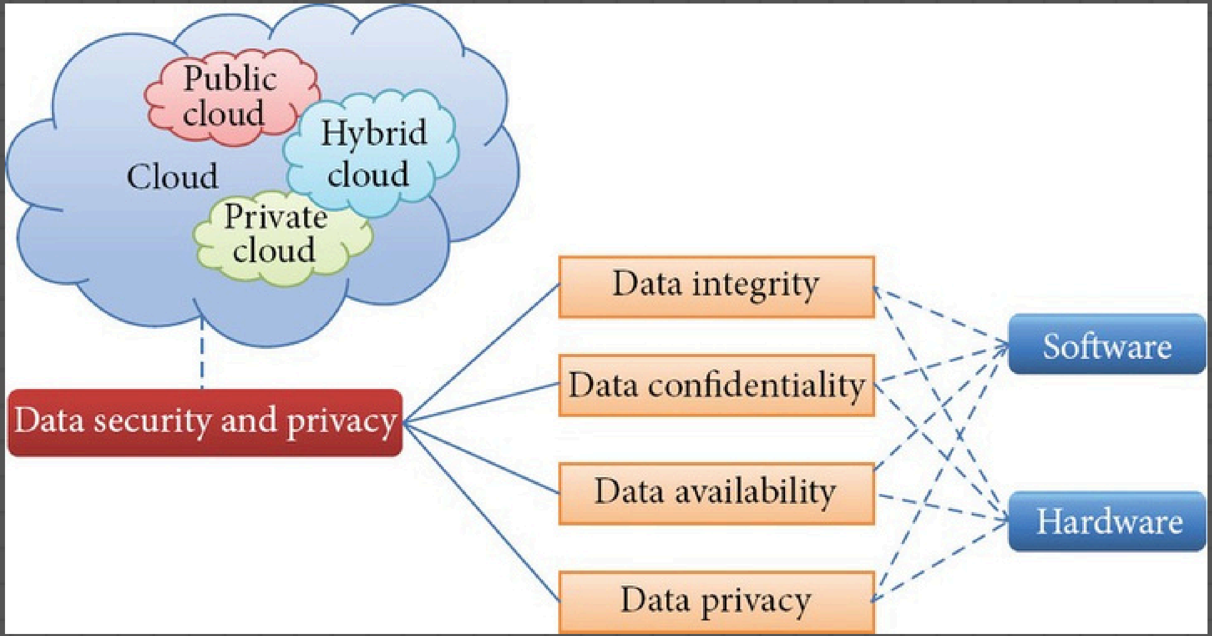
Range Optimised Duoram

UGP PRESENTATION

PRESENTED BY SRISHTI CHANDRA(221088)

MENTORED BY: PROF. ADITHYA VADAPALLI

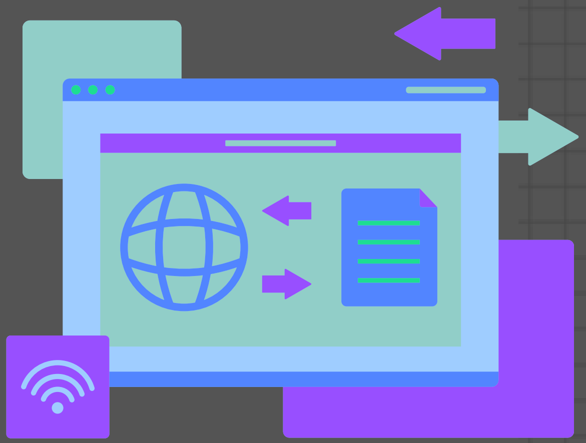
INTRODUCTION



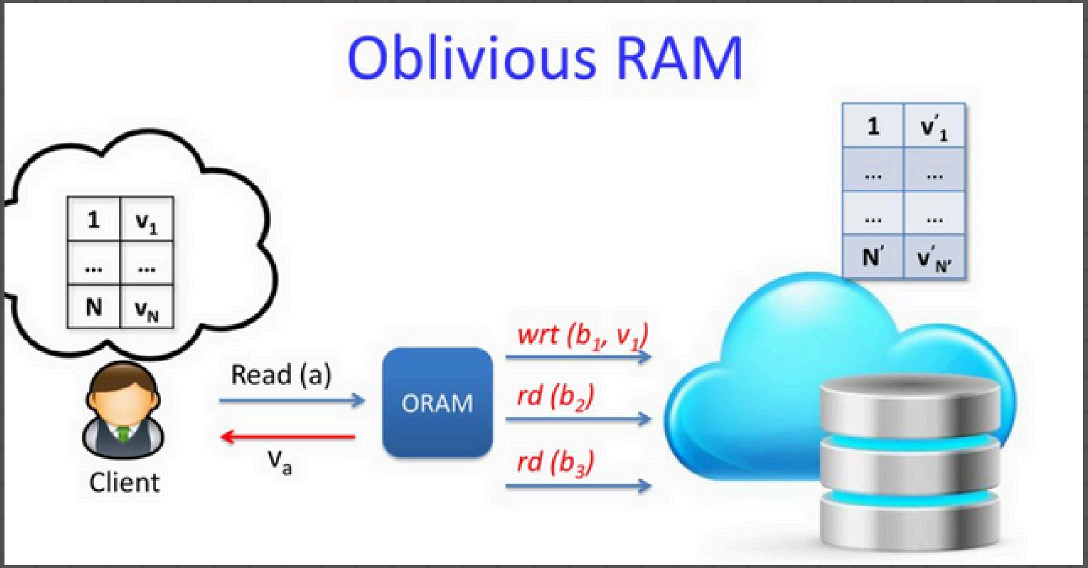
SRC: SAGE JOURNALS

1. ENCRYPTING
2. ACCESS PATTERN LEAKS
3. ORAM
4. COMPUTATIONAL COSTLY
5. TRADE OFF BTW COST & PRIVACY

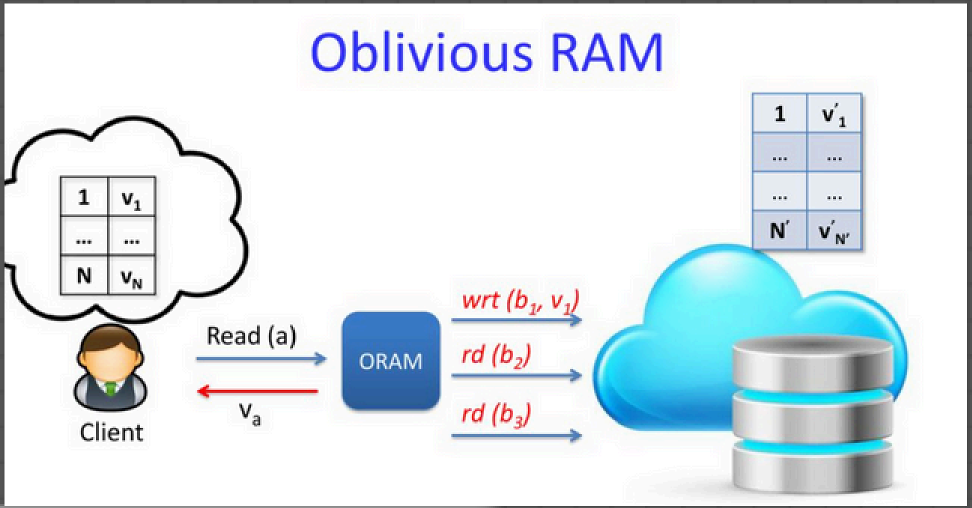
WHY CARE ABOUT ACCESS PATTERN PRIVACY



In secure systems like encrypted databases or cloud storage, even encrypted data leaks information through access patterns (e.g., repeated access to the same location reveals user preferences). Oblivious RAM (ORAM) protocols mask these patterns, making all accesses appear indistinguishable to prevent such leaks.



ORAM TO THE RESCUE



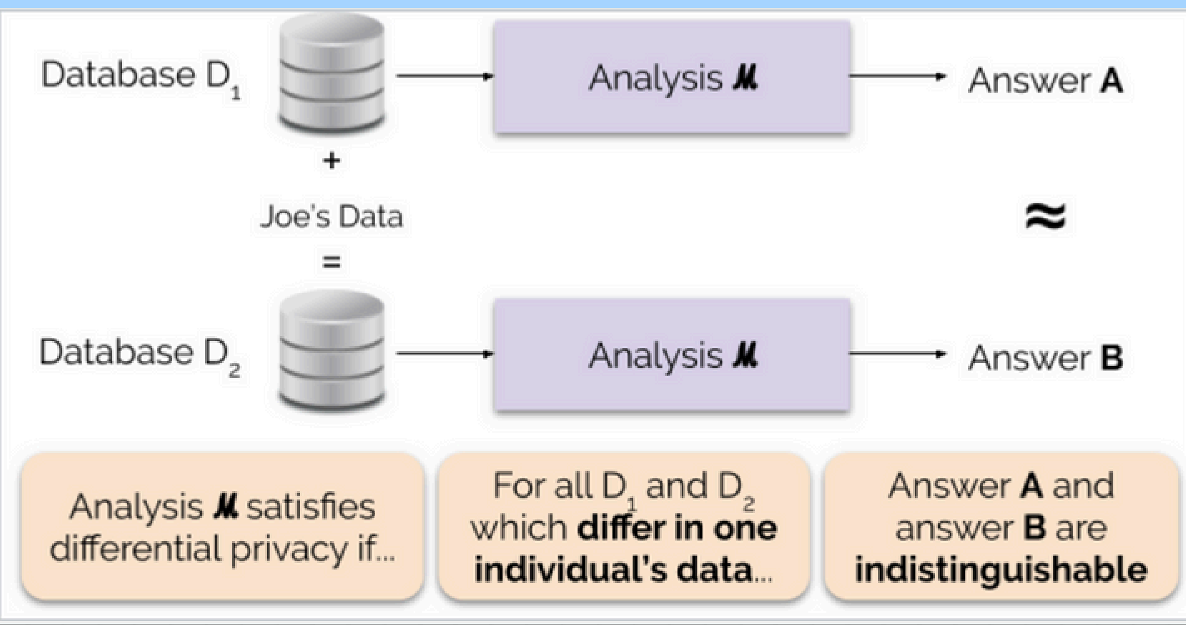
1. *ORAM (Oblivious RAM) hides which data the user is accessing from the server.*

2. *It works by reading and writing extra data and shuffling block locations after each access.*

3. *This makes every access look the same to the server, protecting user privacy.*

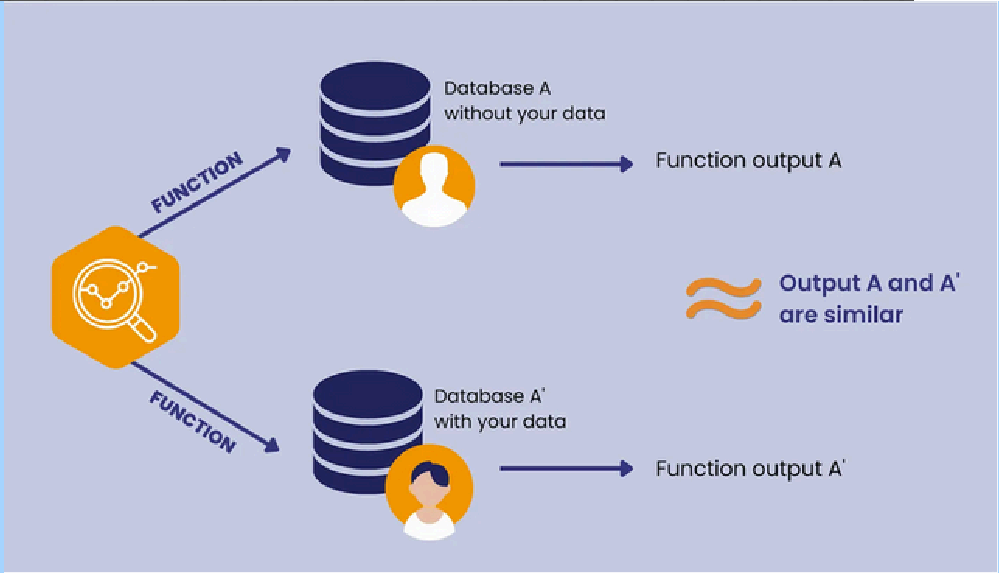
DIFFERENTIAL PRIVACY

- *Differential Privacy ensures that the output of a computation doesn't significantly change when a single individual's data is added or removed, protecting their presence.*
- *It is controlled by two parameters: ε (privacy loss) and δ (failure probability), where smaller values imply stronger privacy.*



DIFFERENTIAL PRIVACY EXAMPE

Suppose a researcher wants to know:
"Have you ever cheated in an exam?"
But of course, people may not feel safe answering honestly, even if the survey is anonymous.
So, to protect individuals' privacy, we uss the randomized response technique.

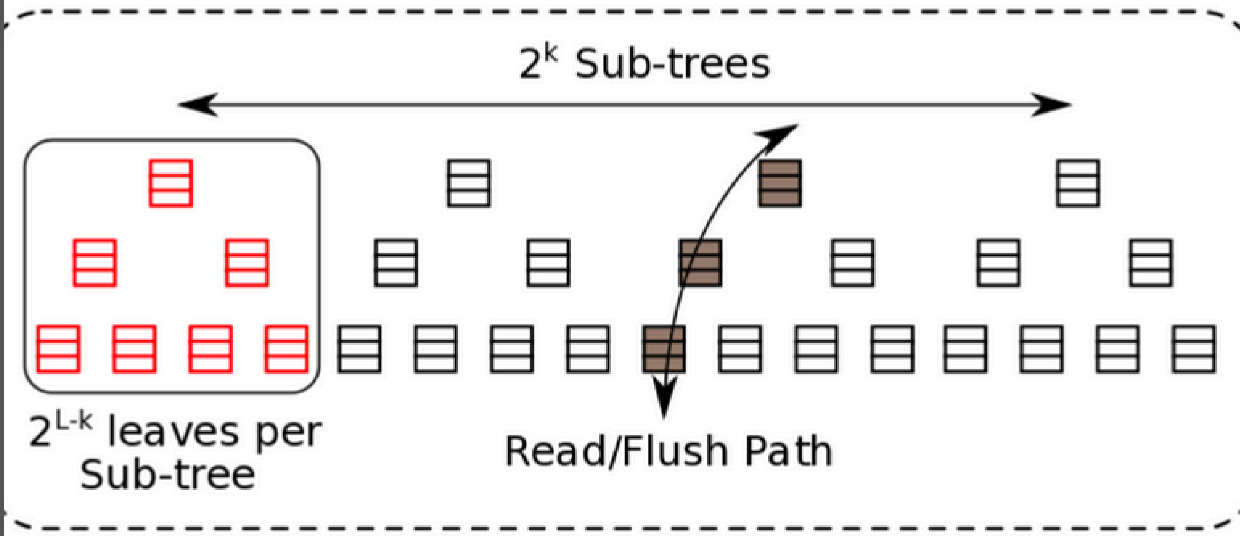


Randomized Response Mechanism:
Each person does the following privately:
Flip a coin.
If heads, answer truthfully.
If tails, flip the coin again:
If heads, answer "Yes".
If tails, answer "No".

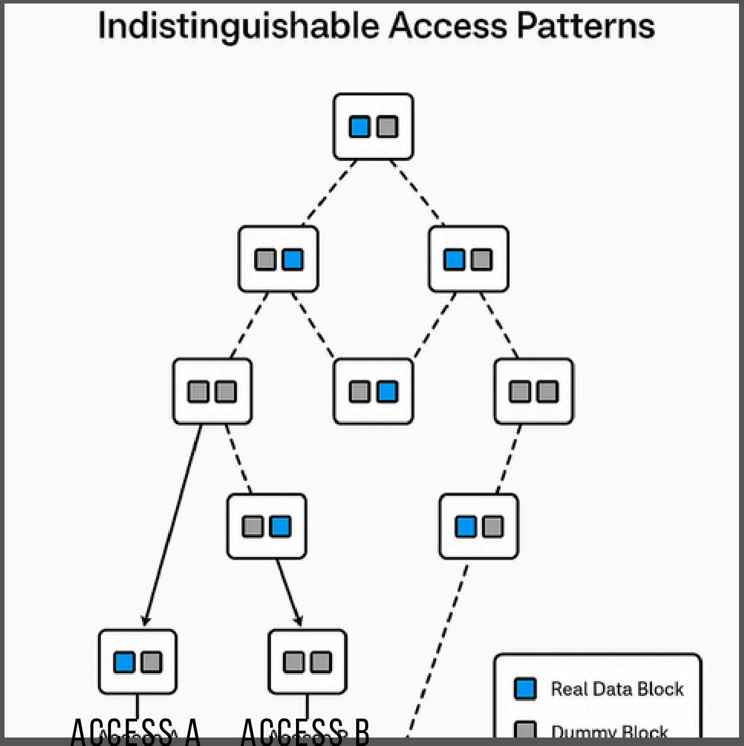
Let's assume 1000 people participate.
Because of the second coin flip (when the first is tails), random noise is introduced. So even if someone says "Yes", you can't tell if:
They actually cheated and told the truth, or
The randomization made them say "Yes".
This gives each participant plausible deniability.

ABOUT THE ROOT ORAM

- *Root ORAM splits the data into subtrees and stores them in separate roots. To access a block, the client downloads a path from the corresponding root to a leaf and reshuffles the block along a new random path to hide access patterns.*

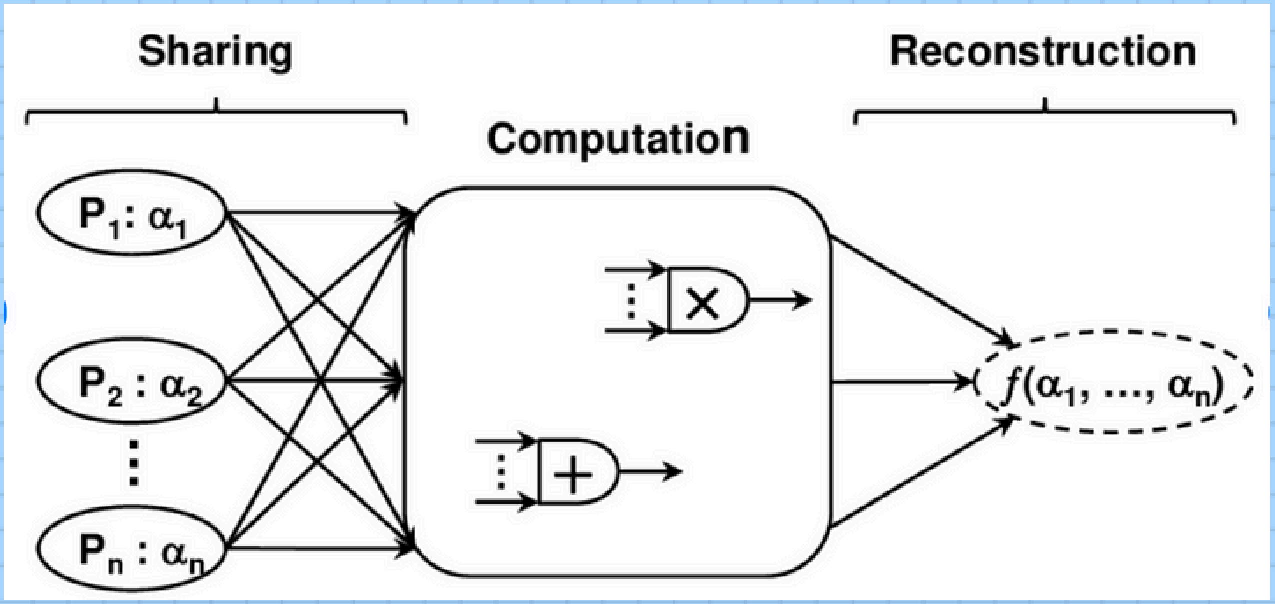


IS THE ROOT ORAM DIFFERENTIALLY PRIVATE ?



In differentially private ORAMs (like Root ORAM), this is done by using a differential privacy mechanism, which assigns higher probabilities to access paths that are closer to the previously used path, and lower probabilities to others, in a way that the overall leakage remains within a DP bound.

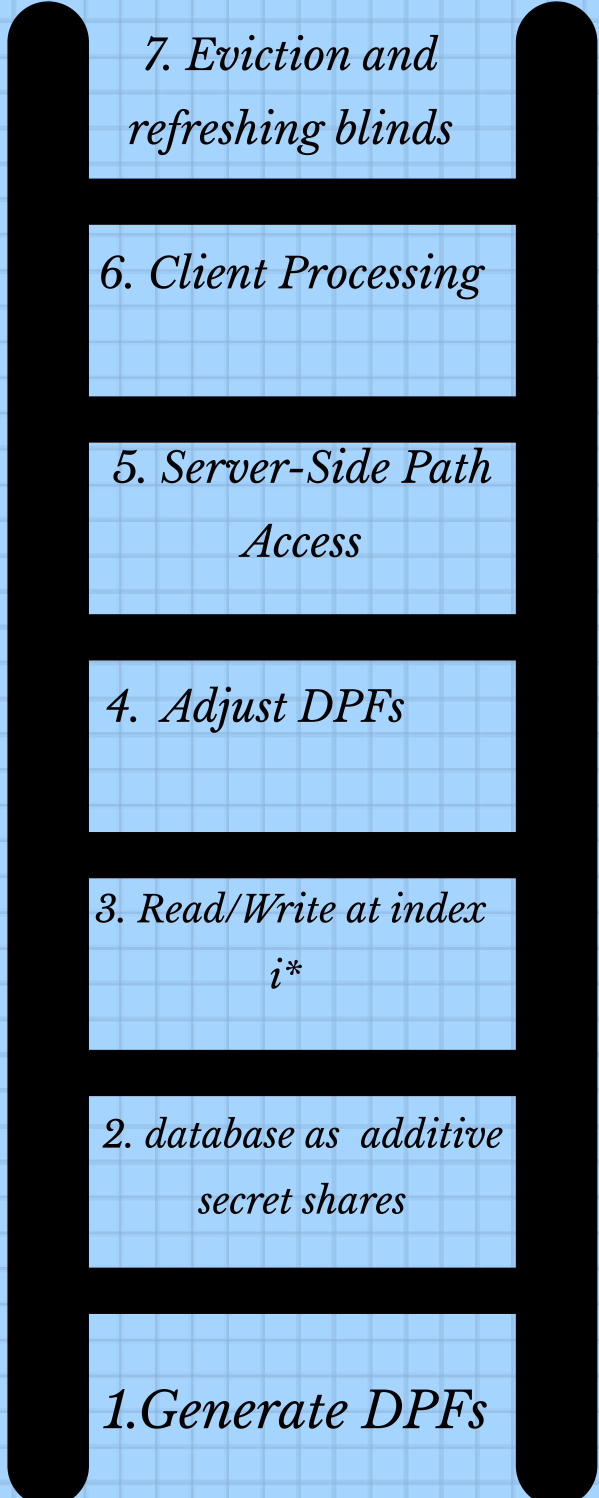
THE DISTRIBUTED ORAM



Multi-Party Computation (MPC):

- *Secure MPC allows multiple parties to jointly compute a function over their inputs without revealing those inputs to each other.*
- *Each party learns only the output—nothing more—ensuring privacy even if parties don't fully trust each other.*

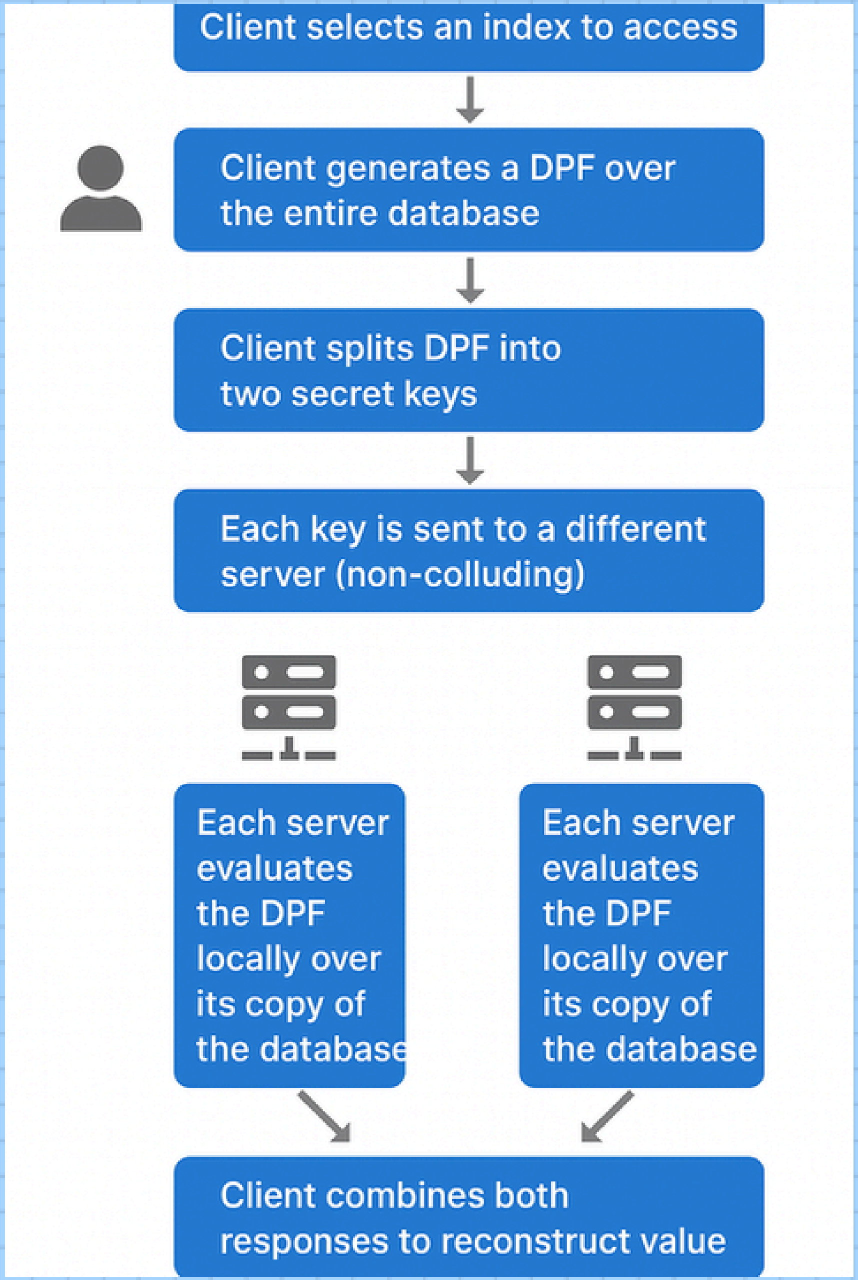
THE DISTRIBUTED ORAM



DUORAM = Distributed ORAM for MPC:

- *Specially designed for 2-party and 3-party secure computations.*
- *Stores memory as secret shares (no party knows the full data).*
- *Enables read/write operations with access patterns fully hidden.*

THE DISTRIBUTED ORAM



WHERE WE NEED TO FOCUS

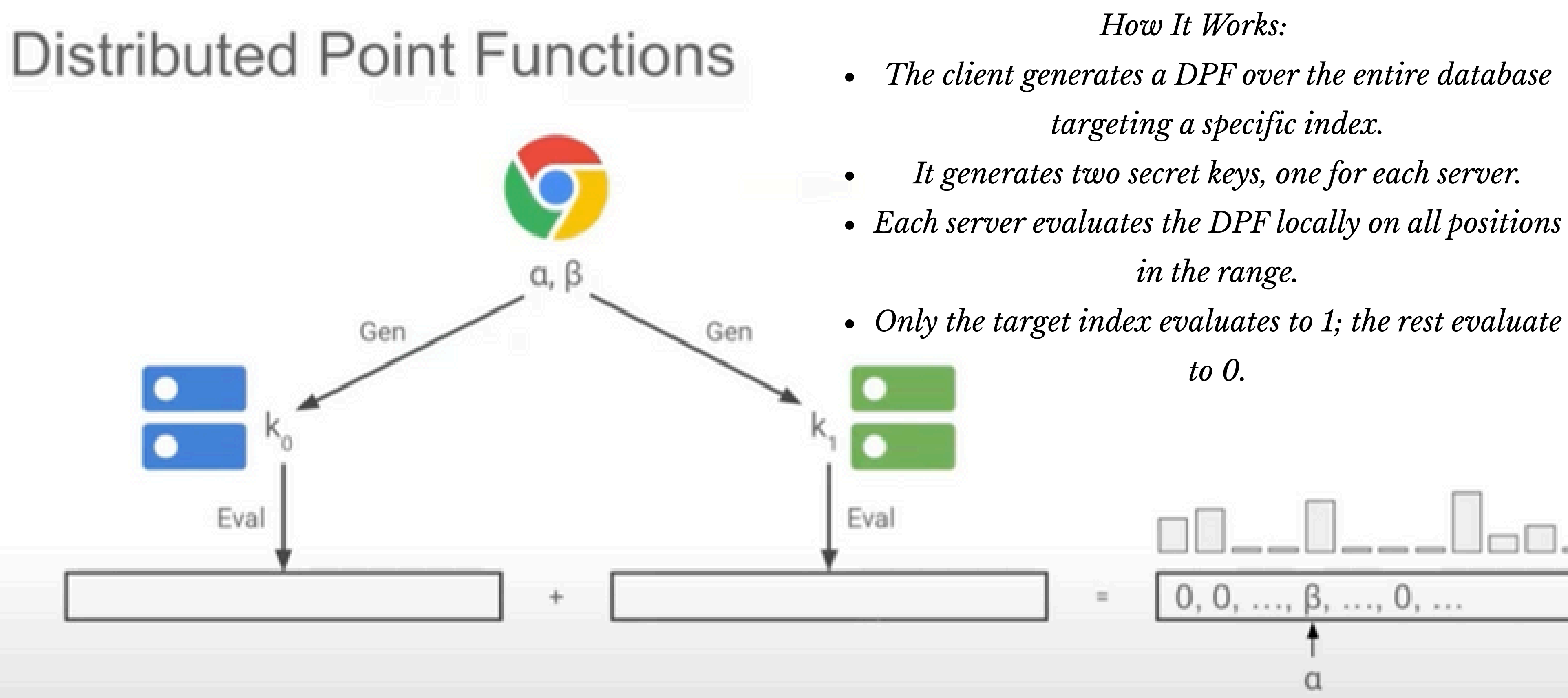
Memory Sharing:

Database is split into additive shares: Party P0 holds D_0, P1 holds D_1 ($D = D_0 + D_1$).

Key Technique:

Distributed Point Functions (DPFs)

Represent a single index access compactly. Allow reads/writes to be secret-shared and efficient.

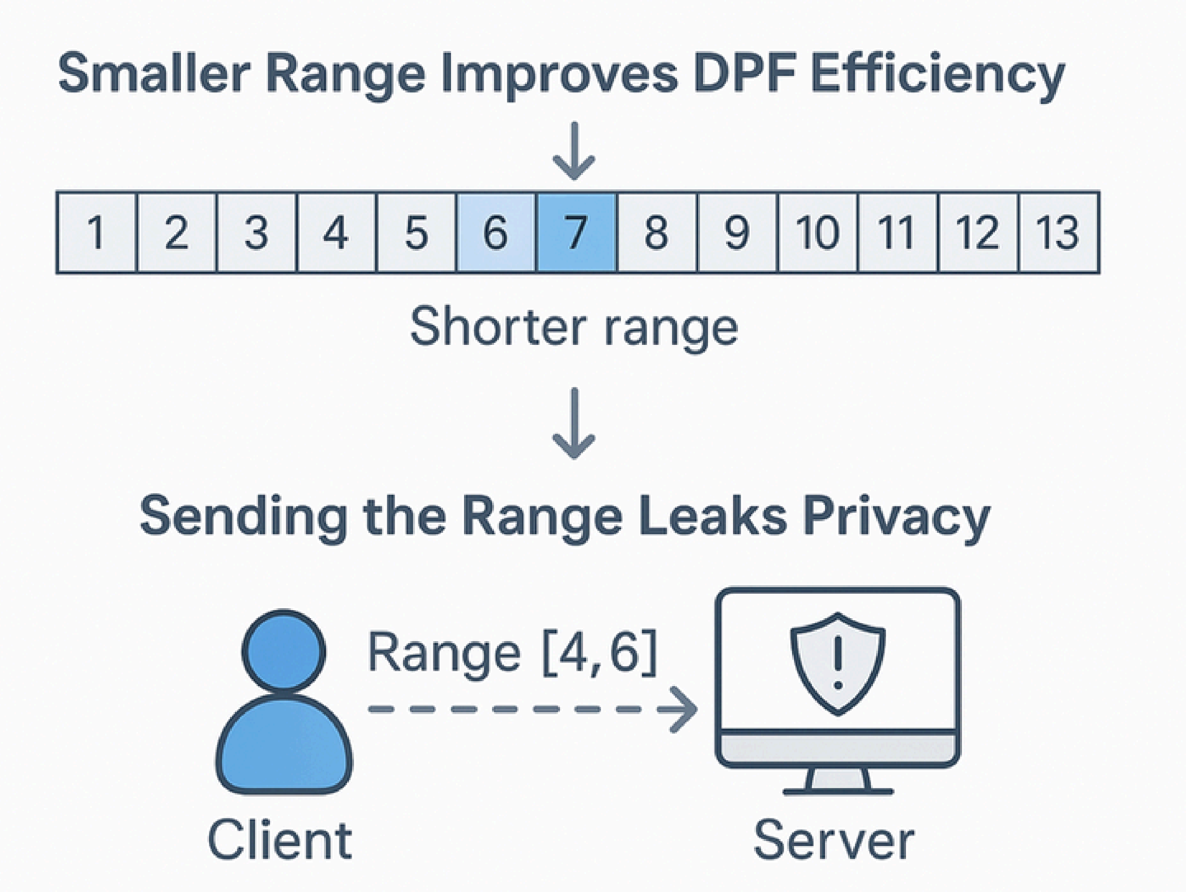


How It Works:

- *The client generates a DPF over the entire database targeting a specific index.*
- *It generates two secret keys, one for each server.*
- *Each server evaluates the DPF locally on all positions in the range.*
- *Only the target index evaluates to 1; the rest evaluate to 0.*

DISTRIBUTED POINT FUNCTION (DPF)

RETHINKING DPF USAGE FOR EFFICIENT DUORAM



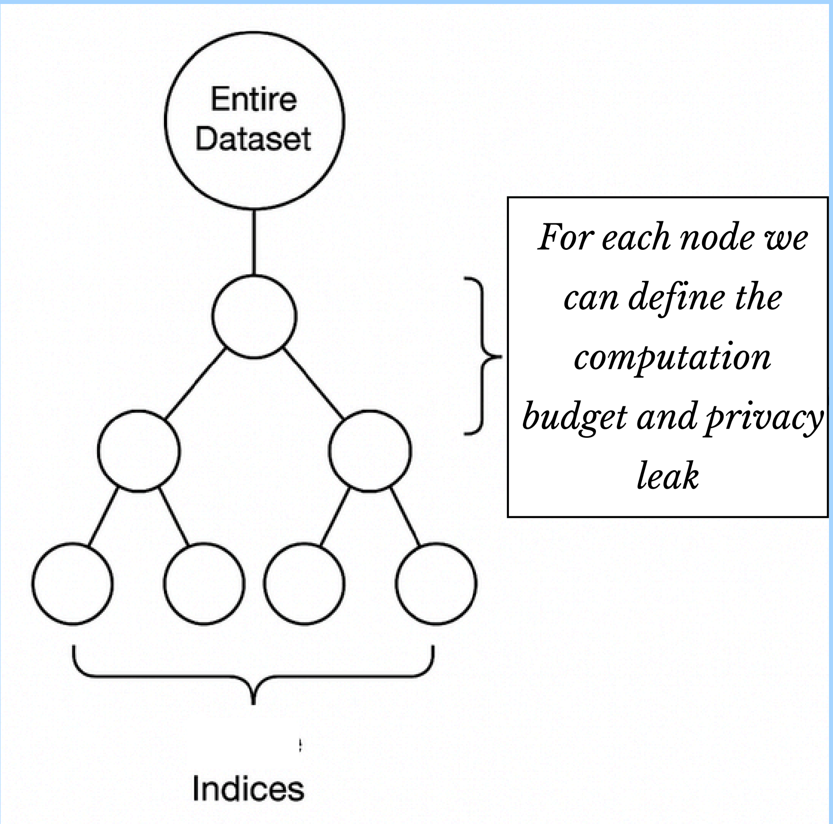
Efficiency Gains

- *DPF generation and evaluation cost is linear in the range size (O(s)).*
- *Smaller ranges significantly speed up computation.*

Privacy Leakage

- *Sending a small range with the secret key reveals that access lies within it.*
- *Smaller ranges improve efficiency but increase the adversary's chance of guessing the target.*

DPFS WITH SMALLER RANGE



Organize n indices in a binary tree (depth = $\log_2 n$)

Each node = a possible query range

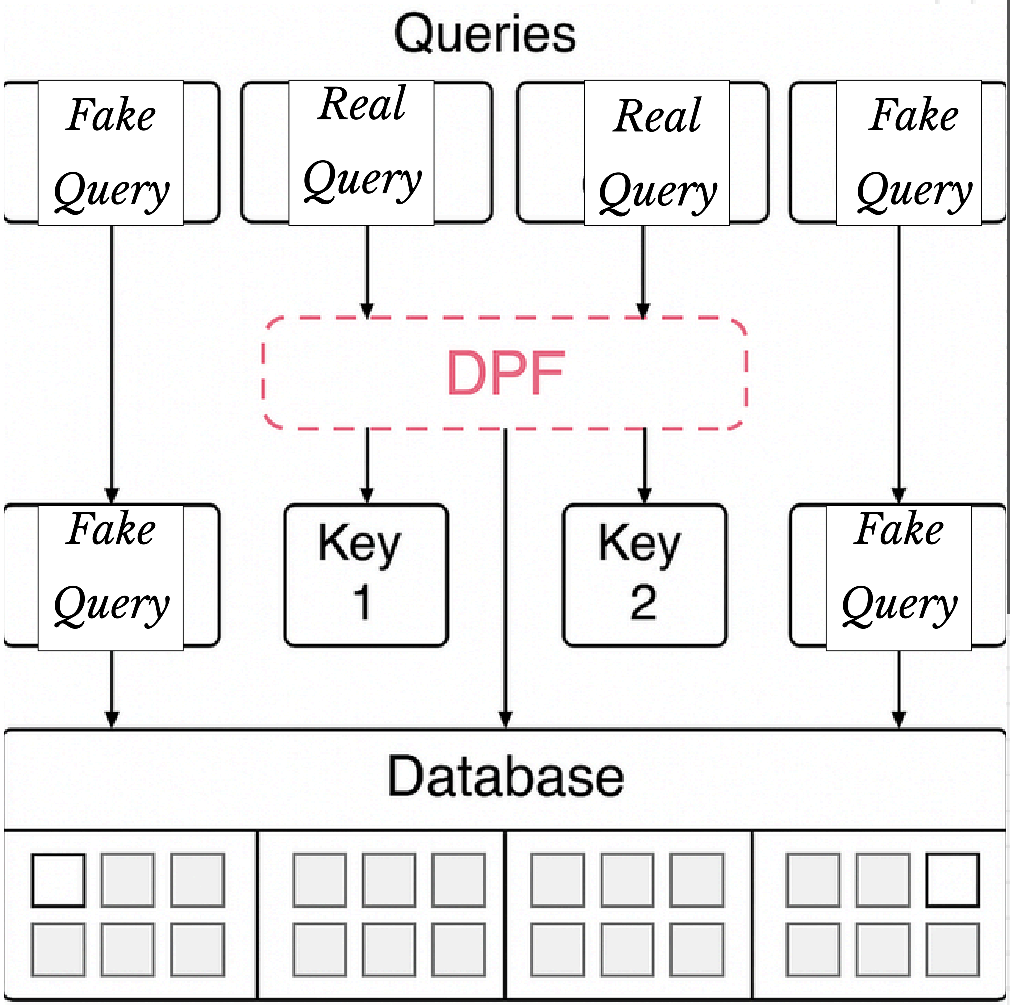
Client identifies which index they want to query

Choose a Range (Node on Path to Root)

Compute DPF over Range

Ensure Constraints Hold & Send DPF Key + Range → Server

PREVIOUS ATTEMPT: INTRODUCING FAKE QUERIES ALONG WITH SHORT RANGE



How It Works:

- *→ Insert fake queries alongside real queries*
- *→ Divide database into smaller segments*
- *→ Build DPFs only over accessed segments*
- *→ Adjust number of fake queries and segment size*

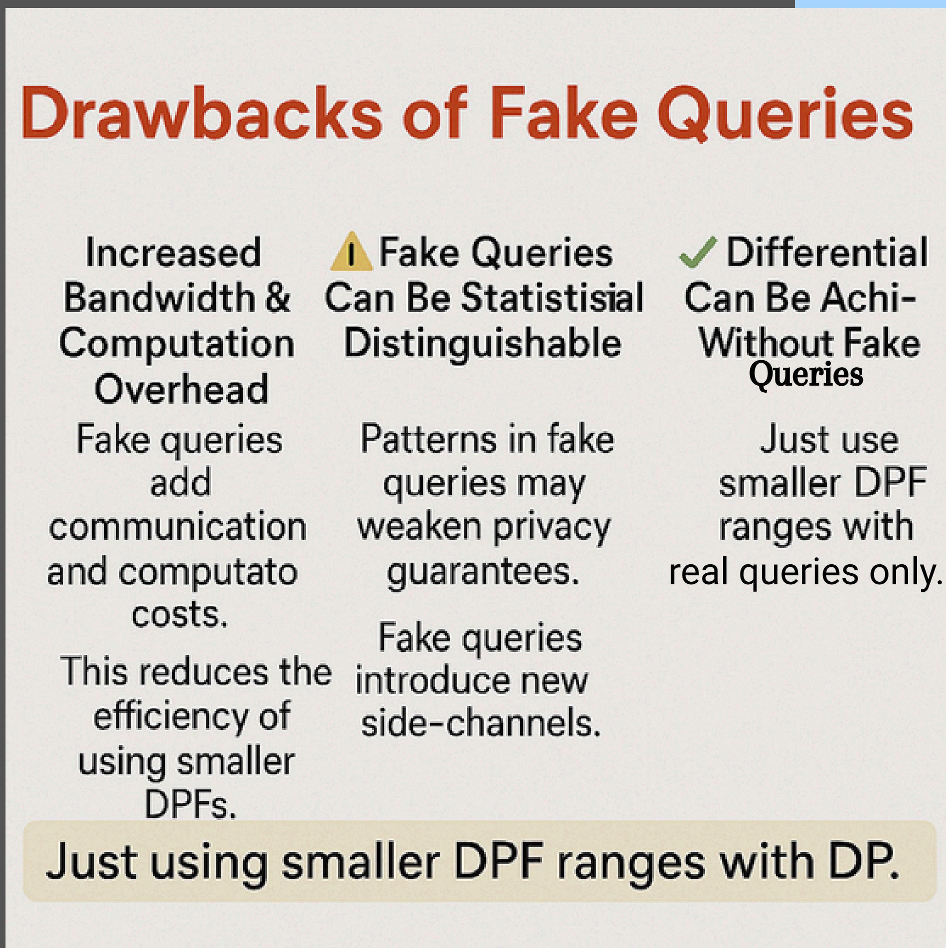
PROBLEM WITH THE PREVIOUS ATTEMPT

- *Fake queries increase communication and computation overhead, reducing the performance gains from smaller DPF domains.*

- *Deterministic fake queries reveal patterns, enabling the server to infer them over time.*

- *Smaller, randomized DPF ranges with DP noise can achieve (ϵ,δ)-privacy more efficiently, without the need for fake queries.*

PROBLEM WITH THE PREVIOUS ATTEMPT



WHAT THE FUTURE HOLDS

→ *Develop smarter algorithms to dynamically choose range sizes based on real-time privacy and efficiency needs.*

→ *Optimise server-side processing for even faster queries.*

→ *Explore advanced differential privacy mechanisms to further reduce leakages.*

References

Duoram: A Bandwidth-Efficient Distributed ORAM for 2- and 3-Party Computation
Adithya Vadapalli, Ryan Henry, Ian Goldberg

Differentially Private Oblivious RAM
Sameer Wagh, Paul Cuff, and Prateek Mittal*

Batched Differentially Private Information Retrieval
Kinan Dak Albab,Rawane Issa and Mayank Varia, Kalman Graffi

THANKS