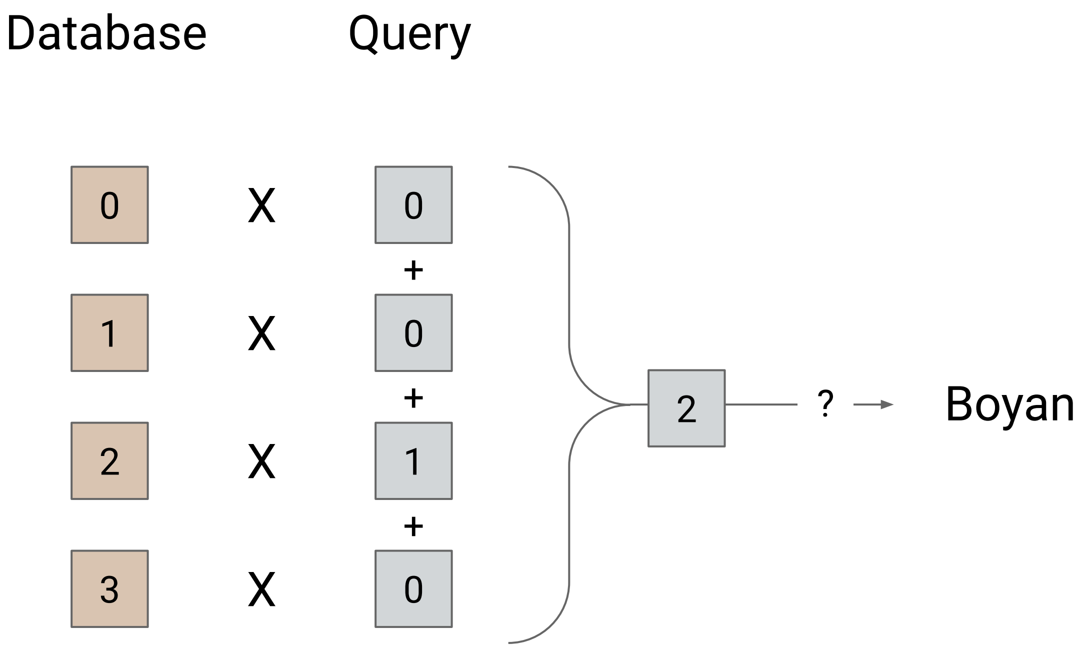


The Goal

- Compress the database into one element
    - Minimizes network costs



The Approach



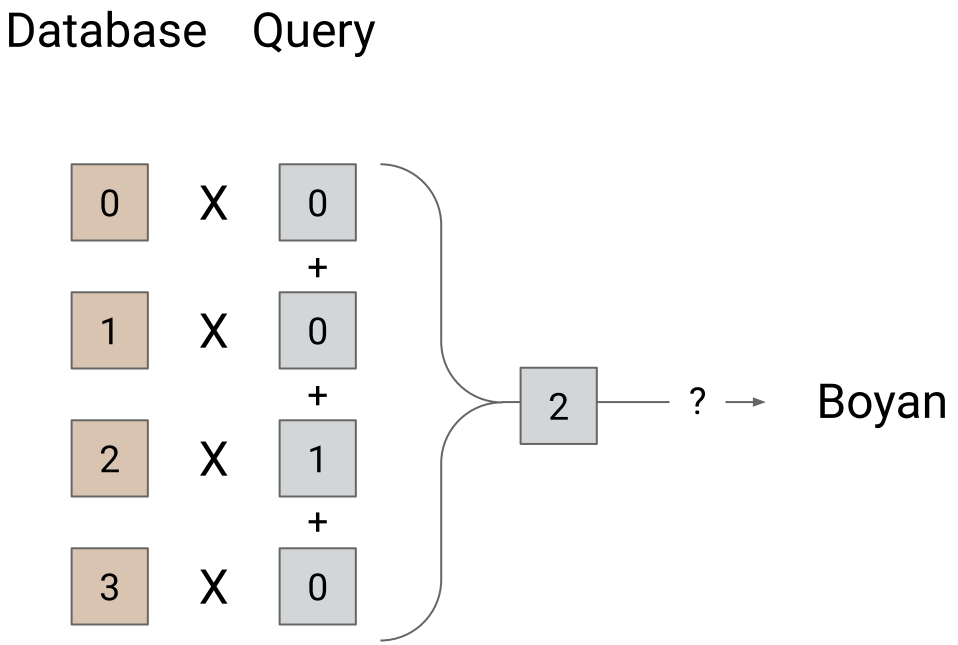
Costs

Network

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

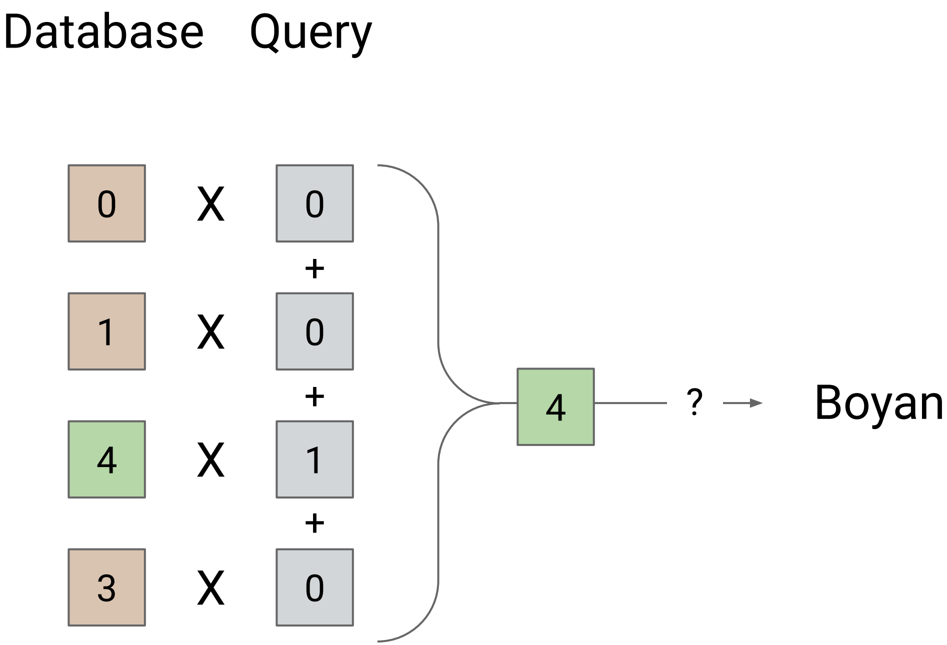
Computational

- n multiplications
- n-1 additions

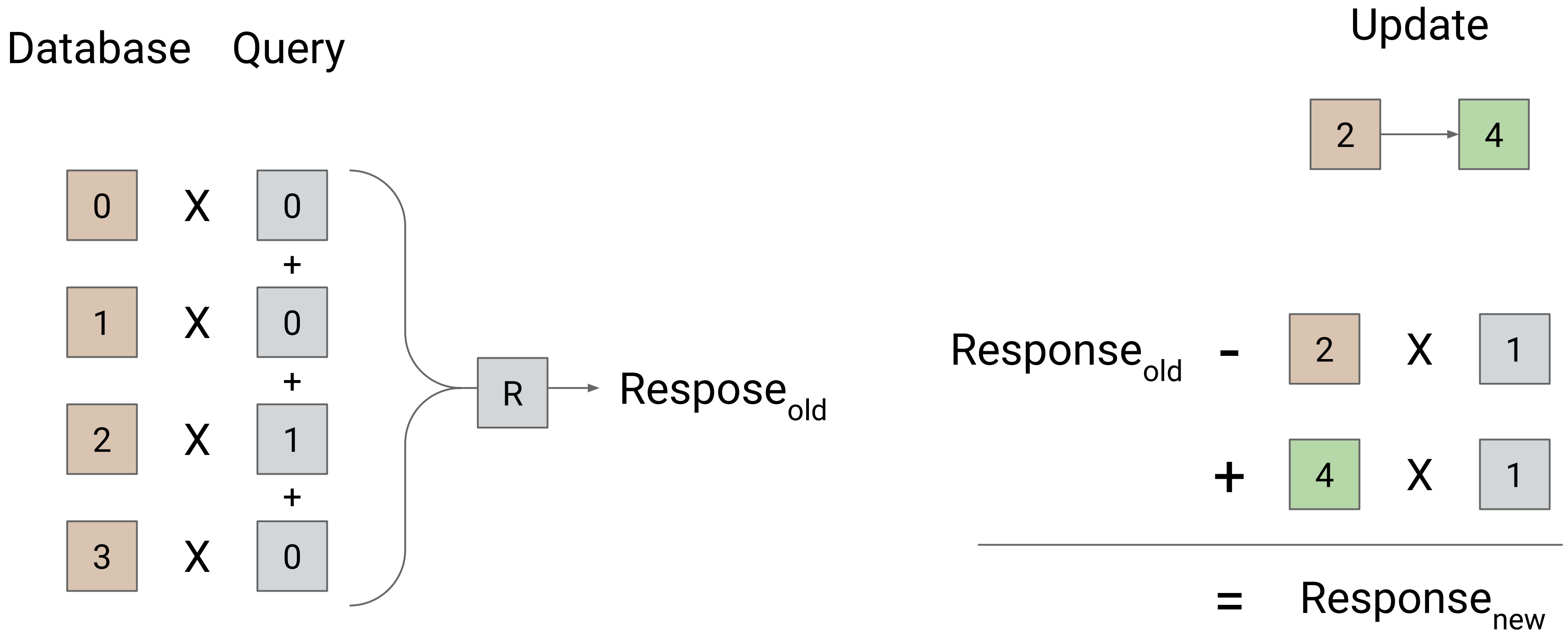


Updatability (1/2)

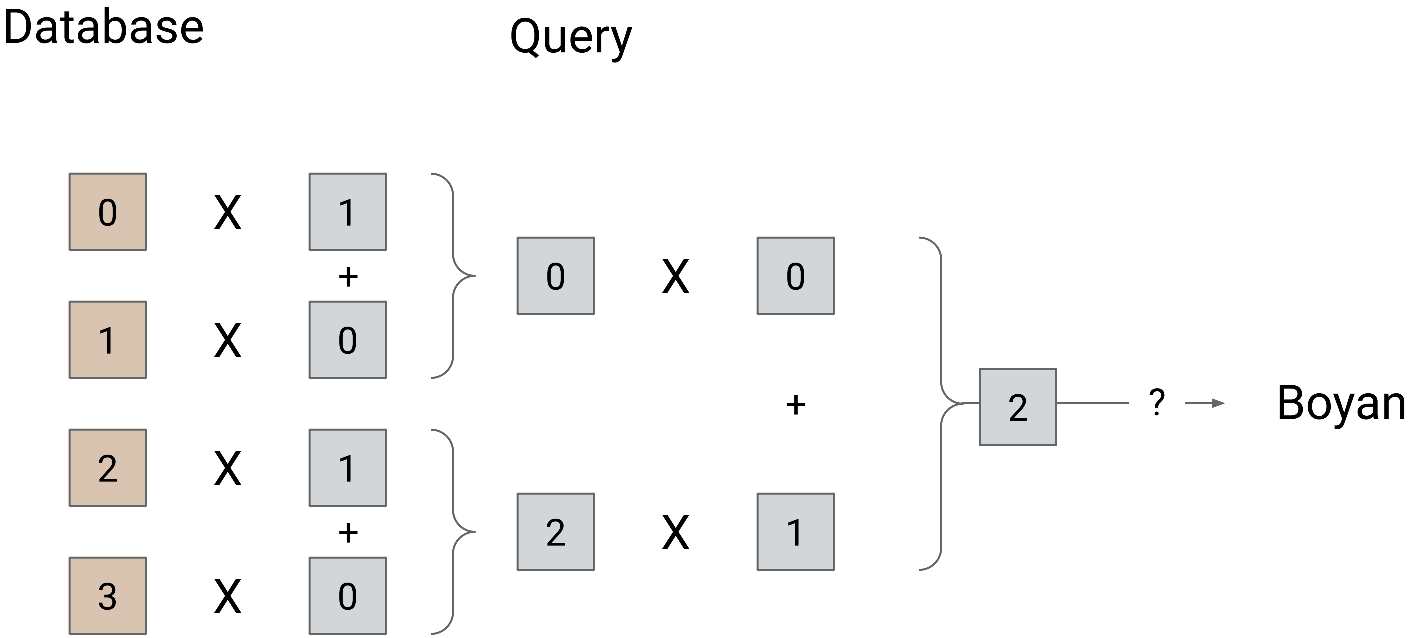
- If the database changes, the old response can be updated without computing on a large part of the database



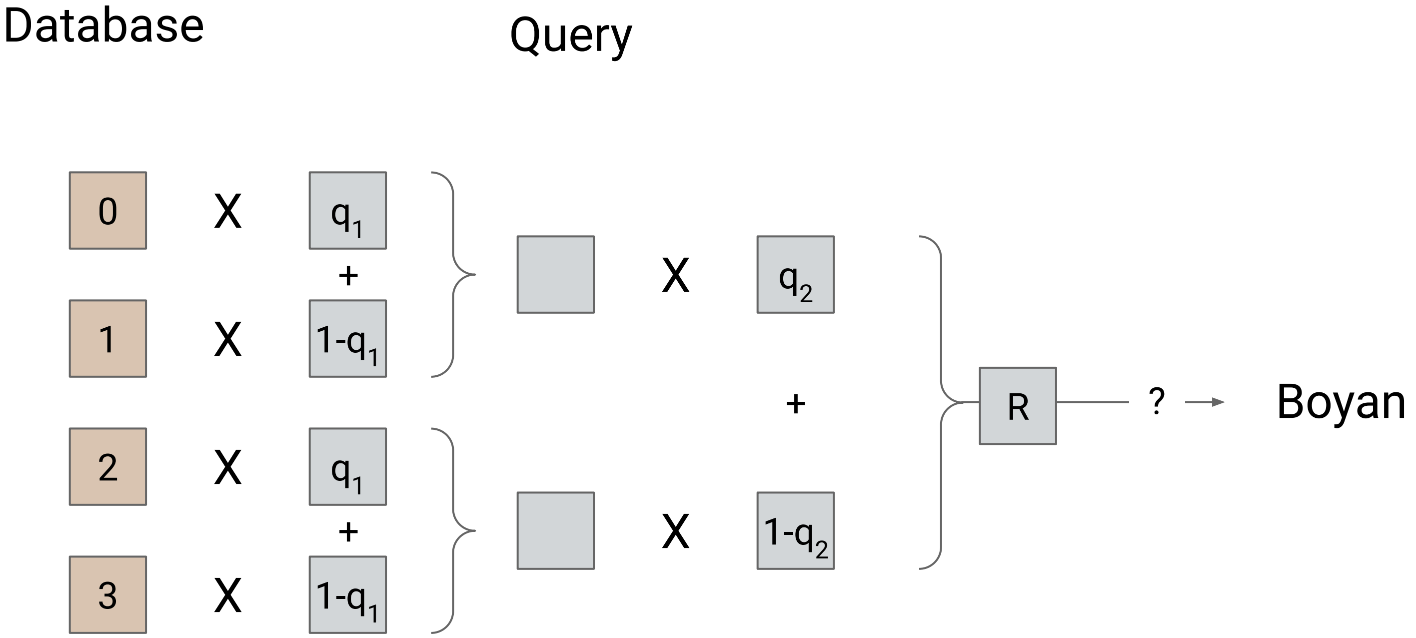
Updatability (2/2)



Folding (1/2)



Folding (2/2)



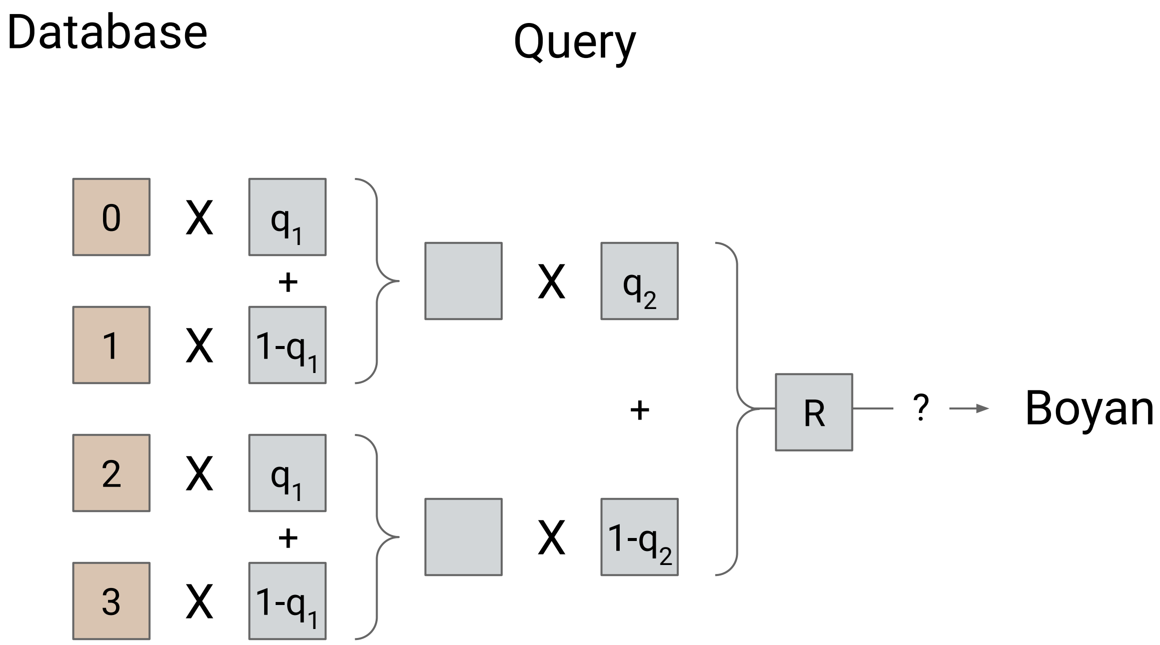
Costs

Network

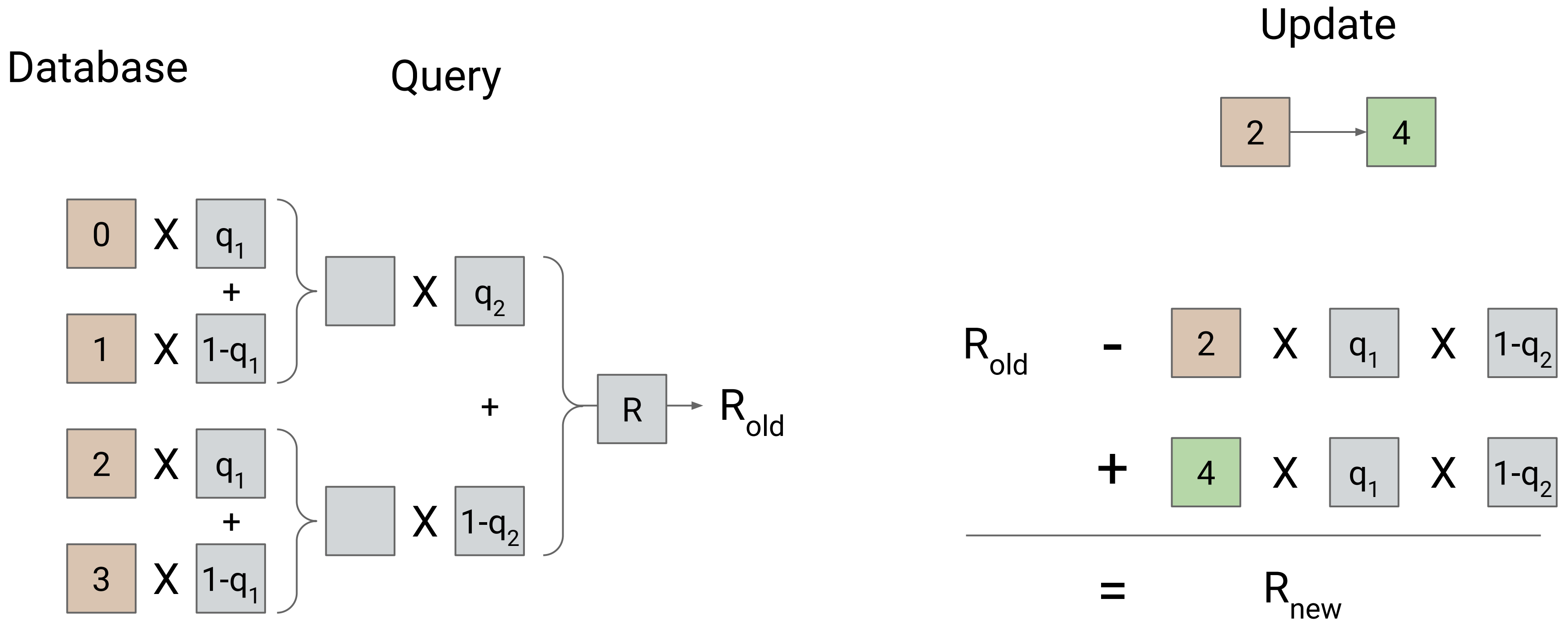
- Query is $\log_2(n)$ elements
- Response is 1 element

Computational

- 2n-1 multiplications
- n-1 additions



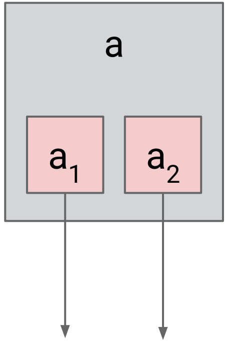
Updatability



Homomorphic Encryption & Number-Theoretic Transforms (NTTs)

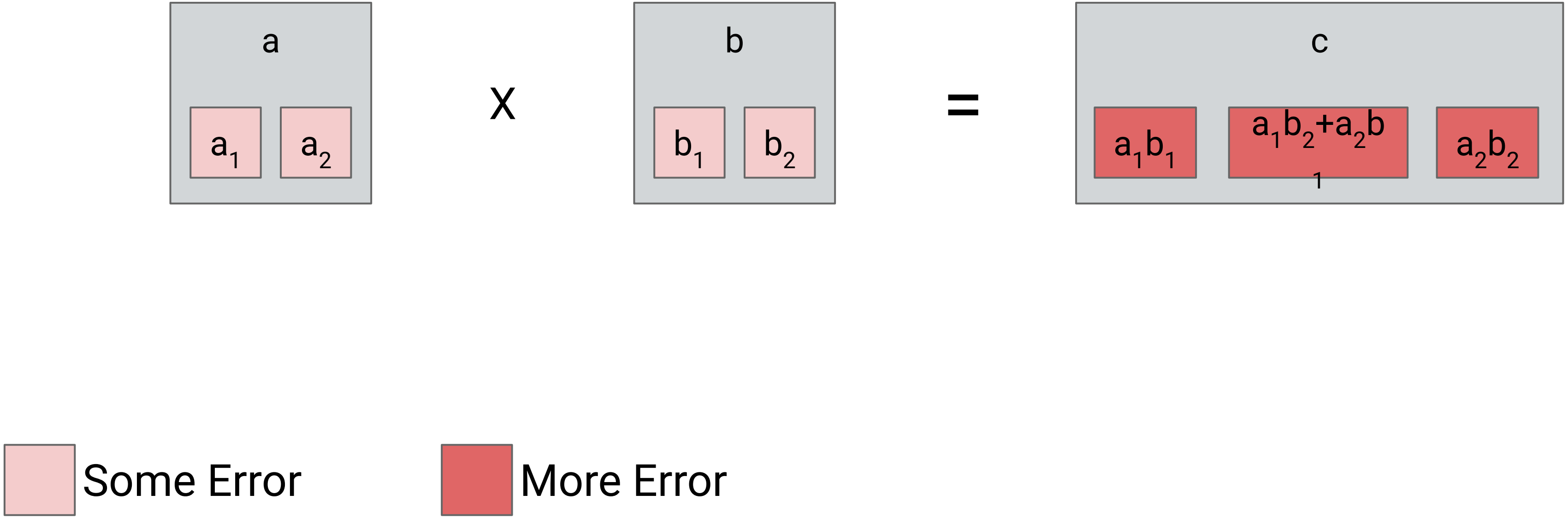
Fully Homomorphic Encryption Ciphertext

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

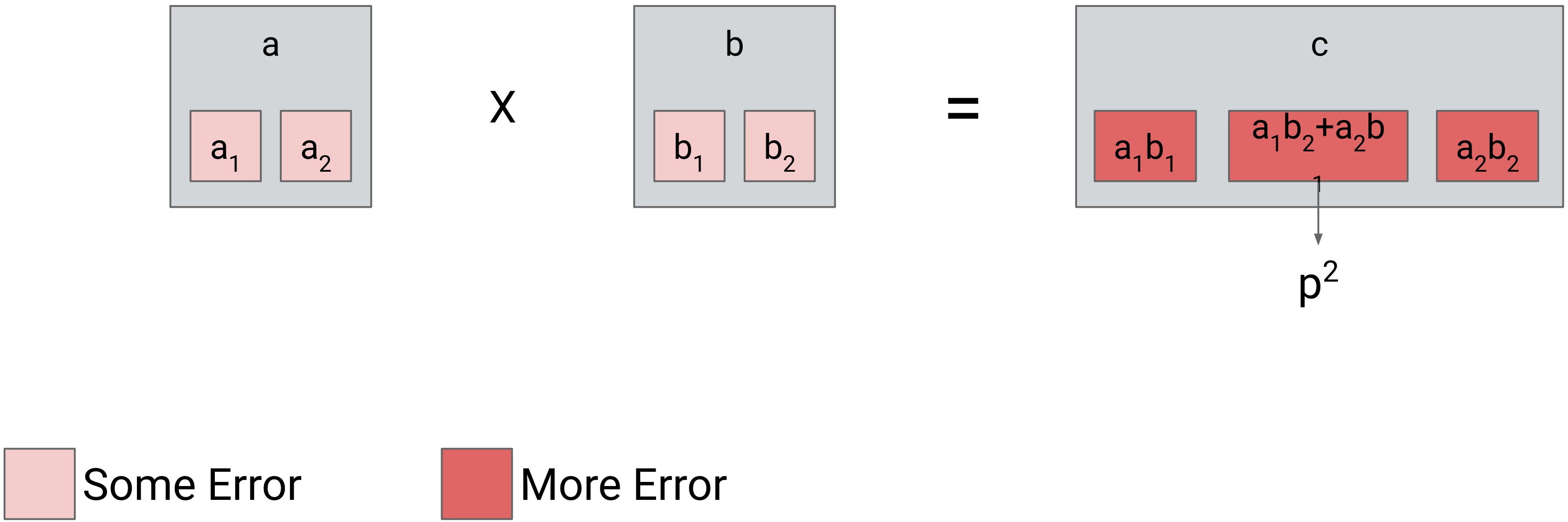


Polynomials
(Elements of $Z_Q[X]/[X^p+1]$)

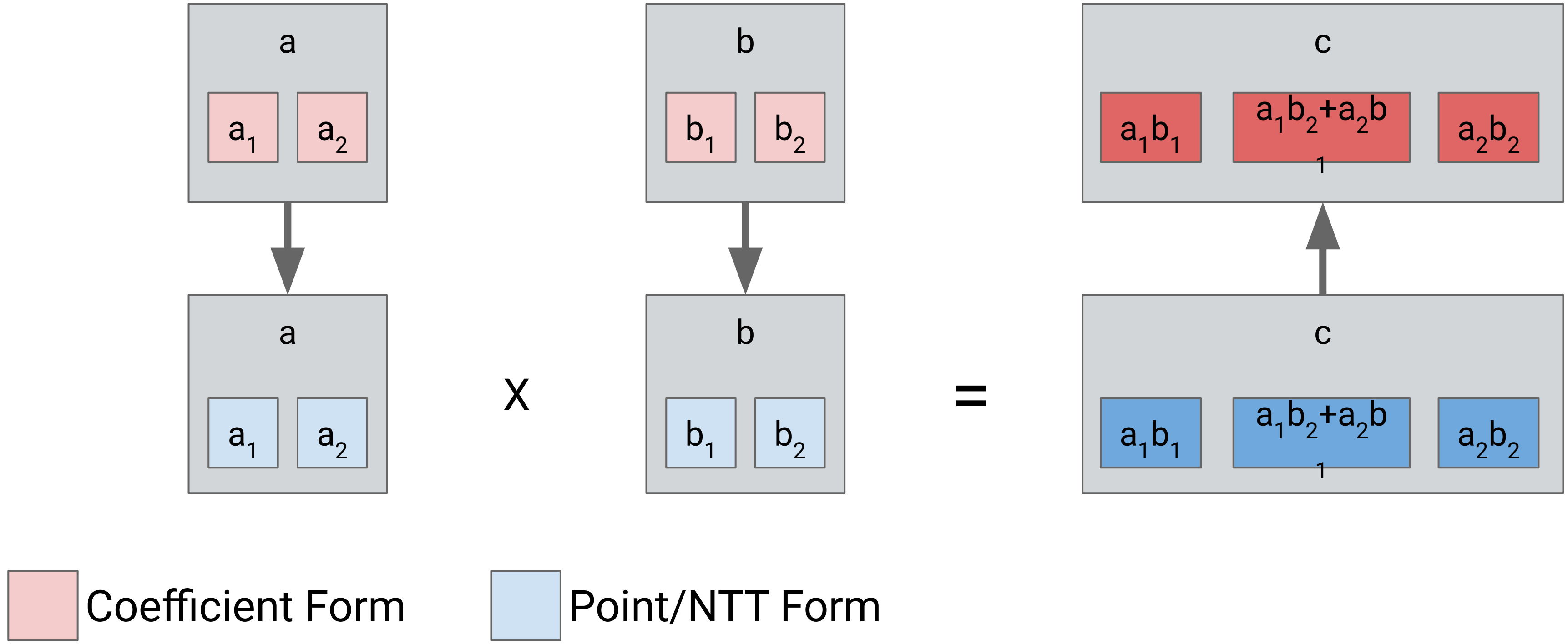
Multiplication



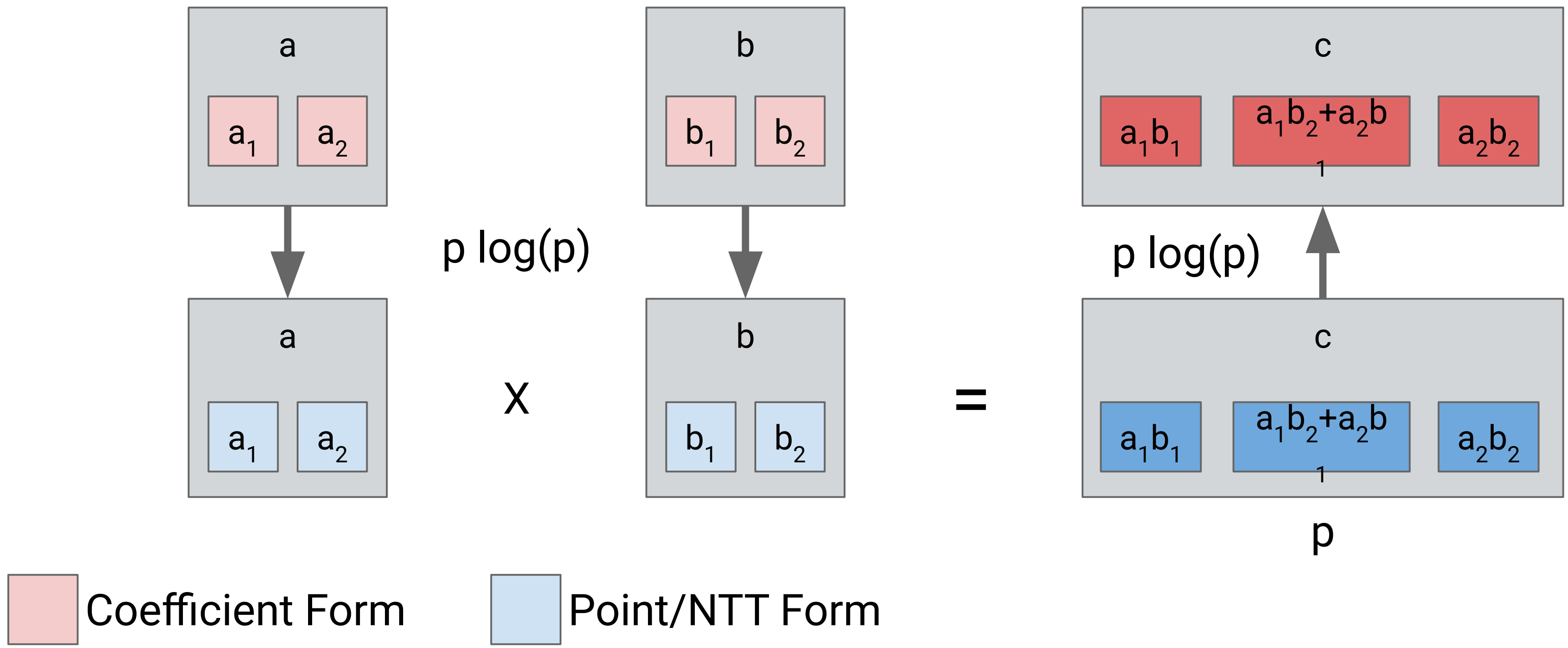
Multiplication (Time Complexity)



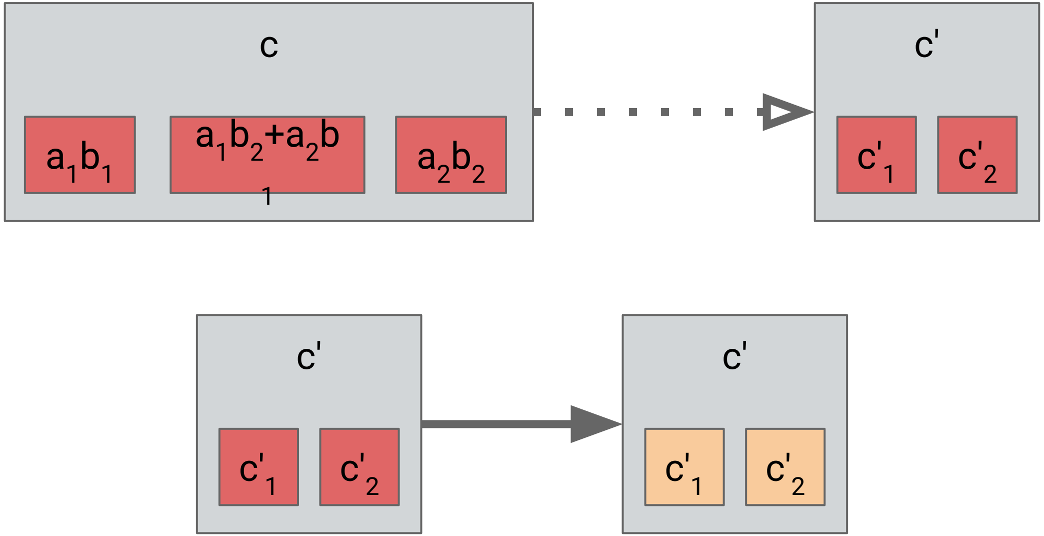
Number-Theoretic Transforms (Point Form)



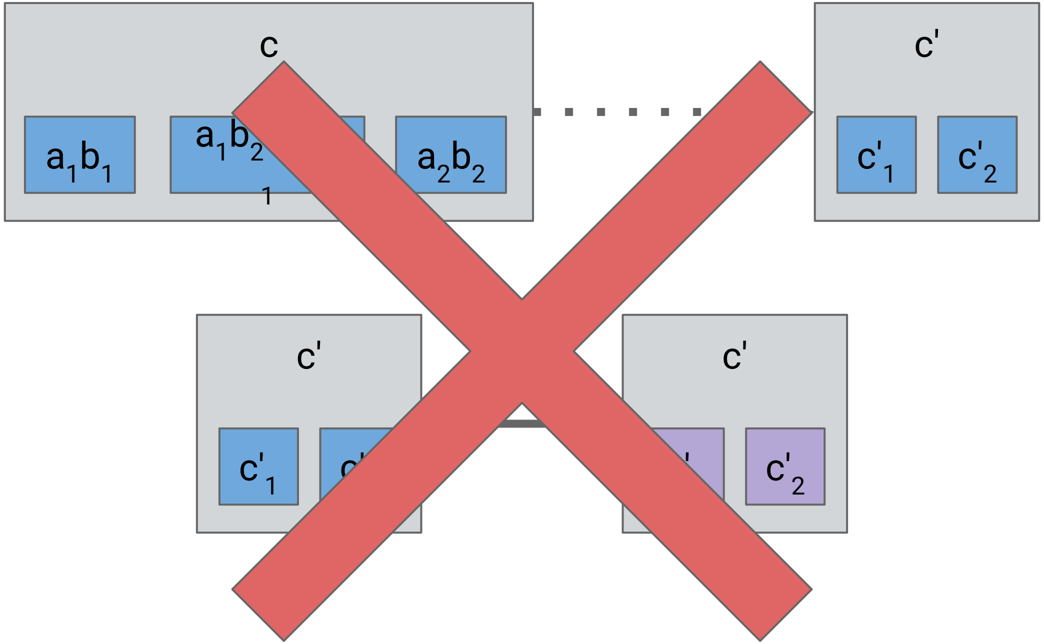
Number-Theoretic Transforms (Point Form)



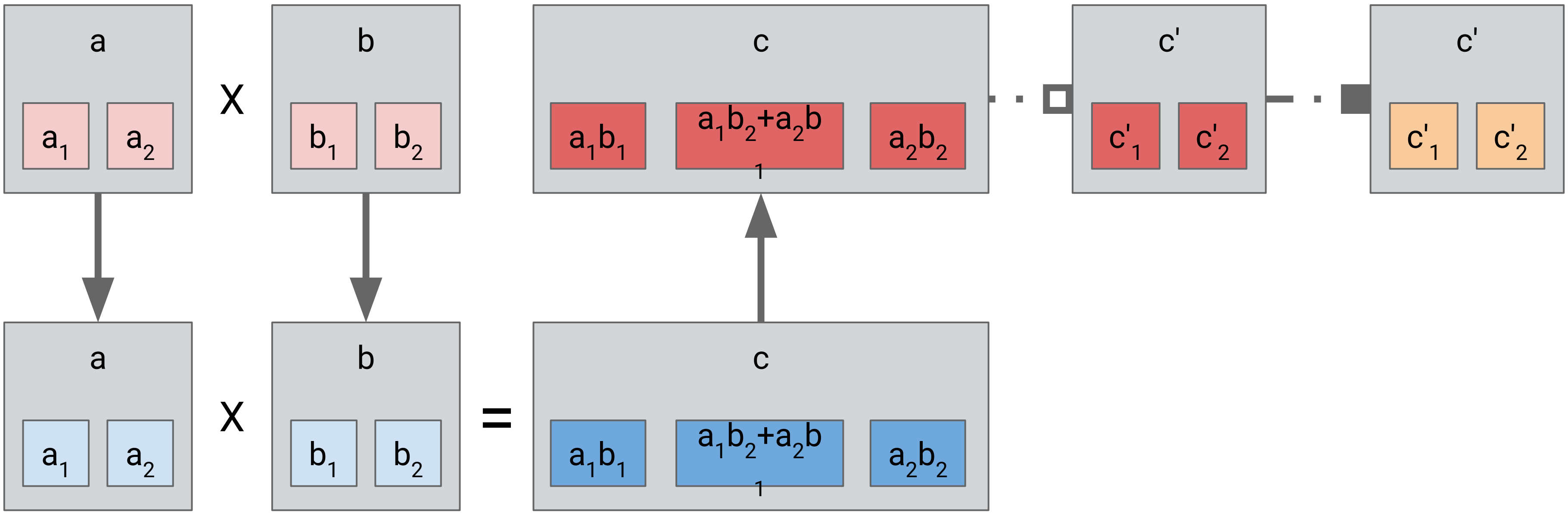
Key & Mod Switching



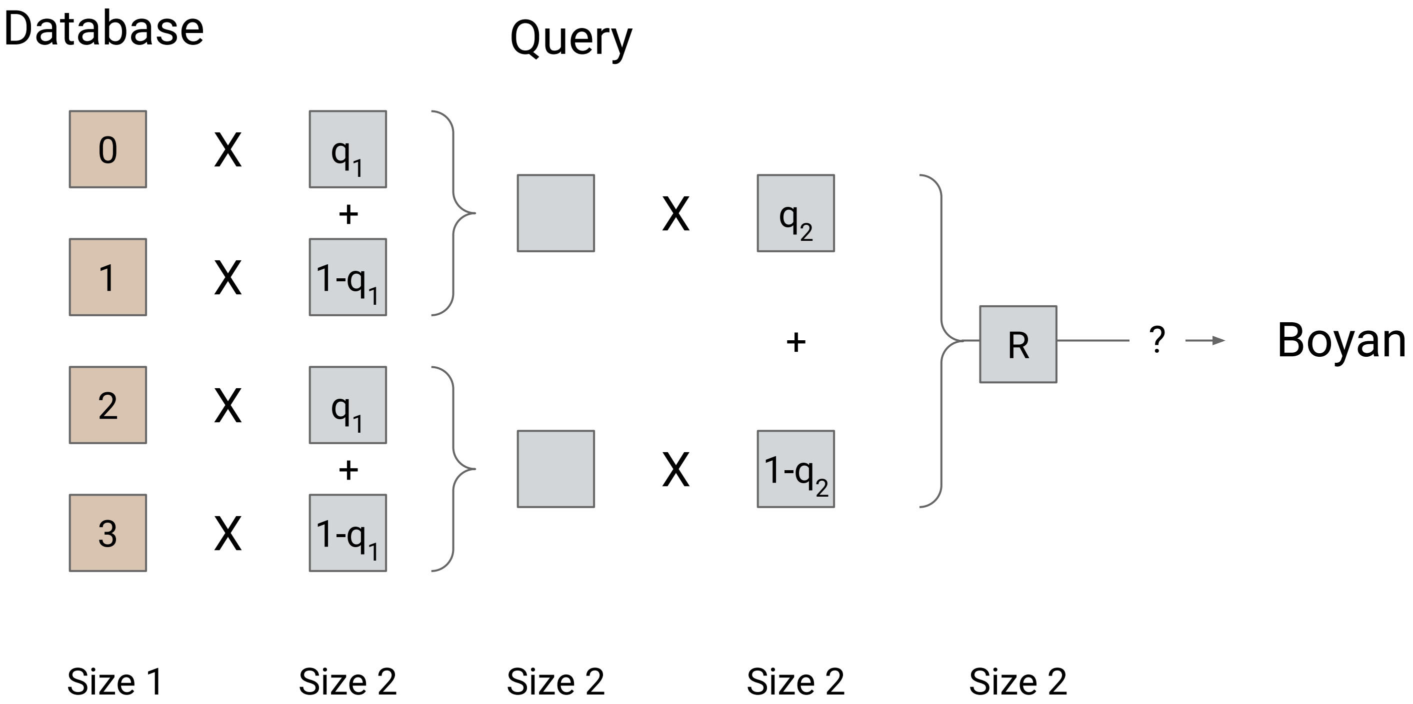
The Cost



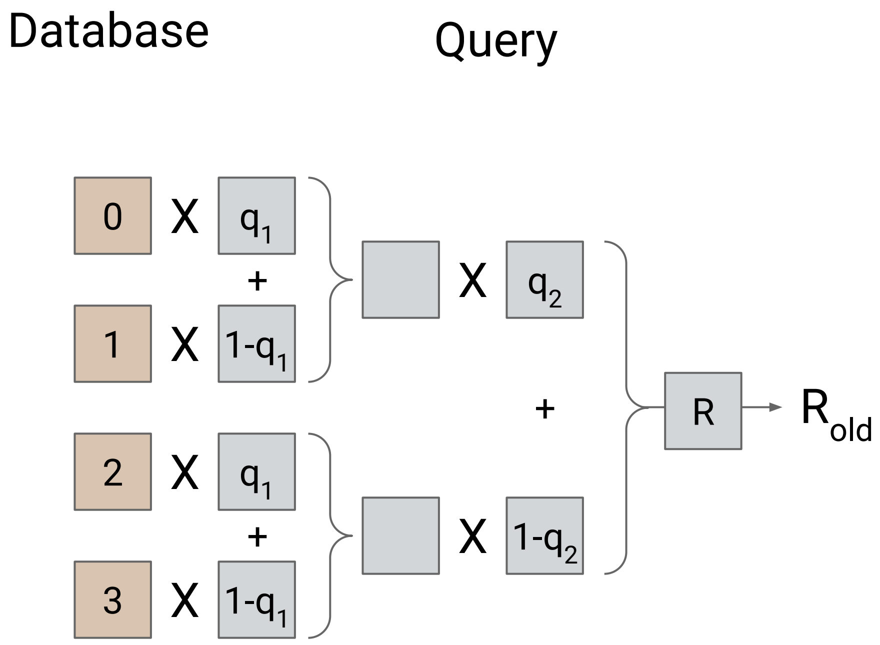
Current PIR (1/2)

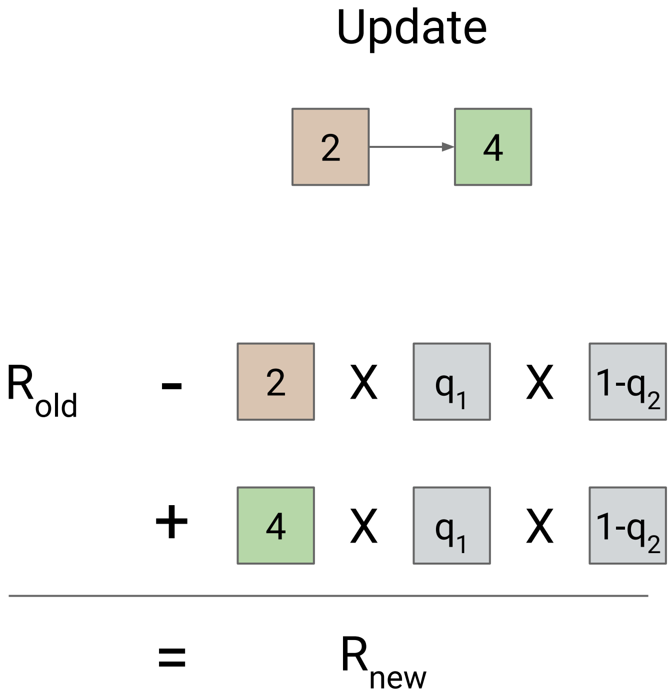


Current PIR (2/2)



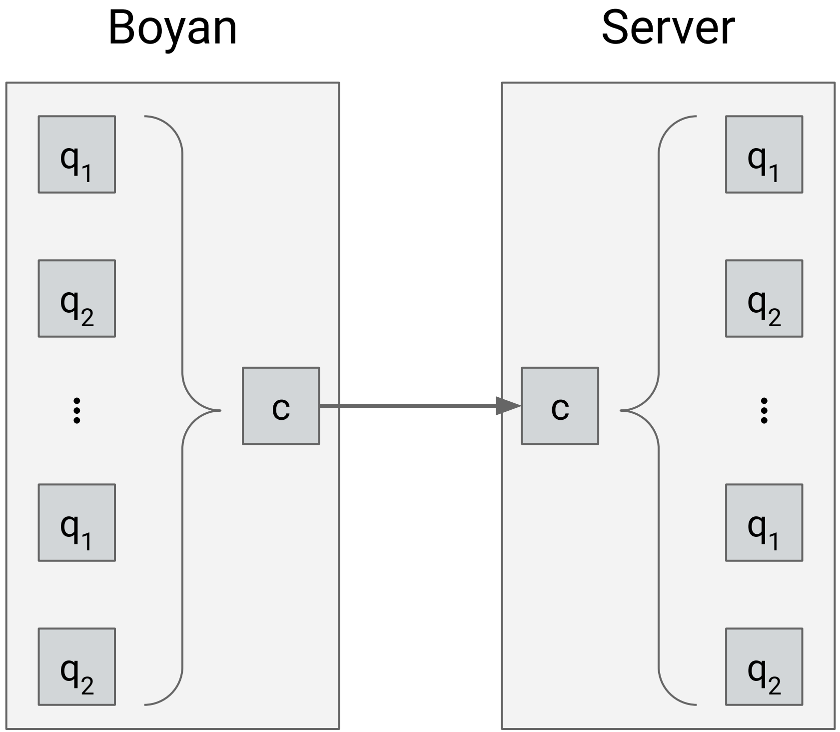
Folding & Updatability





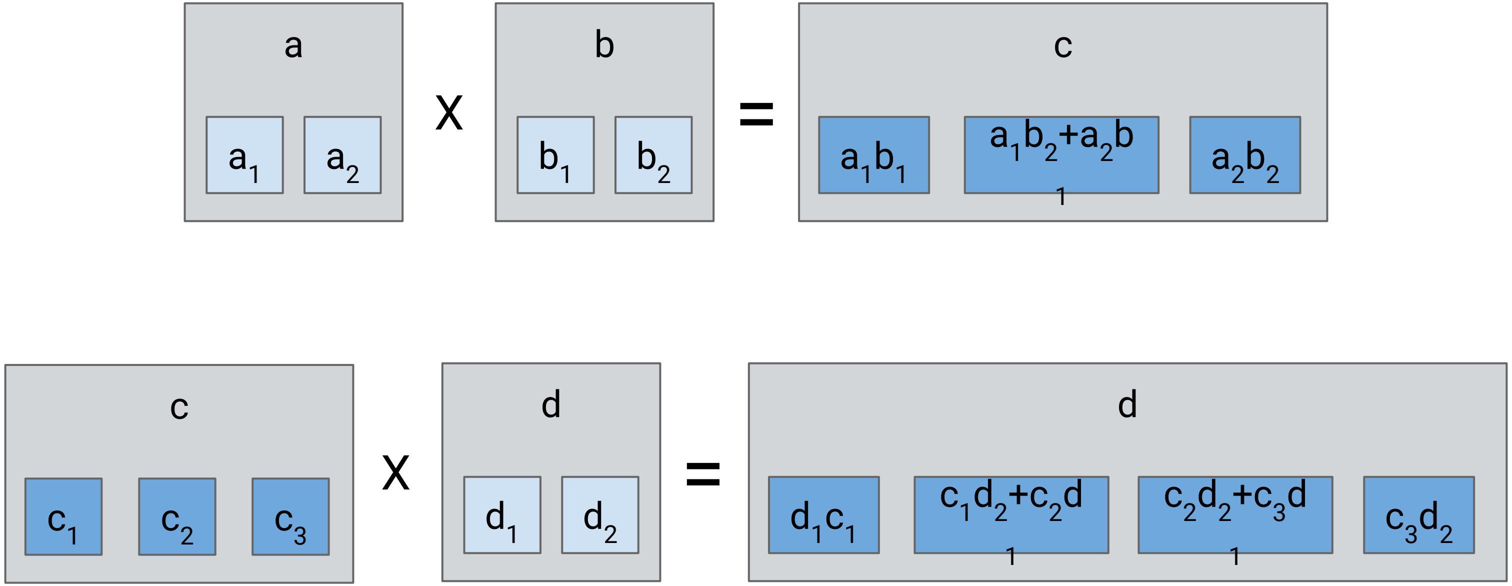
Side Note: Query Packing

- 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

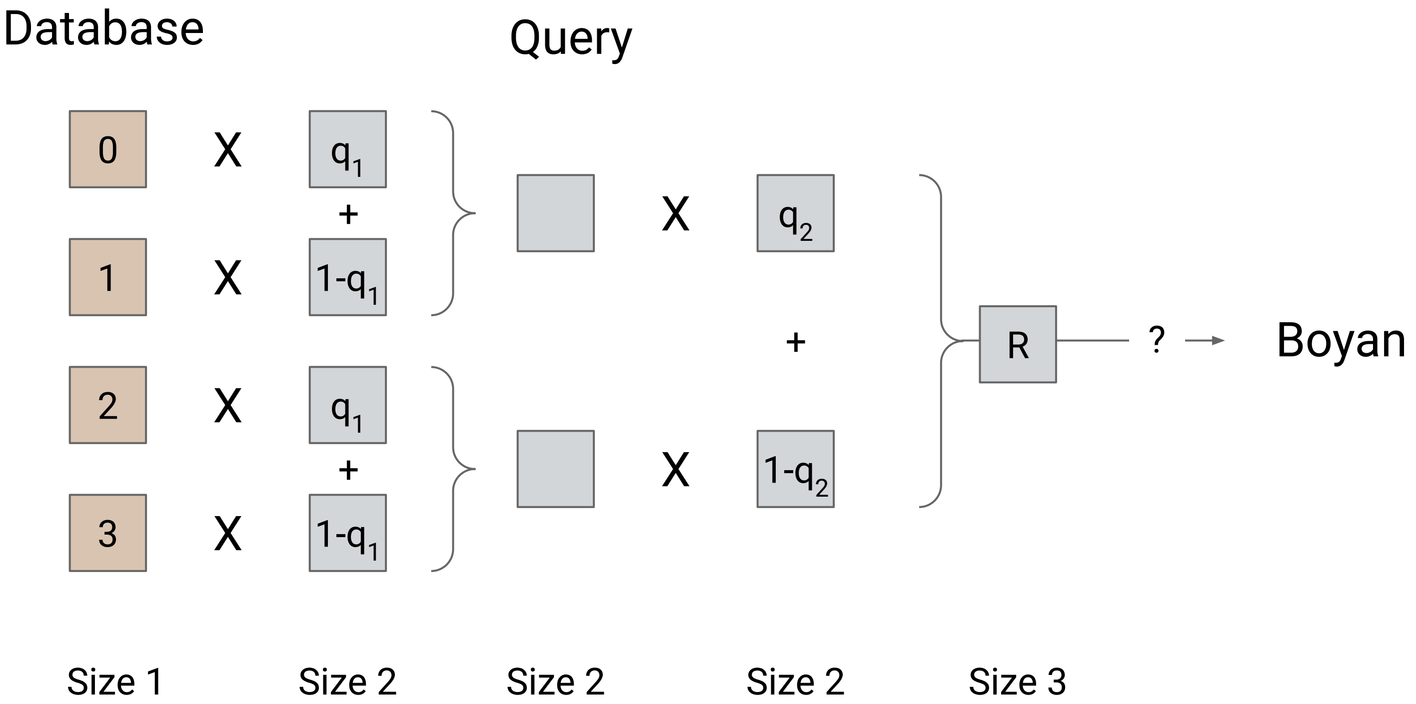


Our Scheme

Only NTT Form



Scheme Diagram



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?