# PRIVATE INFORMATION RETRIEVAL

**Alex Davidson**

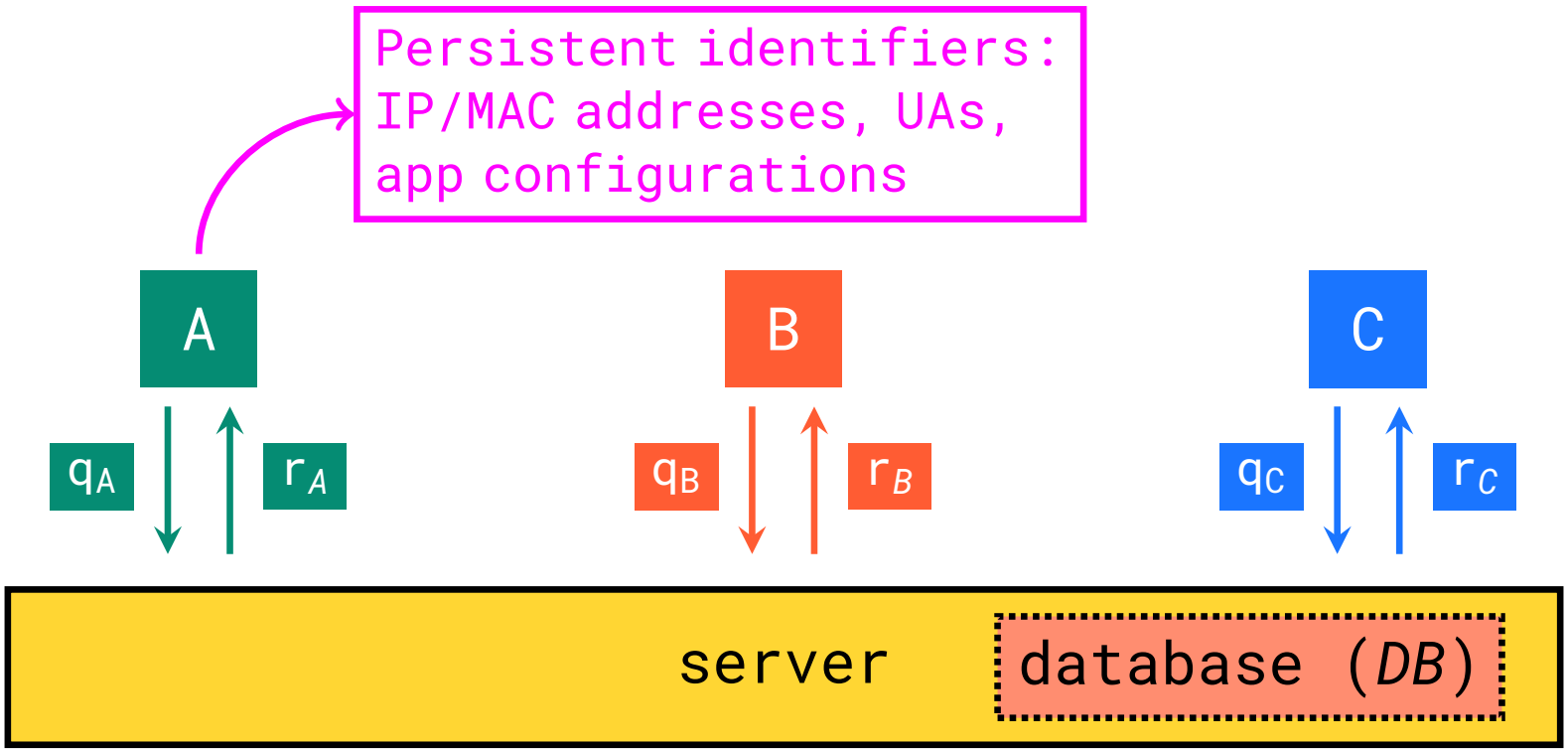
Universidade NOVA de Lisboa

**ASCRYPTO 2023 @ LATINCRYPT**

3rd October 2023

a.davidson@fct.unl.pt


ACK: Some inspiration from Dima Kogan’s talk from BIU Winter School 2022




- Communication channels are secure
- Server learns from each client interaction

## CLIENT-SERVER INTERACTIONS

Sacramento, California

147 languages

Article Talk Read Edit View history Tools

From Wikipedia, the free encyclopedia

Coordinates: 38°34′54″N 121°29′40″W

*"Sacramento" redirects here. For other uses, see Sacramento (disambiguation).*

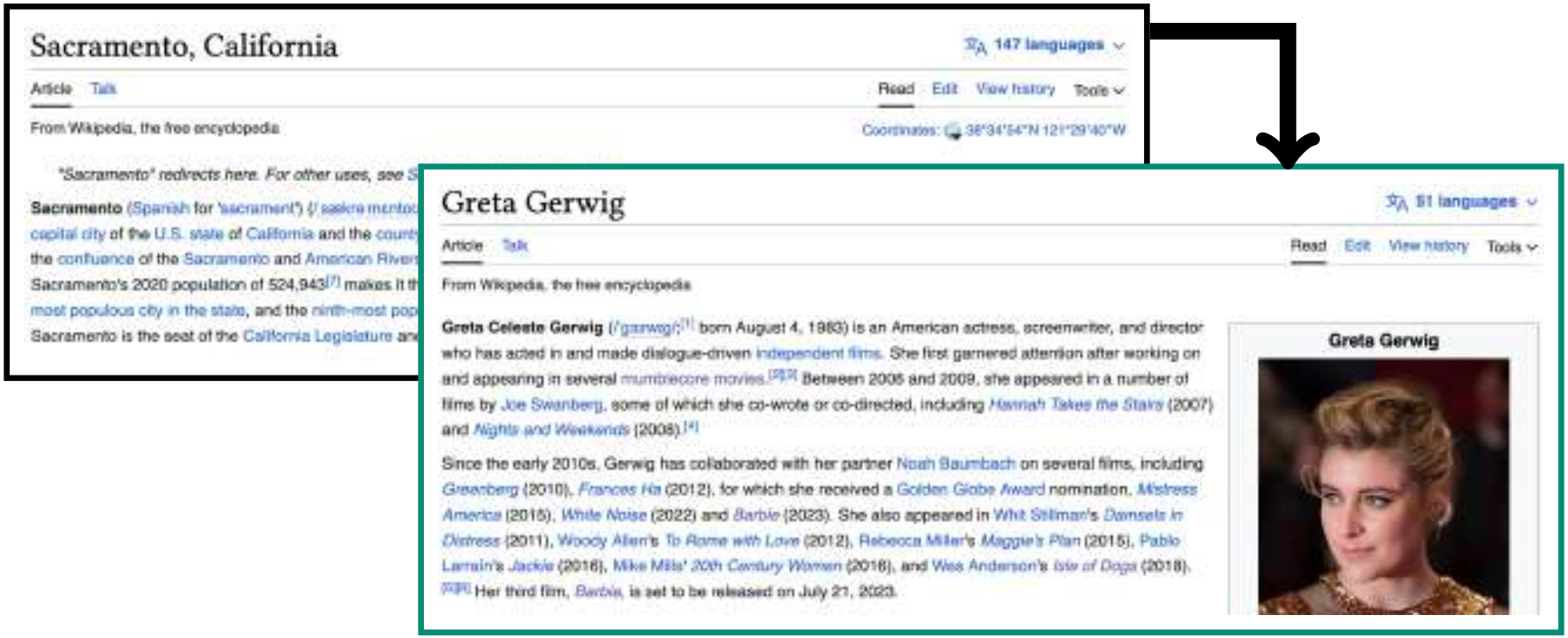
**Sacramento** (Spanish for 'sacrament') (/ˌsækrəˈmɛntoʊ/ *SAK-rə-MEN-toh*; Spanish: [sakraˈmento]) is the capital city of the U.S. state of California and the county seat of the county of the same name. Located at the confluence of the Sacramento and American Rivers in Northern California's Sacramento Valley, Sacramento's 2020 population of 524,943[7] makes it the fourth-largest city in Northern California, sixth-most populous city in the state, and the ninth-most populous state capital in the United States.[8][10] Sacramento is the seat of the California Legislature and the Governor of California.

Sacramento

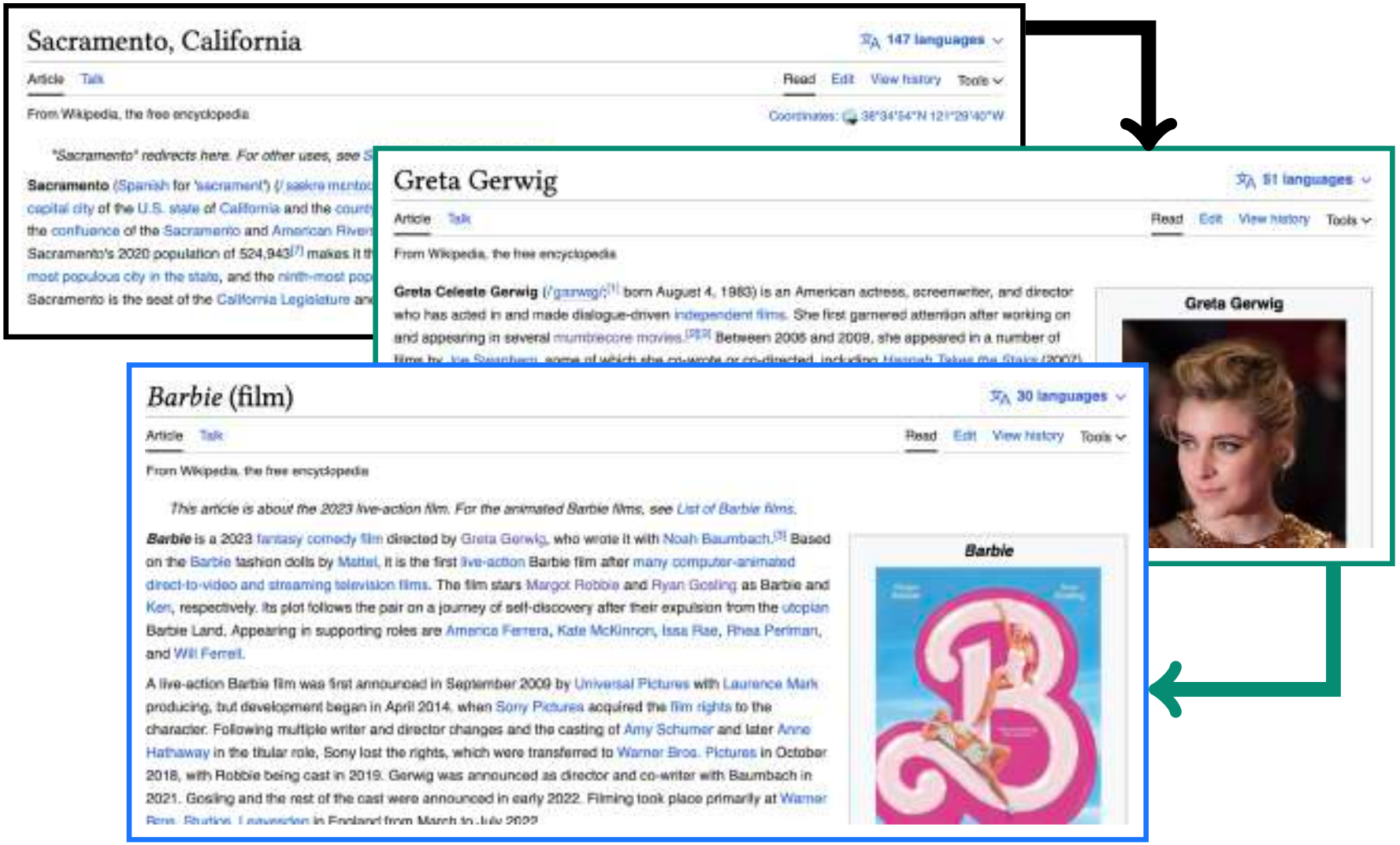
State capital city



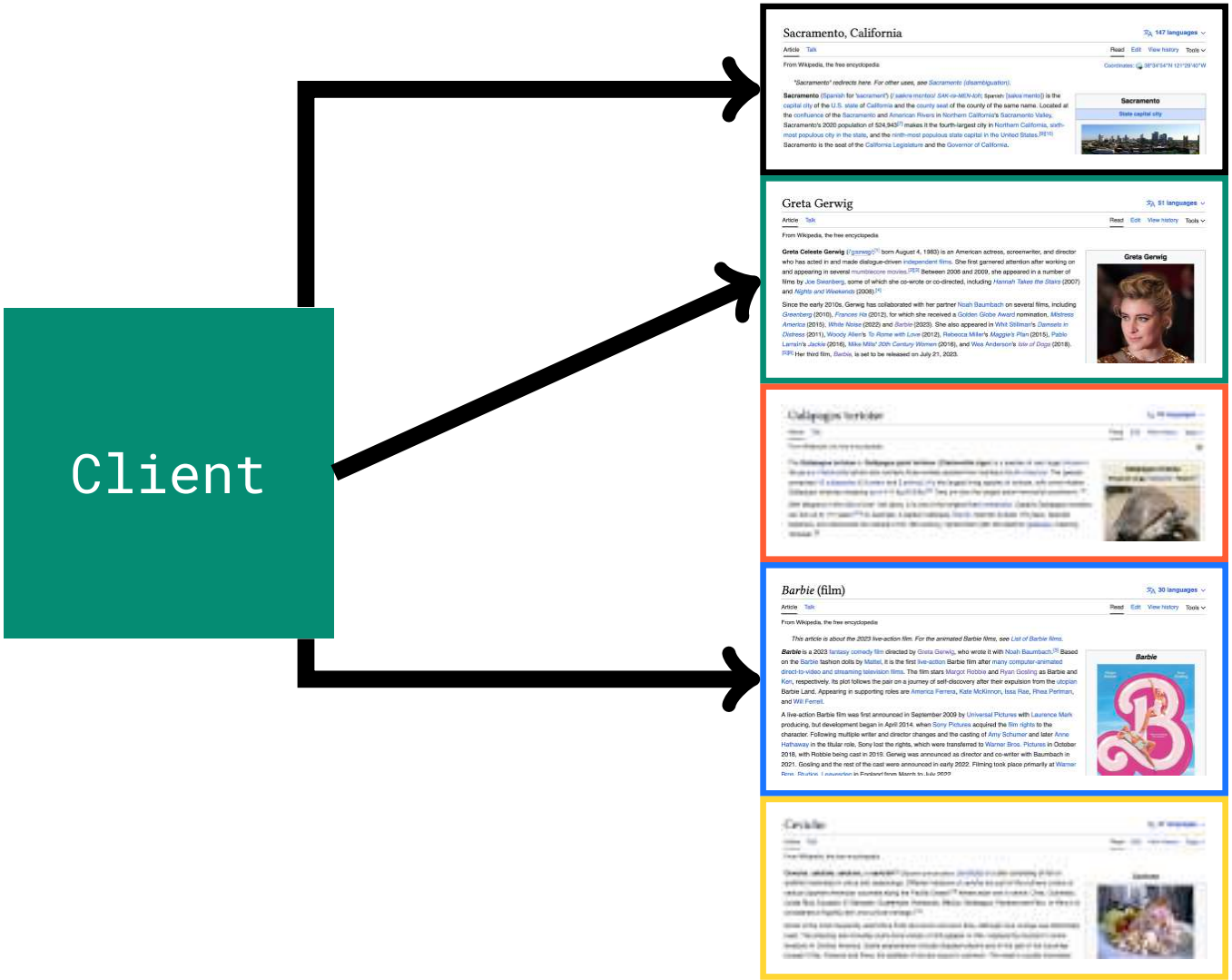
# DATABASE ACCESS




# DATABASE ACCESS

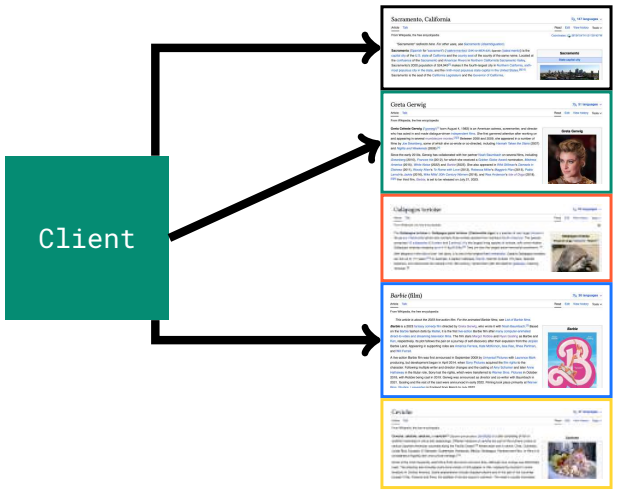



# DATABASE ACCESS

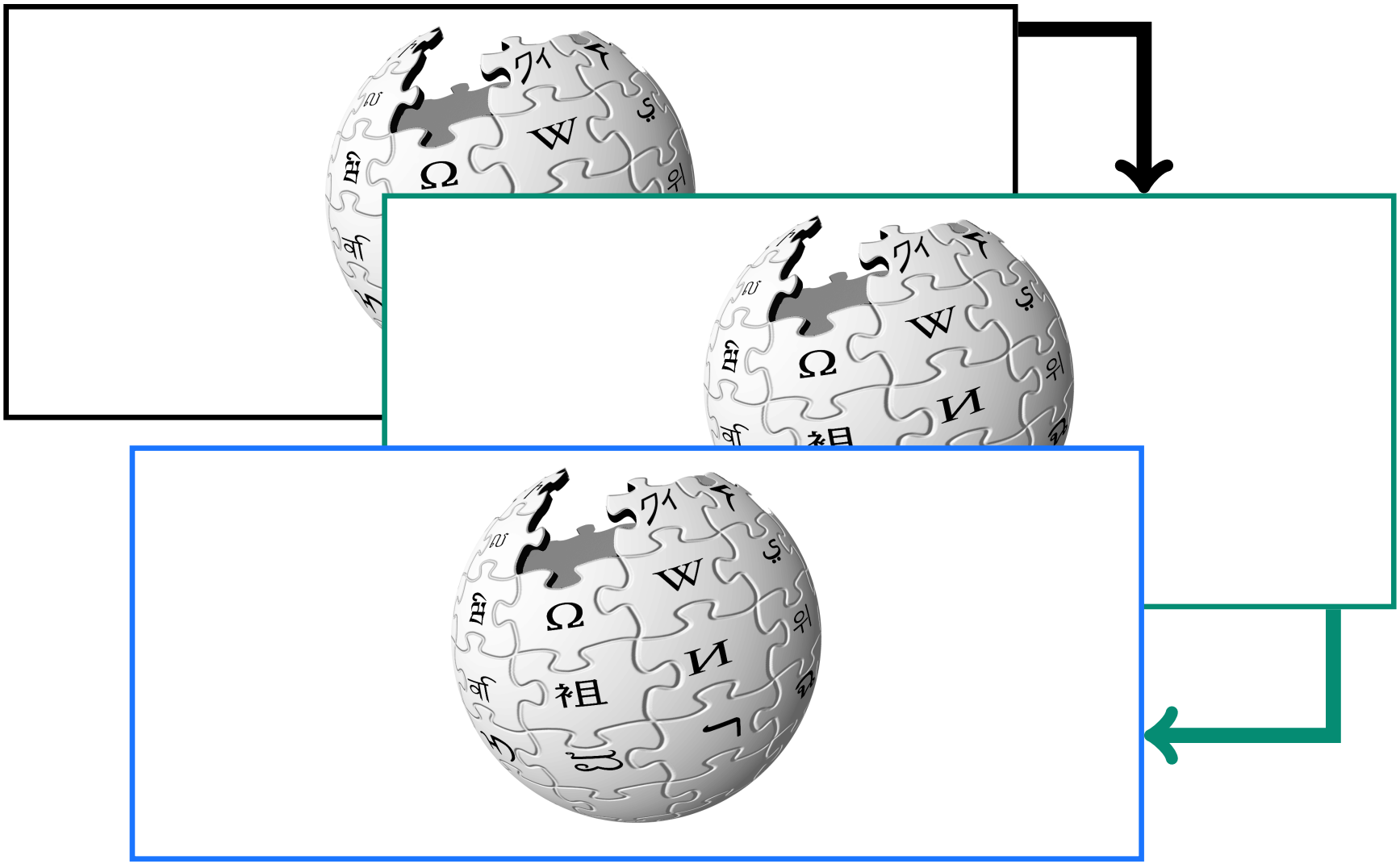



# PRIVACY IMPLICATIONS

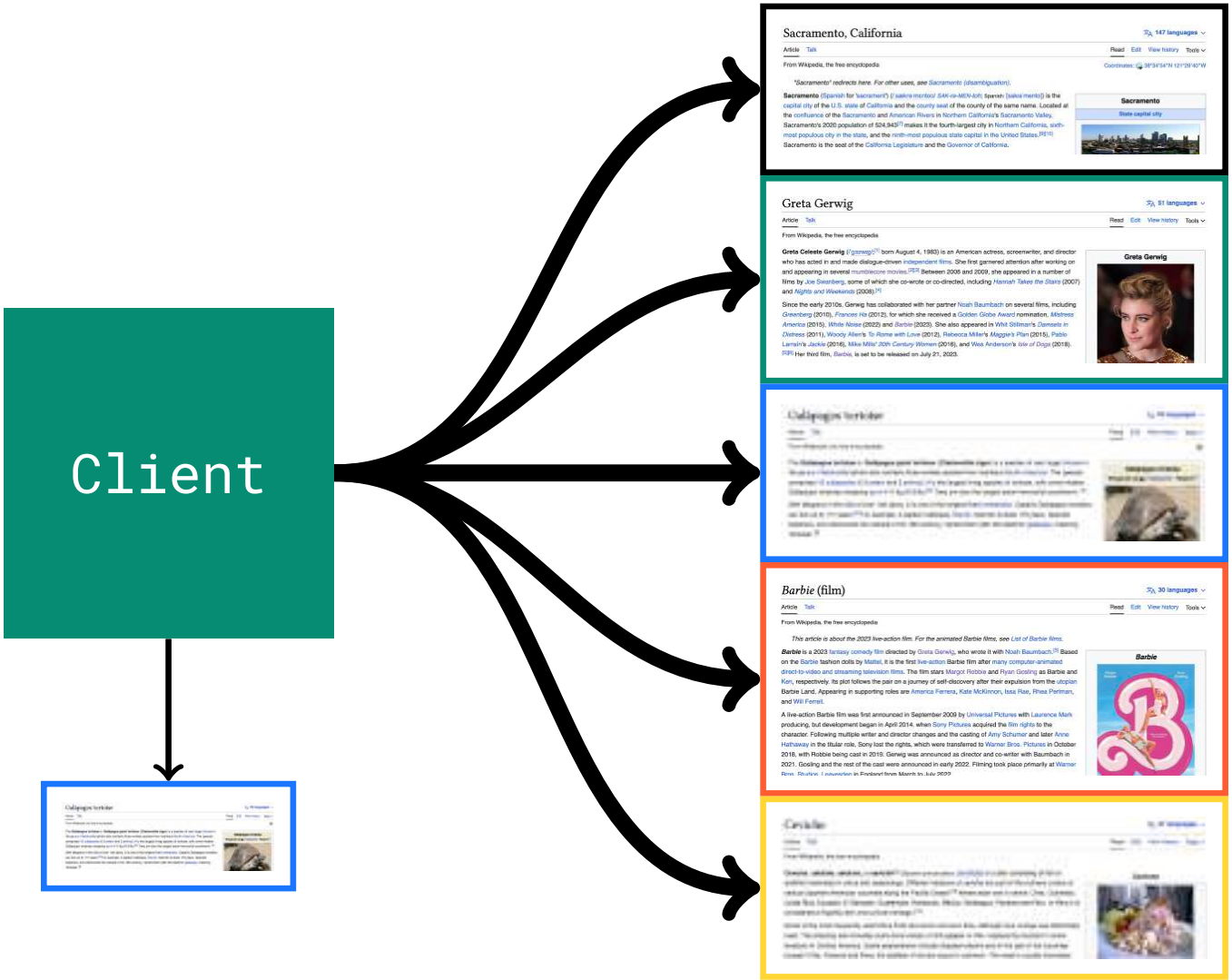
- Database holder learns access patterns
- Can be used to infer:
  - Behavioural trends
  - Innate preferences
  - Personal information (e.g. medical diagnoses)
  - more...




## PRIVACY IMPLICATIONS

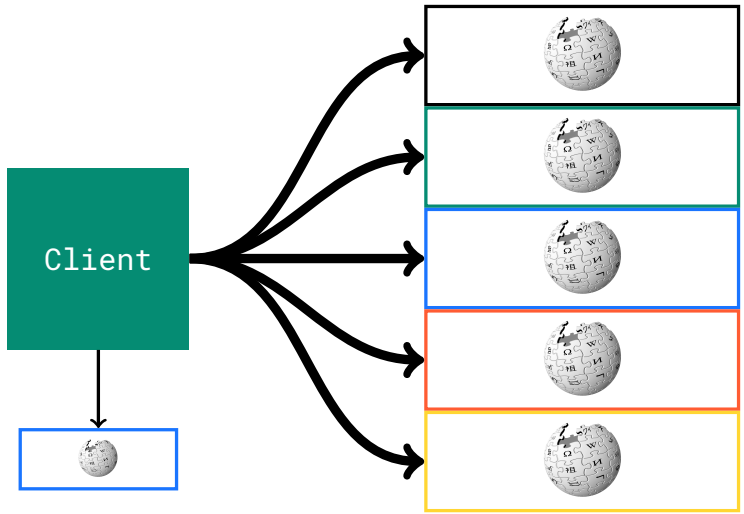


# IDEAL CASE




# PRIVATE INFORMATION RETRIEVAL: PIR

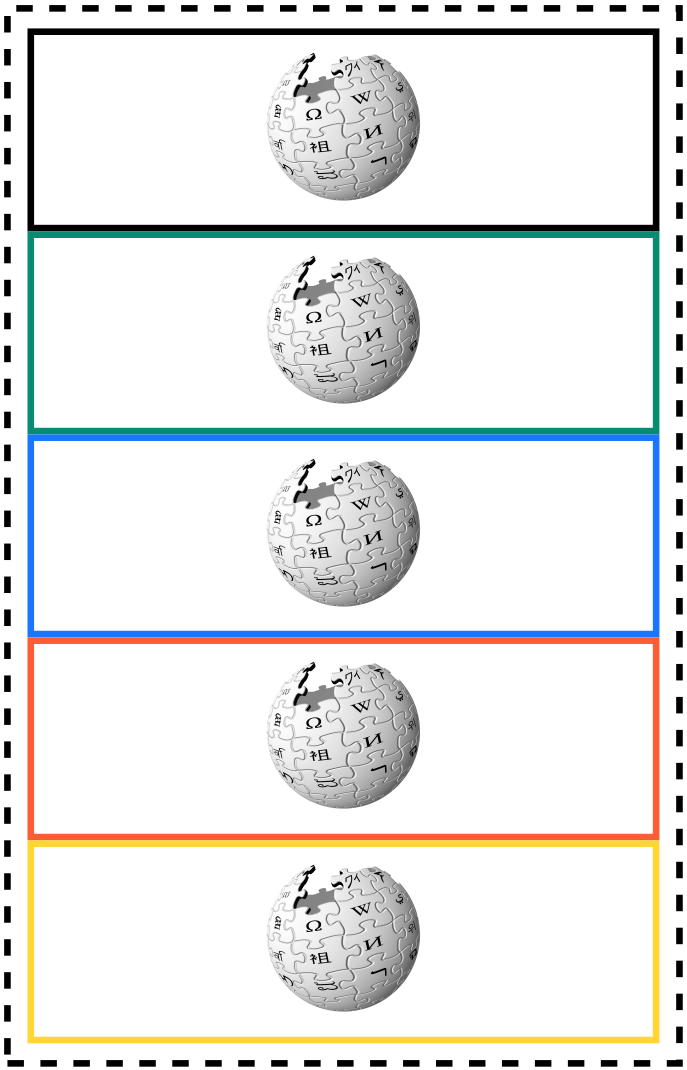
- Hides access patterns
- Ensures database holder learns nothing about the user's queries




# PRIVATE INFORMATION RETRIEVAL: PIR

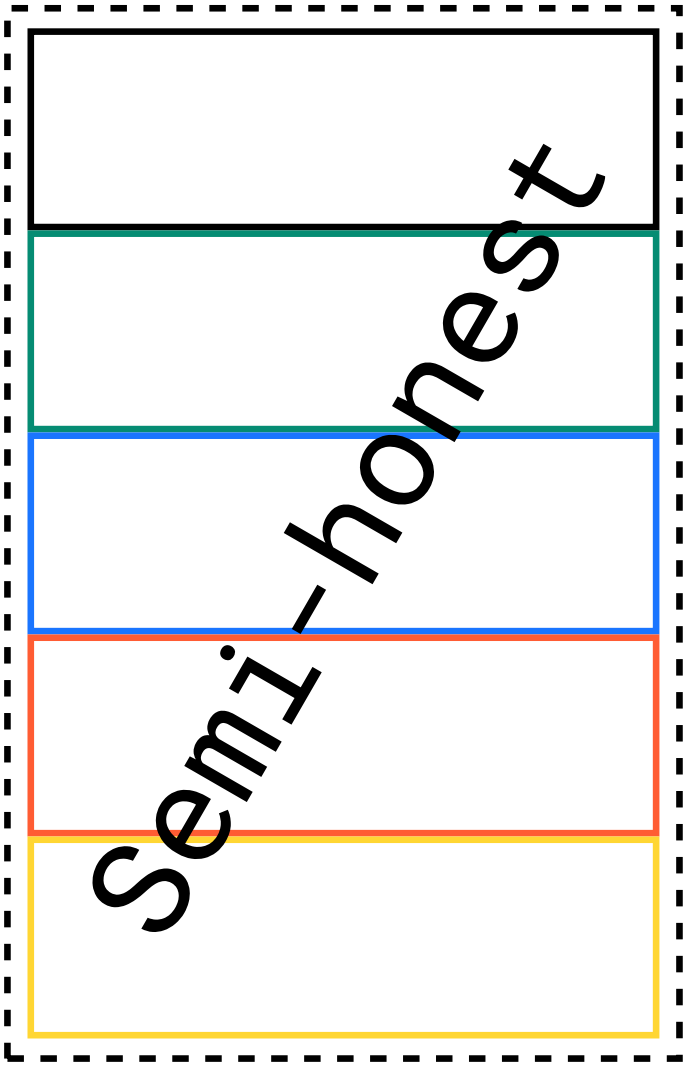
Client

Public

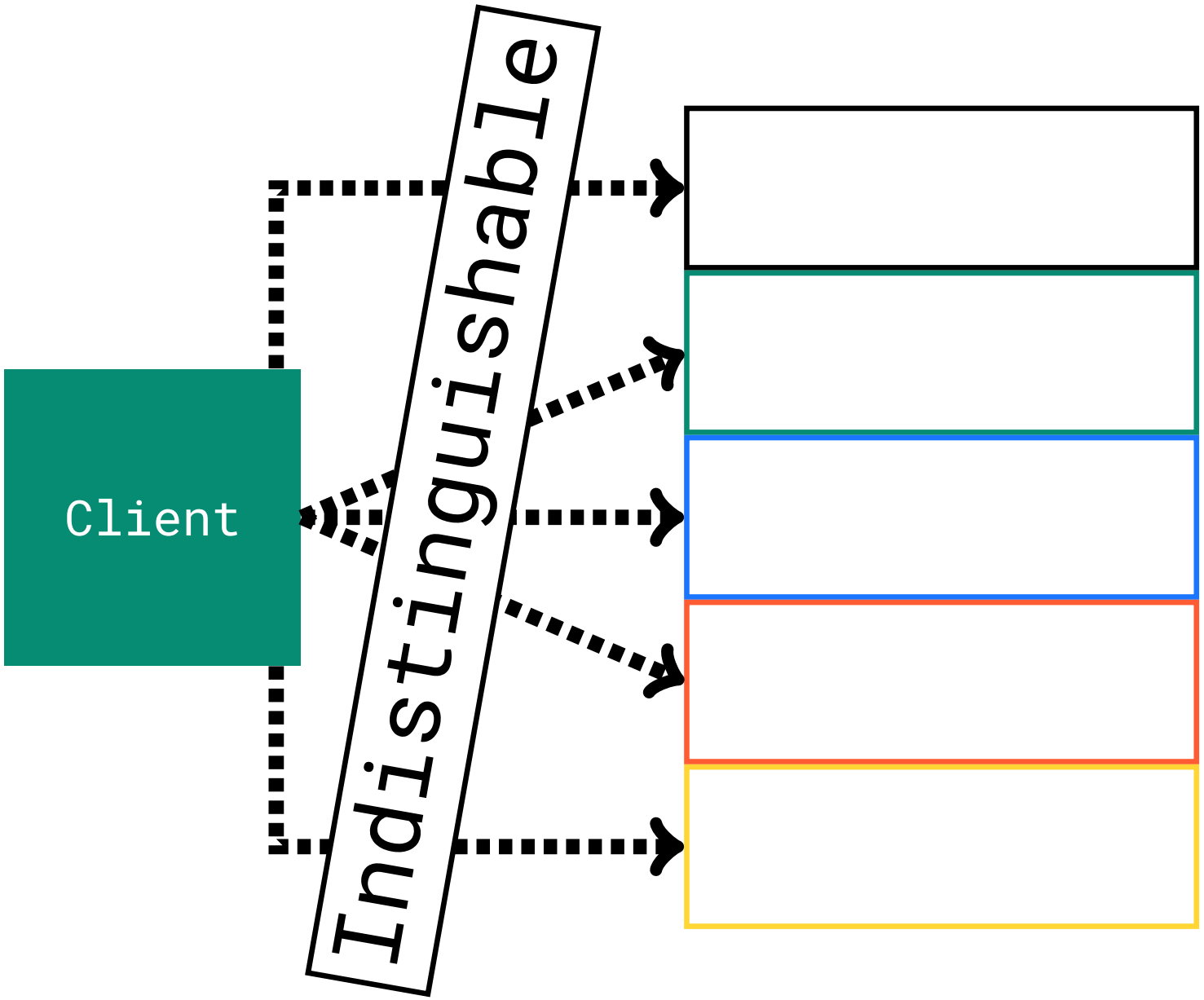


## THREAT MODEL: DATABASE

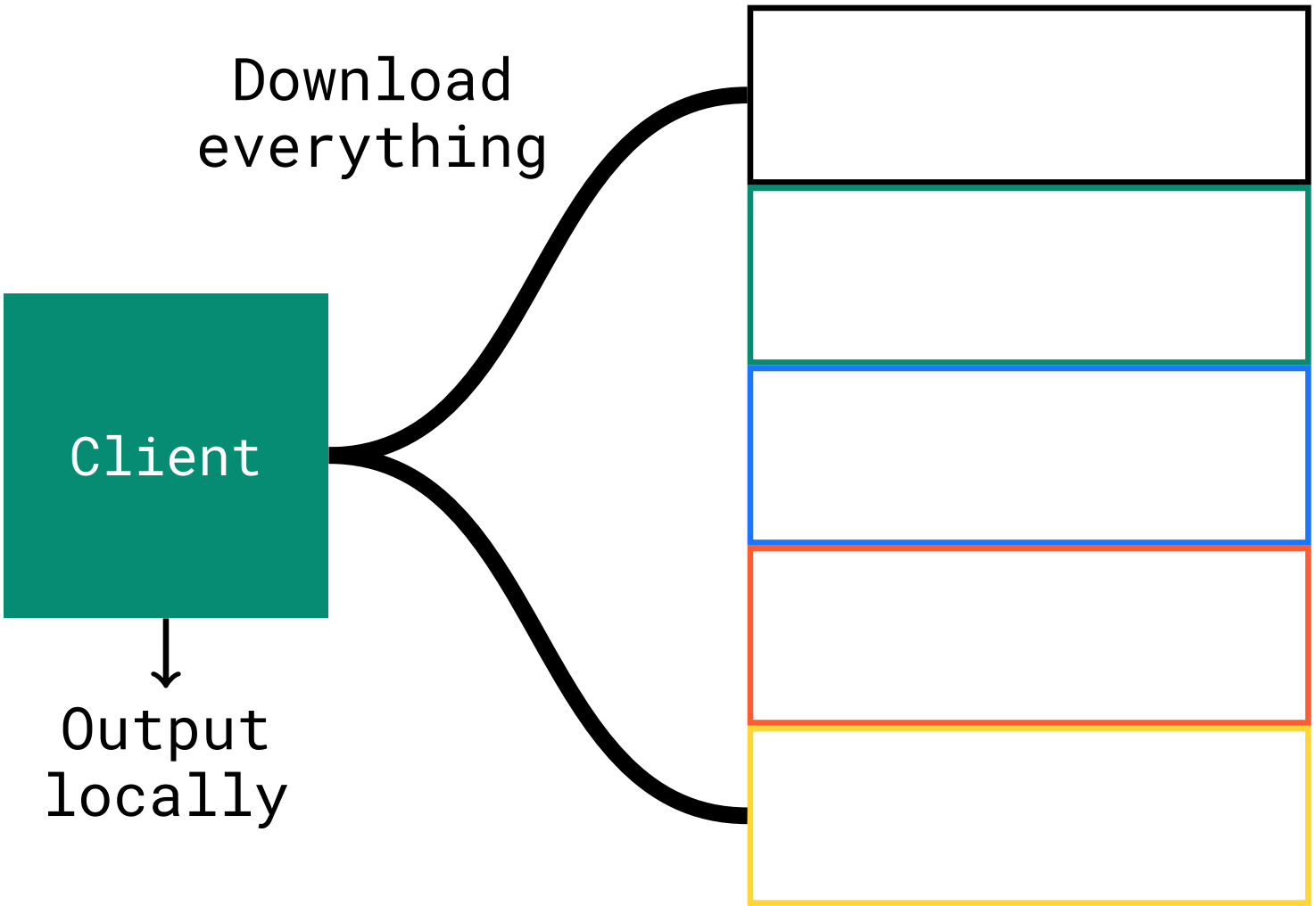
Client



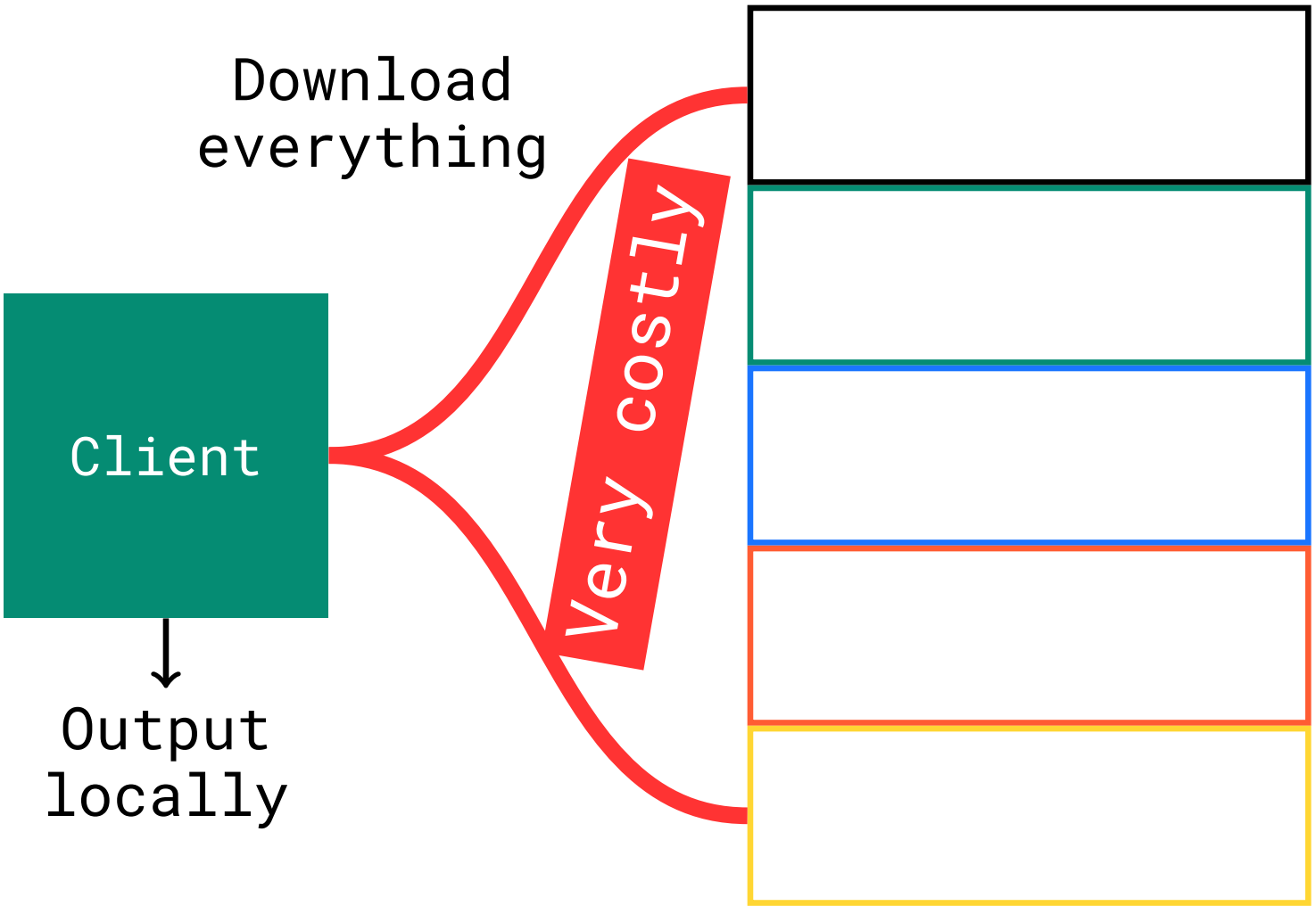

# THREAT MODEL: SERVER



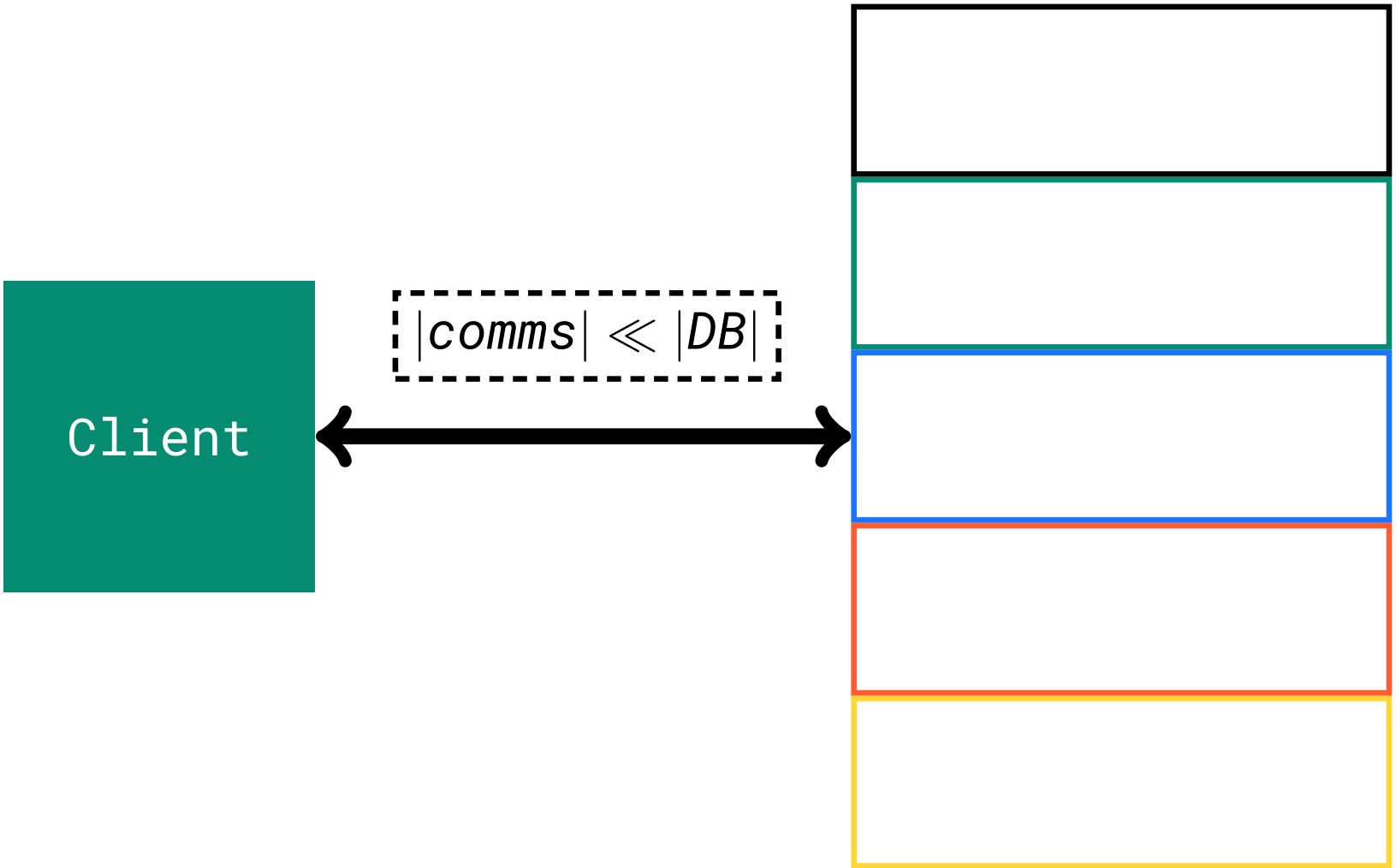

## SECURITY GUARANTEE



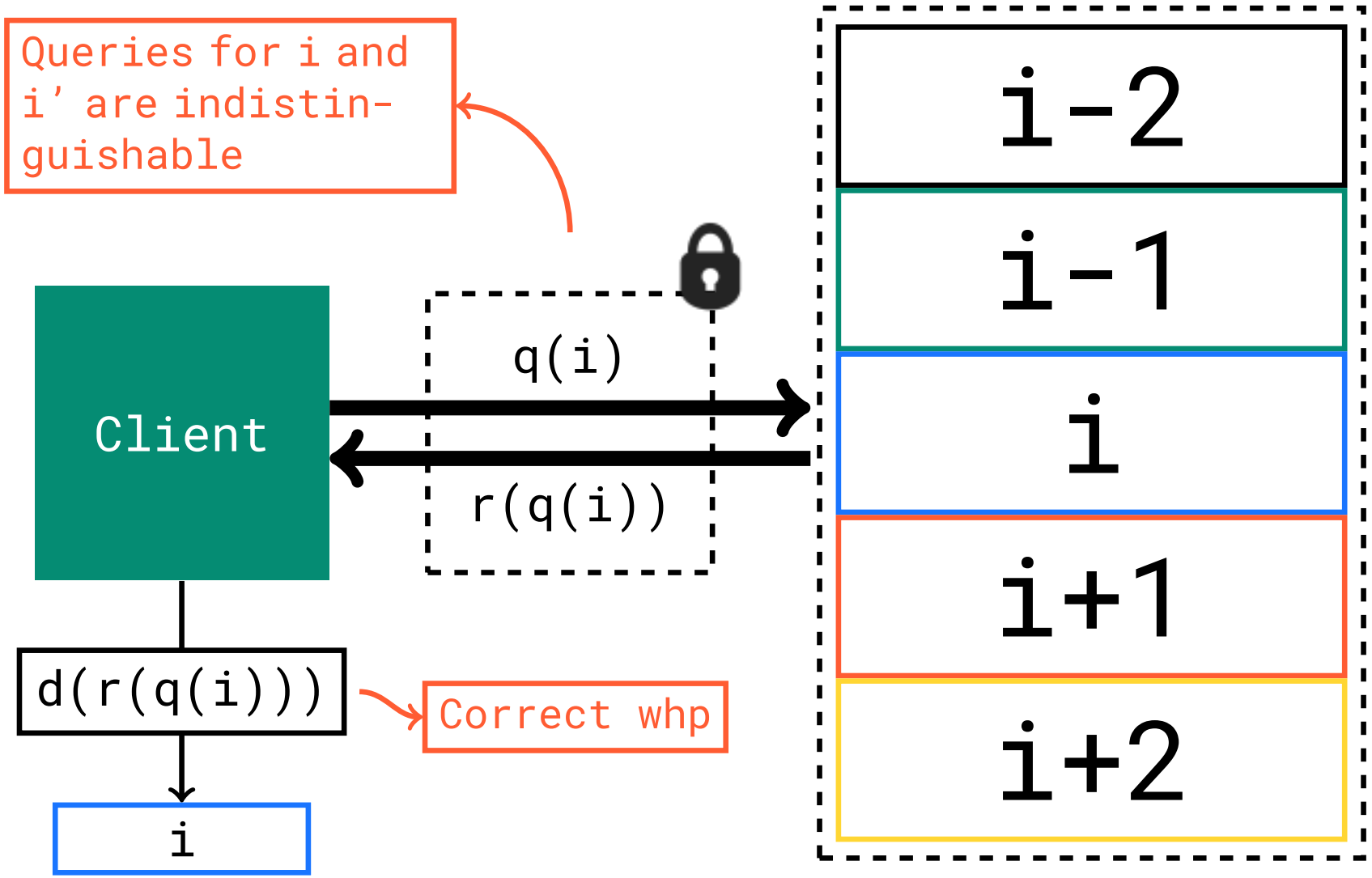

# TRIVIAL SOLUTION




# TRIVIAL SOLUTION




# EFFICIENCY MODEL



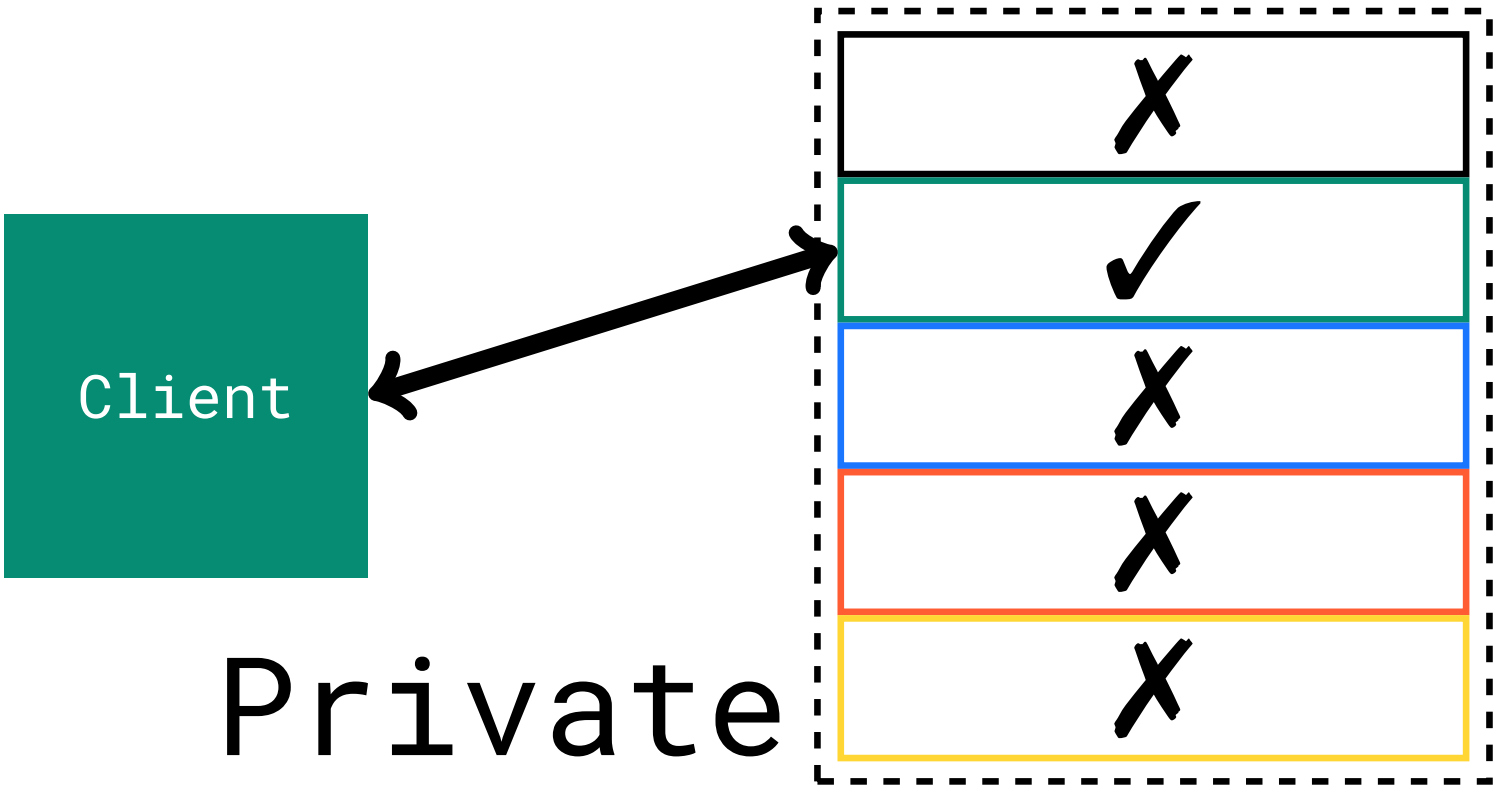

# ALGORITHMIC FLOW

Any that involve accessing a database...

- Private contact list discovery
- Blocklist queries (e.g. SafeBrowsing)
- Compromised credential checking
- Private preference matching

## APPLICATIONS

1-out-of-$n$ OT $\equiv$ **Symmetric** PIR




## RELATION TO OBLIVIOUS TRANSFER

## Similarities

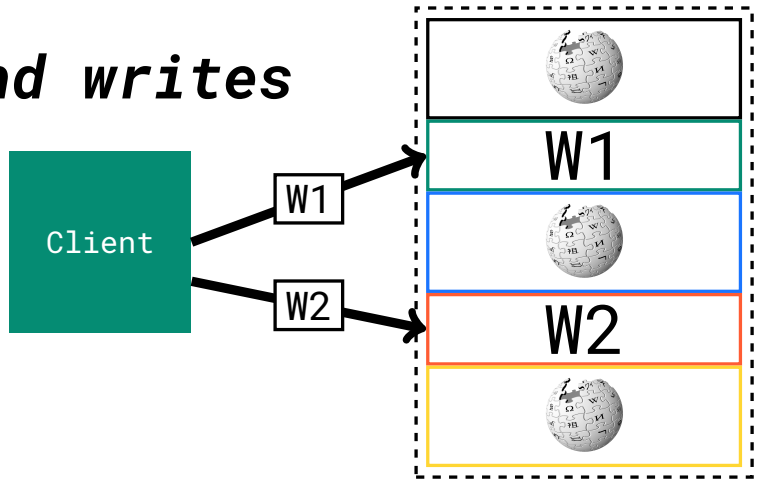
- Both PIR and ORAM hide access patterns

## Differences

- ORAM only allows a single client

*Requires private state to be shared*

- ORAM permits reads ***and writes***




# RELATION TO OBLIVIOUS RAM

# What do you remember?

¡1¡¡QUIZ!!1!

In standard PIR, the database is considered **public**.

- True
- False

## QUESTION #1

In standard PIR, the database is considered **public**.

- True
- False

ANSWER #1

From a client query, the server learns:

- Nothing
- 1 bit of information
- The entire query

## QUESTION #2

From a client query, the server learns:

- Nothing
- 1 bit of information
- The entire query

## ANSWER #2

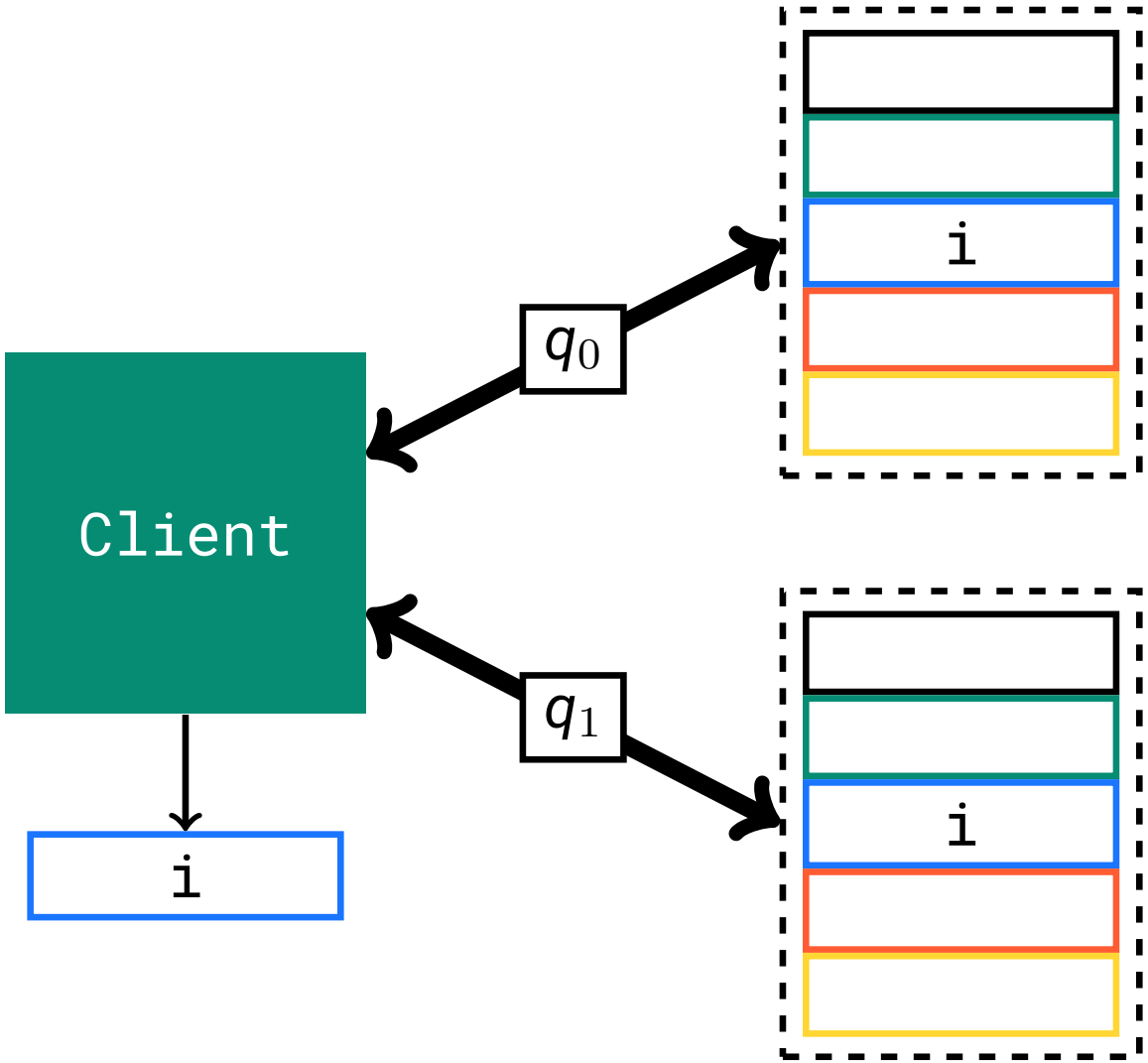
# FUNDAMENTALS

FUNCTIONALITY EXTENSIONS

PERFORMANCE OPTIMISATIONS

CONCLUSIONS

## [Cho+95]: Information-theoretic security



$$q_0, q_1 \leftarrow \mathsf{query}(1^\lambda, i)$$

$$s \leftarrow\!\!\$\ \{0,1\}$$

$$\{q_s : q_0, q_1 \leftarrow \mathsf{query}(1^\lambda, i)\}$$

$$\overset{c}{\simeq}$$

$$\{q_s : q_0, q_1 \leftarrow \mathsf{query}(1^\lambda, j)\}$$

$$q_0, q_1 \leftarrow \mathsf{query}(1^\lambda, i)$$

$$s \leftarrow_\$ \{0,1\}$$

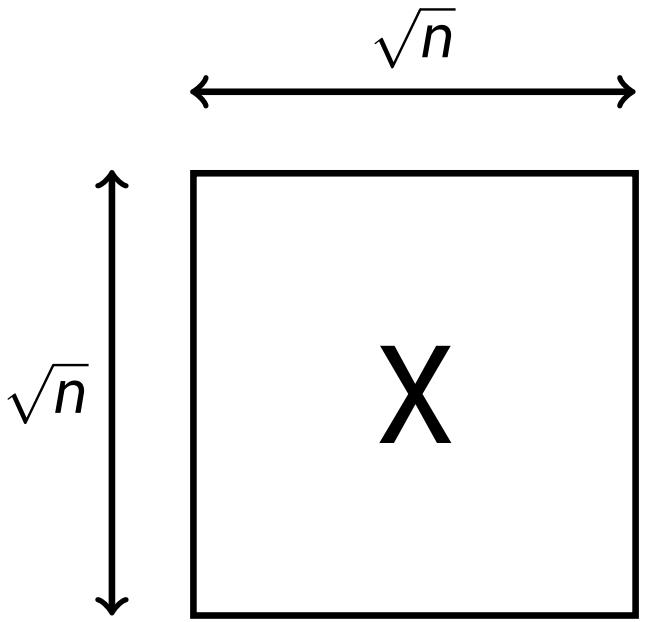
$$\{q_s : q_0, q_1 \leftarrow \mathsf{query}(1^\lambda, i)\}$$

$$\overset{c}{\simeq}$$

$$\{q_s : q_0, q_1 \leftarrow \mathsf{query}(1^\lambda, j)\}$$

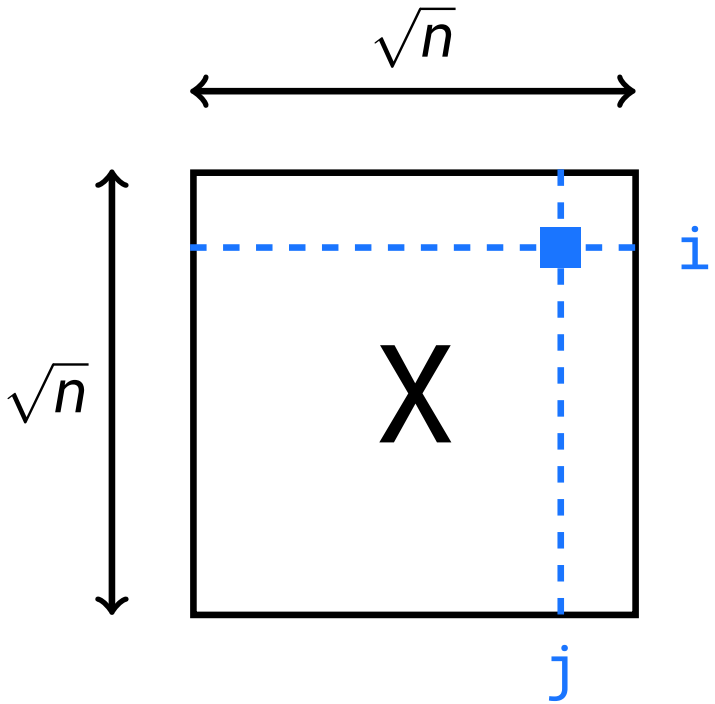
Relies on non-collusion of servers

View the database as a string: $DB \in \{0,1\}^n$

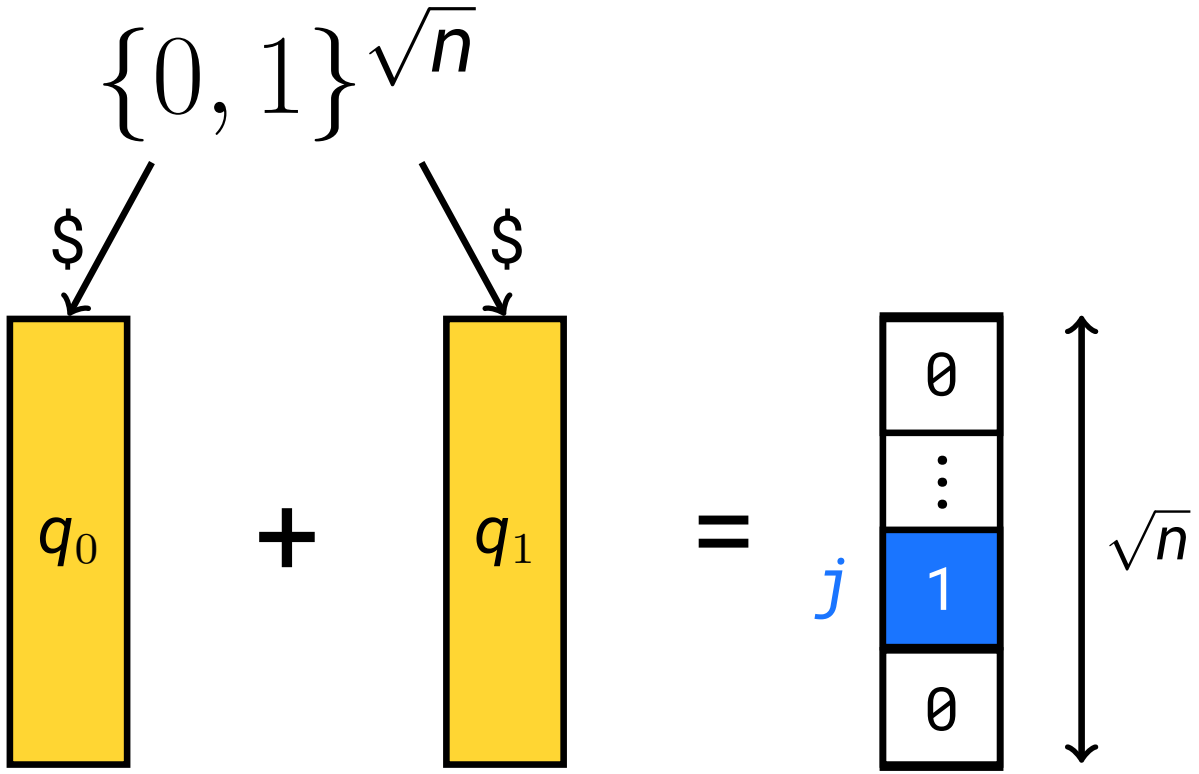



## BASIC CONSTRUCTION

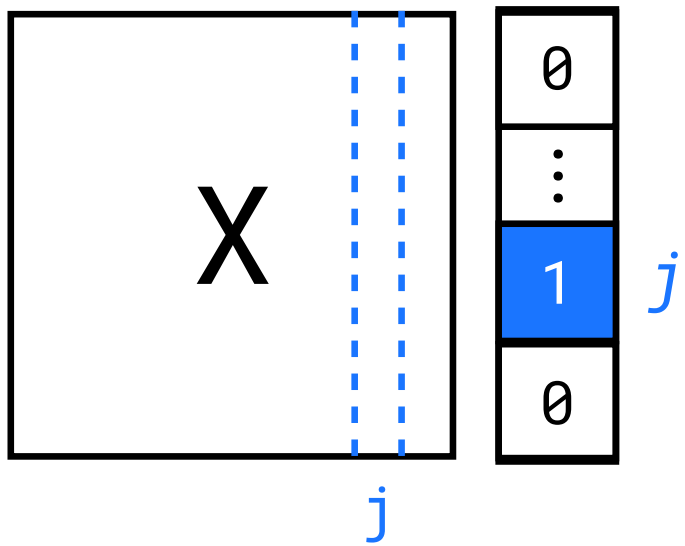
Client wants to read X[i][j]



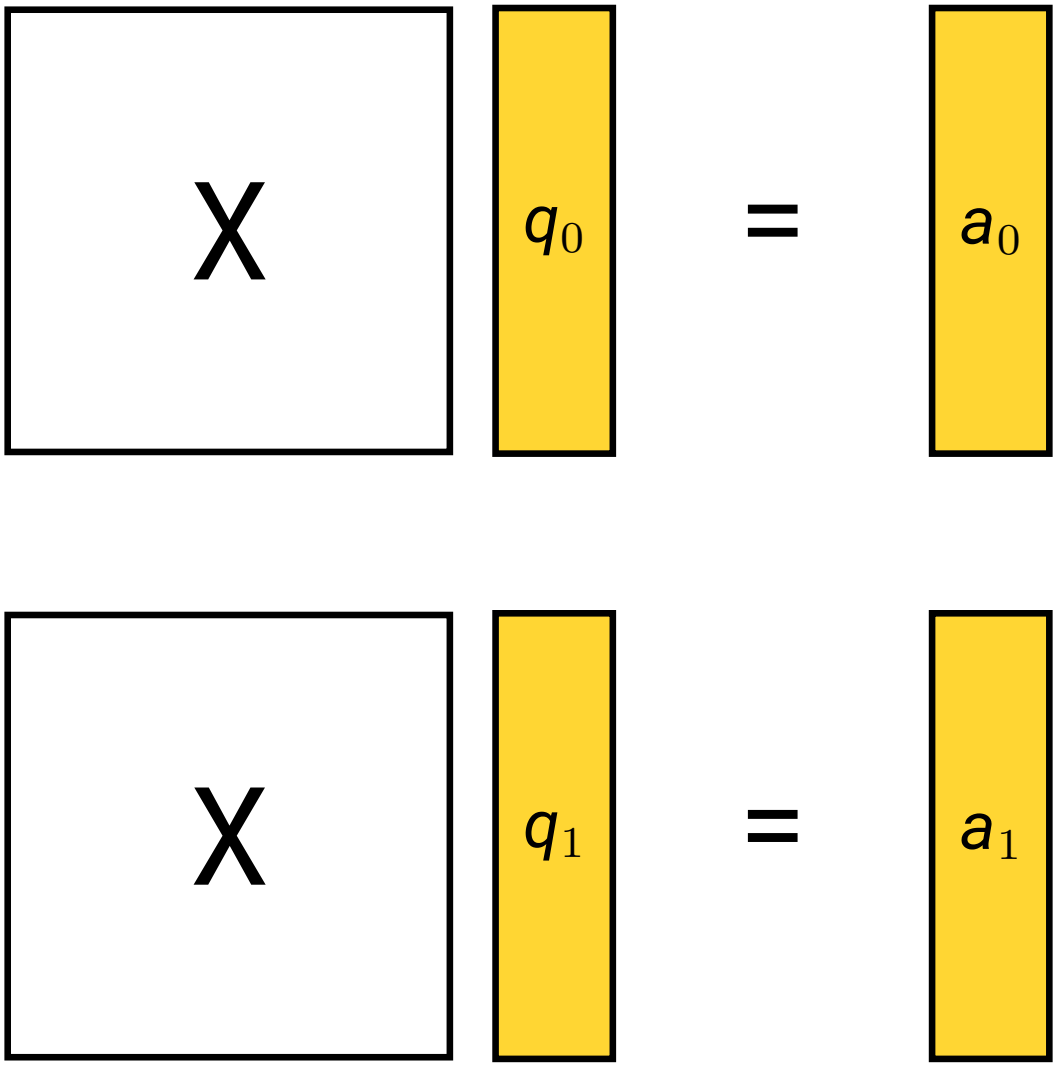

## BASIC CONSTRUCTION



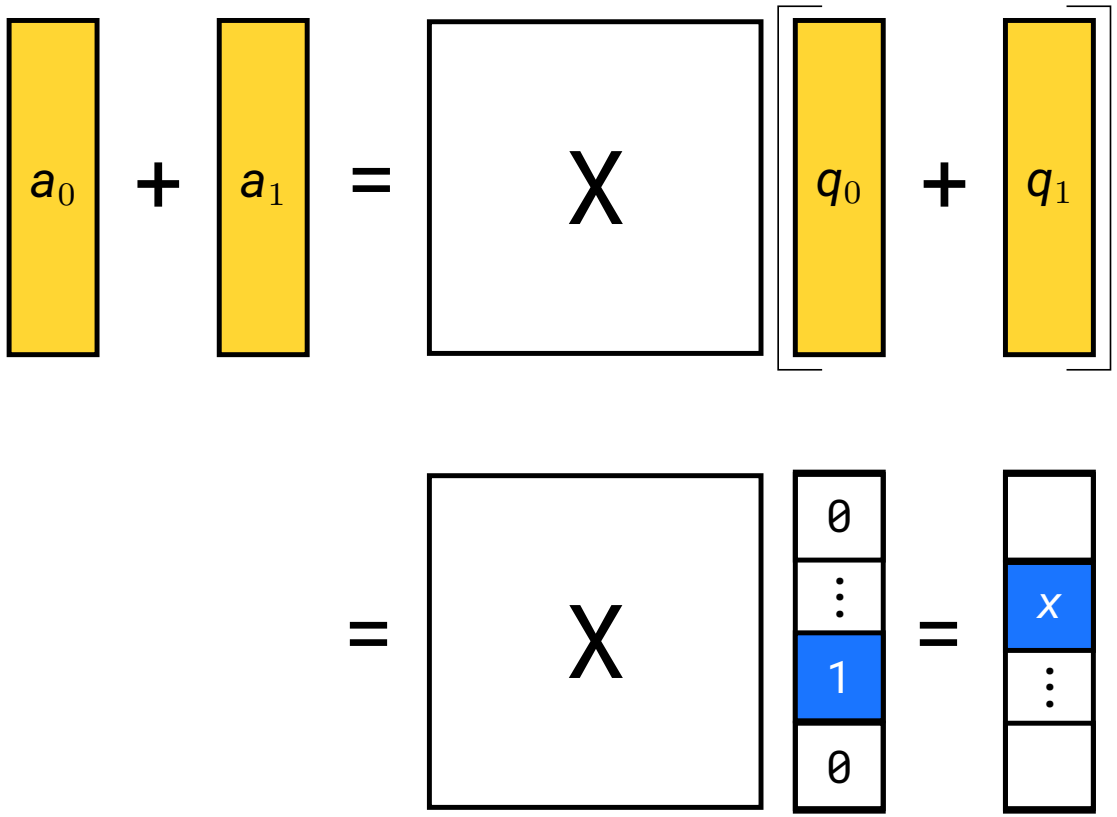

# BASIC CONSTRUCTION




# BASIC CONSTRUCTION




# BASIC CONSTRUCTION




# BASIC CONSTRUCTION

**Security:**

$q_0$ and $q_1$ are random vectors.

**Efficiency:**

$$|q_0| + |q_1| + |a_0| + |a_1| = 4\sqrt{n}$$

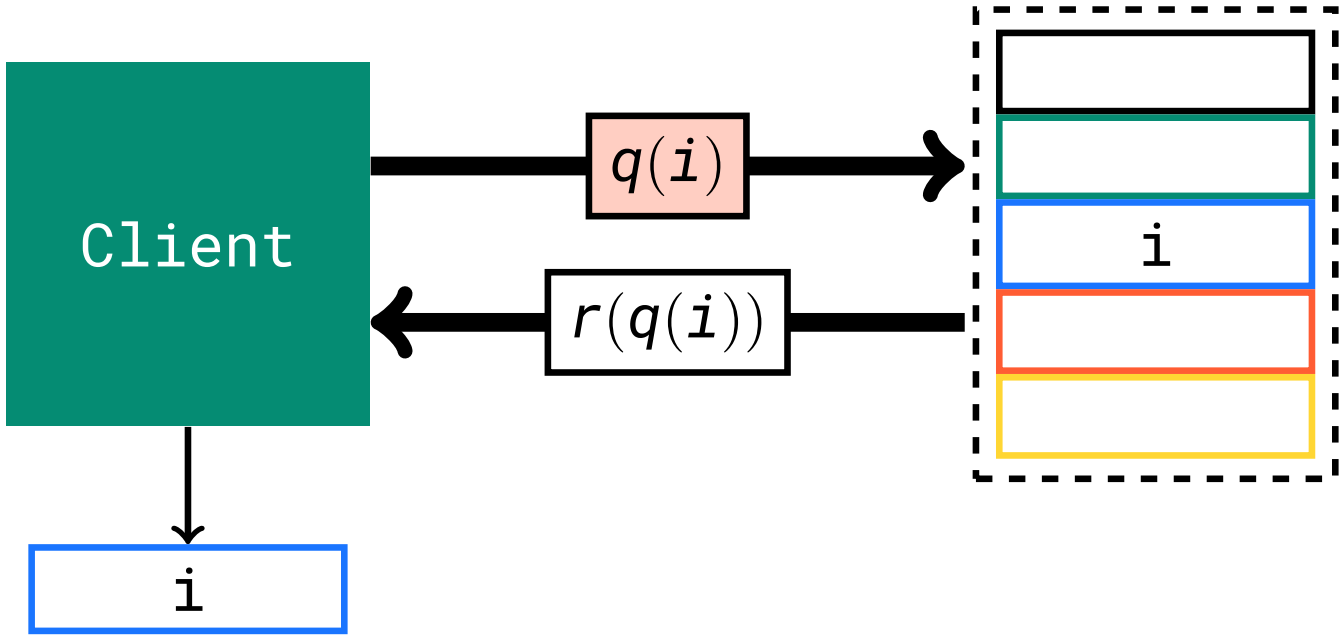
**Information-theoretic:**

$n^{o(1)}$ communication [DG16]

**Computational:**

$O(\log(n))$ communication [GI14; BGI15]
(based on distributed point functions)

[KO97]: Computational security



additively homomorphic encryption (AHE)

$$D(E(k,x) \oplus E(k,y)) = D(E(k,x+y)) = x+y$$

[Gen09] fully homomorphic encryption (FHE)

$$D(E(k,x) \otimes E(k,y)) = D(E(k,x \cdot y)) = x \cdot y$$

$$\textit{AHE} \leftarrow \textit{groups/lattices}; \; \textit{FHE} \leftarrow \textit{lattices}$$

[KO97]

Encryption prevents server learning query index

**client**: iv = E(0) E(0) E(1) E(0) E(0)

**server**: sv = X[0] X[1] X[2] X[3] X[4]

**response**: $\langle iv, sv \rangle$ = $\sum_{i=0} E(i \cdot X[i]) = E(X[2])$

**Problem:**

- $2\sqrt{n}$ communication (for $\sqrt{n} \times \sqrt{n}$ DB)
- $n$ computation

[SC07]: Computational PIR (from groups) is too slow and expensive.

- Faster to send **whole** database over KBps connection

AHE from Ring LWE (fully HE) is *cheaper*

Database with $n = 2^{20}$, and 3KB byte records (OnionPIR [MCR21]):

- 192KB communication
- 400 seconds computation
- Can achieve $\sqrt{n}$ efficiency [CHK22]

*Performance optimisations covered later*

# What do you remember?

¡2¡¡QUIZ!!2!

Multi-server PIR is:

- only computationally secure
- only statistically secure
- perfectly secure

**QUESTION #1**

Multi-server PIR is:

- only computationally secure
- only statistically secure
- perfectly secure

ANSWER #1

Fully homomorphic encryption is **necessary** for building single-server PIR.

- True
- False

## QUESTION #2

Fully homomorphic encryption is **necessary** for building single-server PIR.

- True
- False

ANSWER #2

FUNDAMENTALS

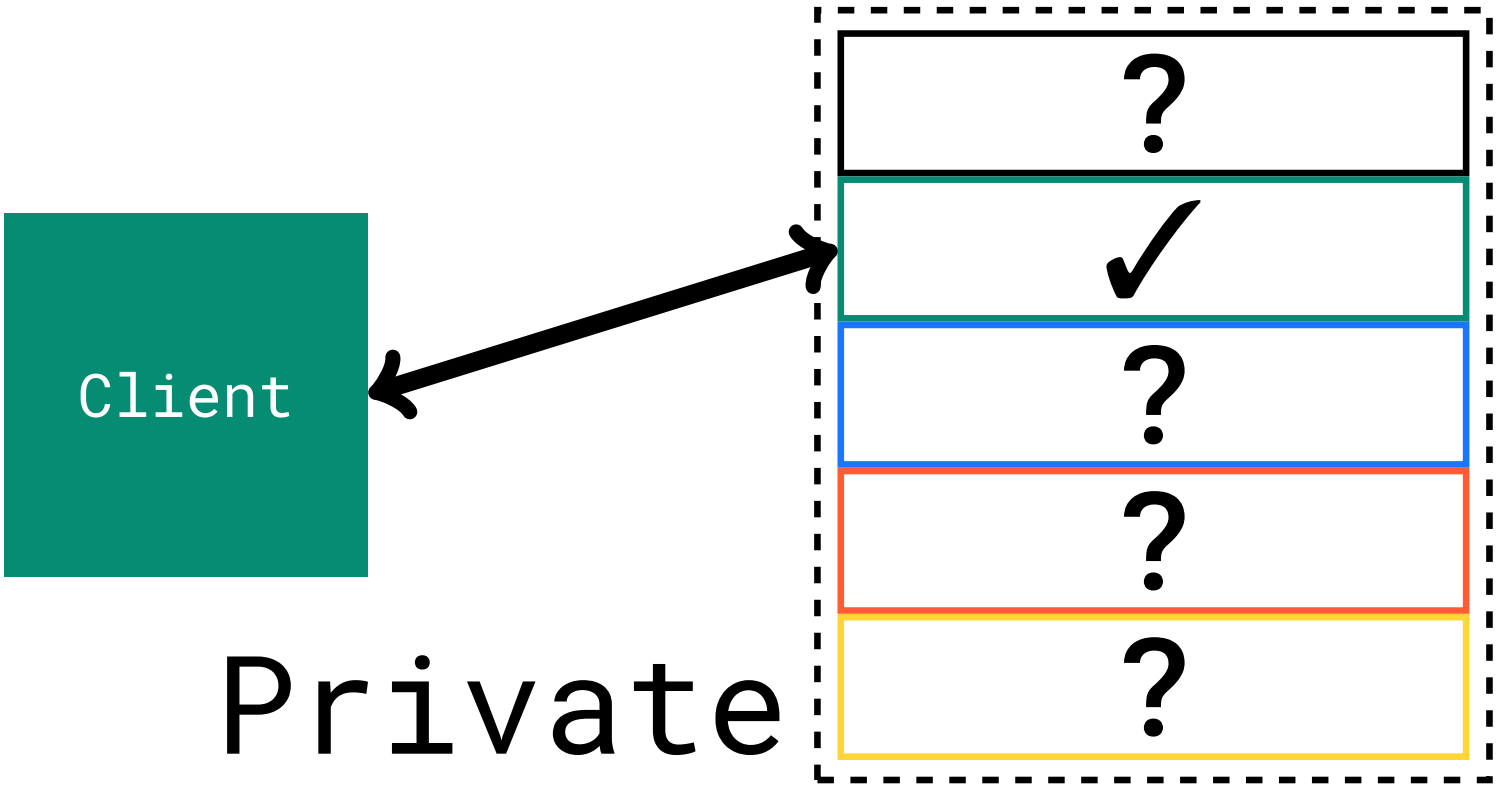
**FUNCTIONALITY EXTENSIONS**

- PROVIDING DATABASE PRIVACY
- KEYWORD QUERIES

PERFORMANCE OPTIMISATIONS

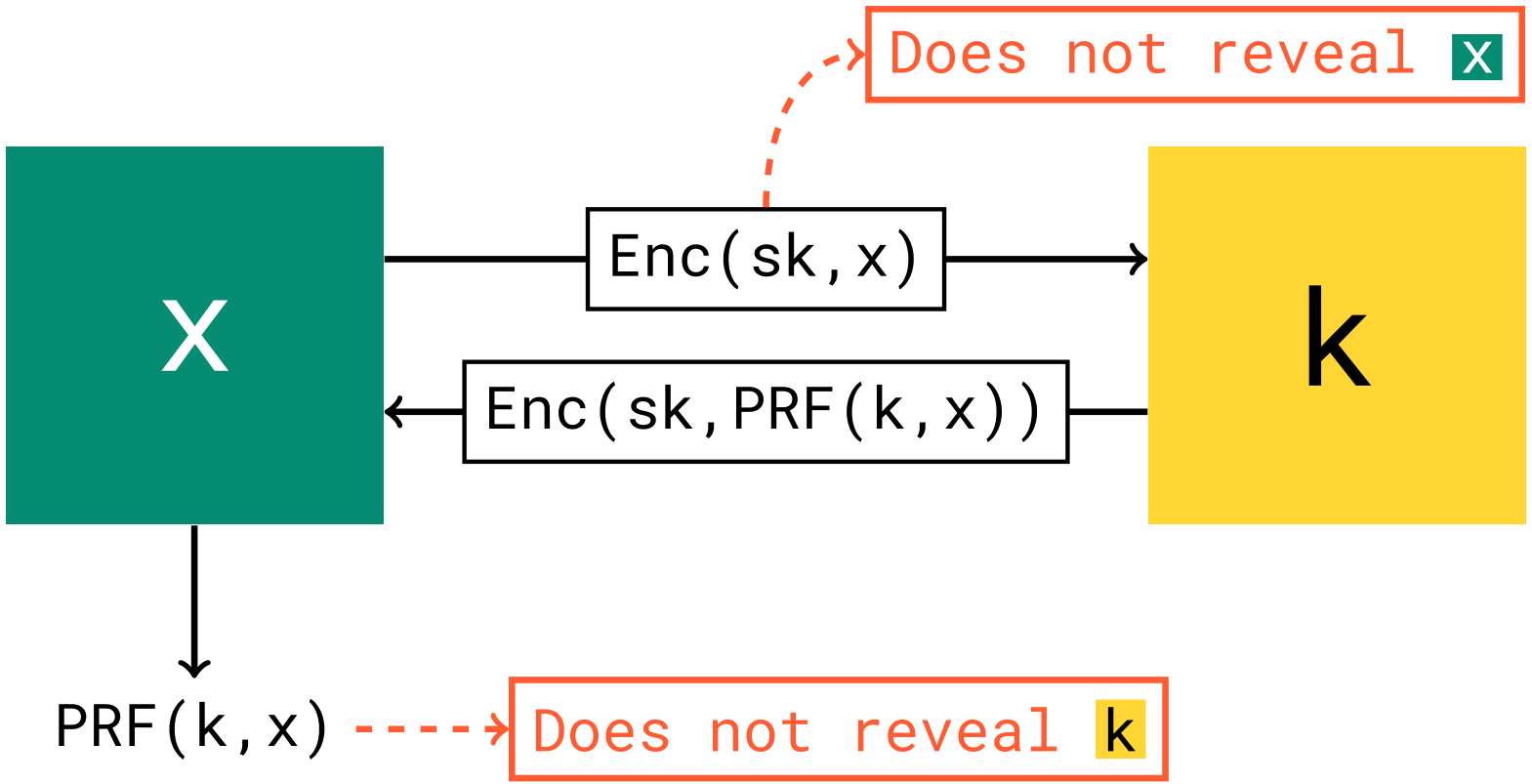
CONCLUSIONS

## Symmetric PIR



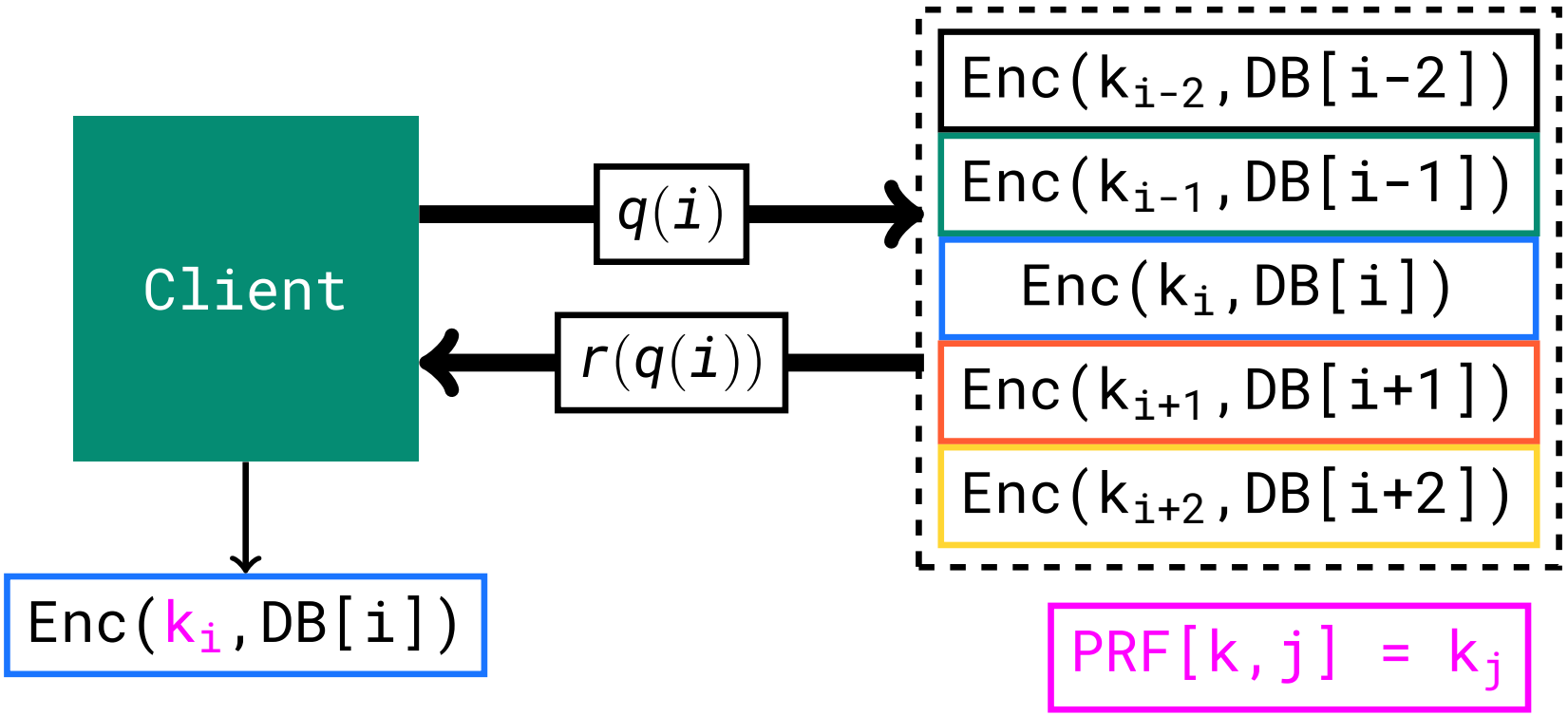

- *Learn nothing about non-queried elements*

# Oblivious Pseudorandom Function protocol

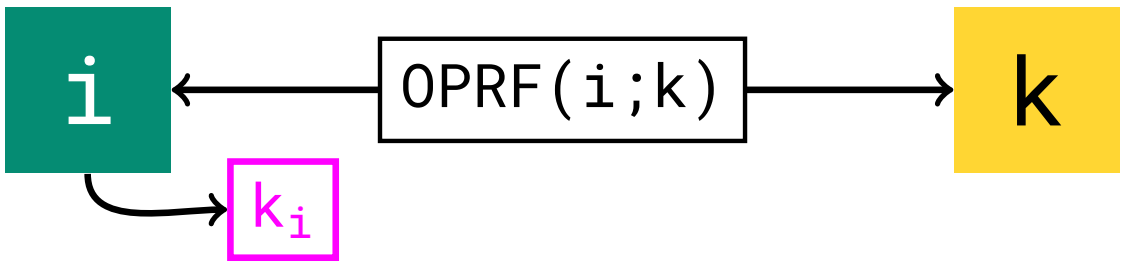



*Can build very efficiently from elliptic curves*

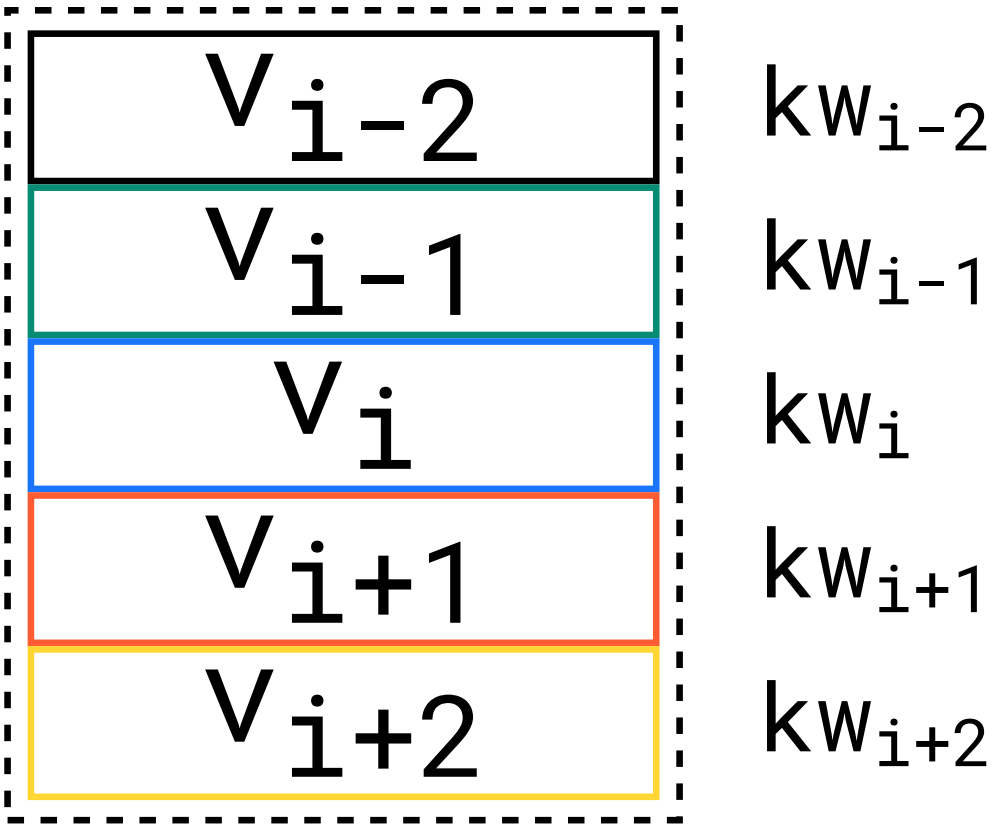
## Phase 01



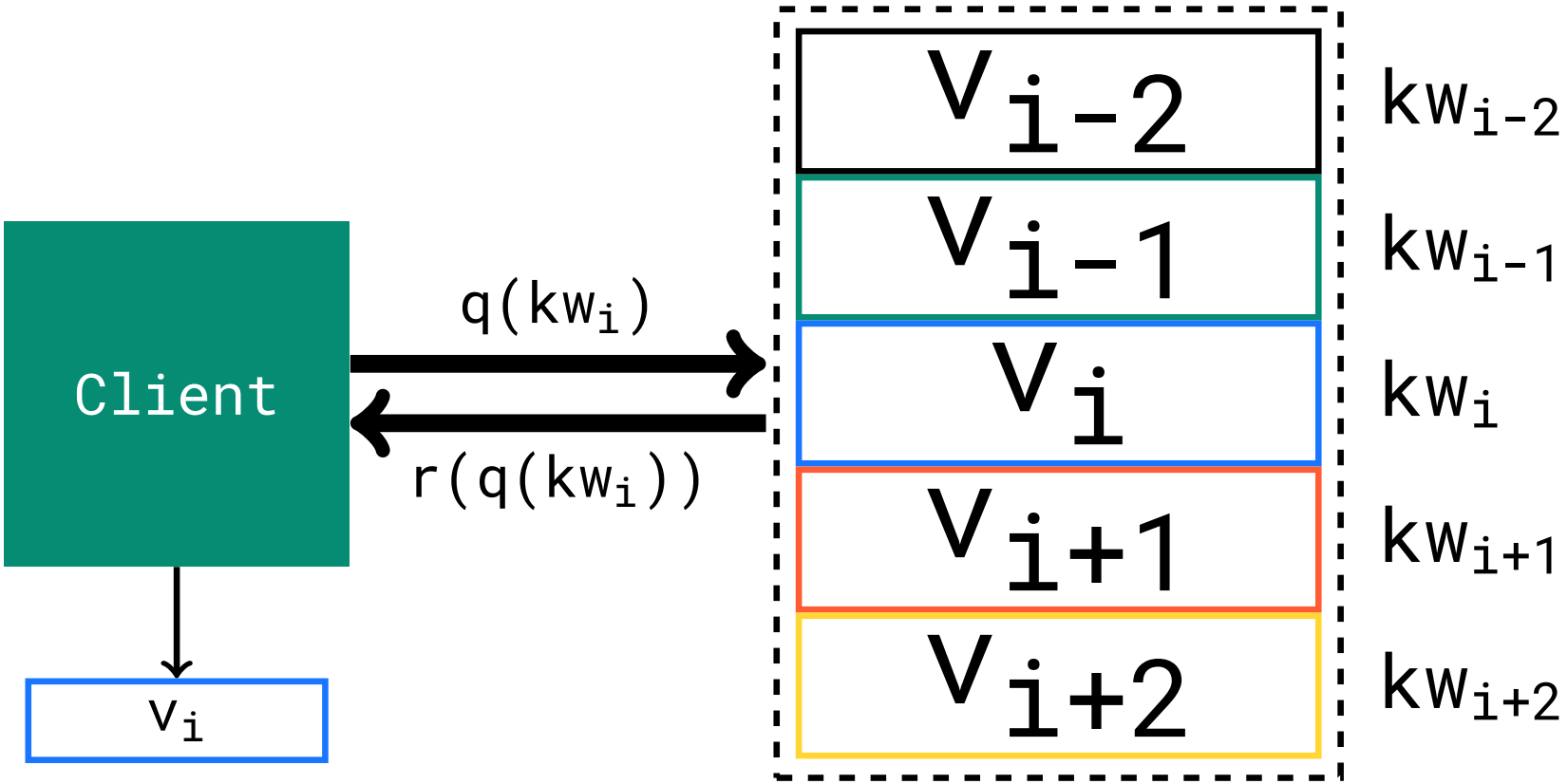

## Phase 02



Client




Closer to real-world database abstractions




Note: allows for sparser DB representations

## KEYWORD PIR

[CGN98]

- Write key-value map as set of pairs $(kw_i, kw_i || v_i)$
- Sort into database by keywords
- Apply binary search with index-based PIR to locate $(kw_i || v_i)$
- Requires $O(\log(n))$ PIR queries
- Single- or Multi-server compatible

## KEYWORD PIR FROM INDEX-BASED PIR

Point function: $f_{a,b} : \mathcal{X} \mapsto \mathcal{Y}$, $a \in \mathcal{X}$, $b \in \mathcal{Y}$

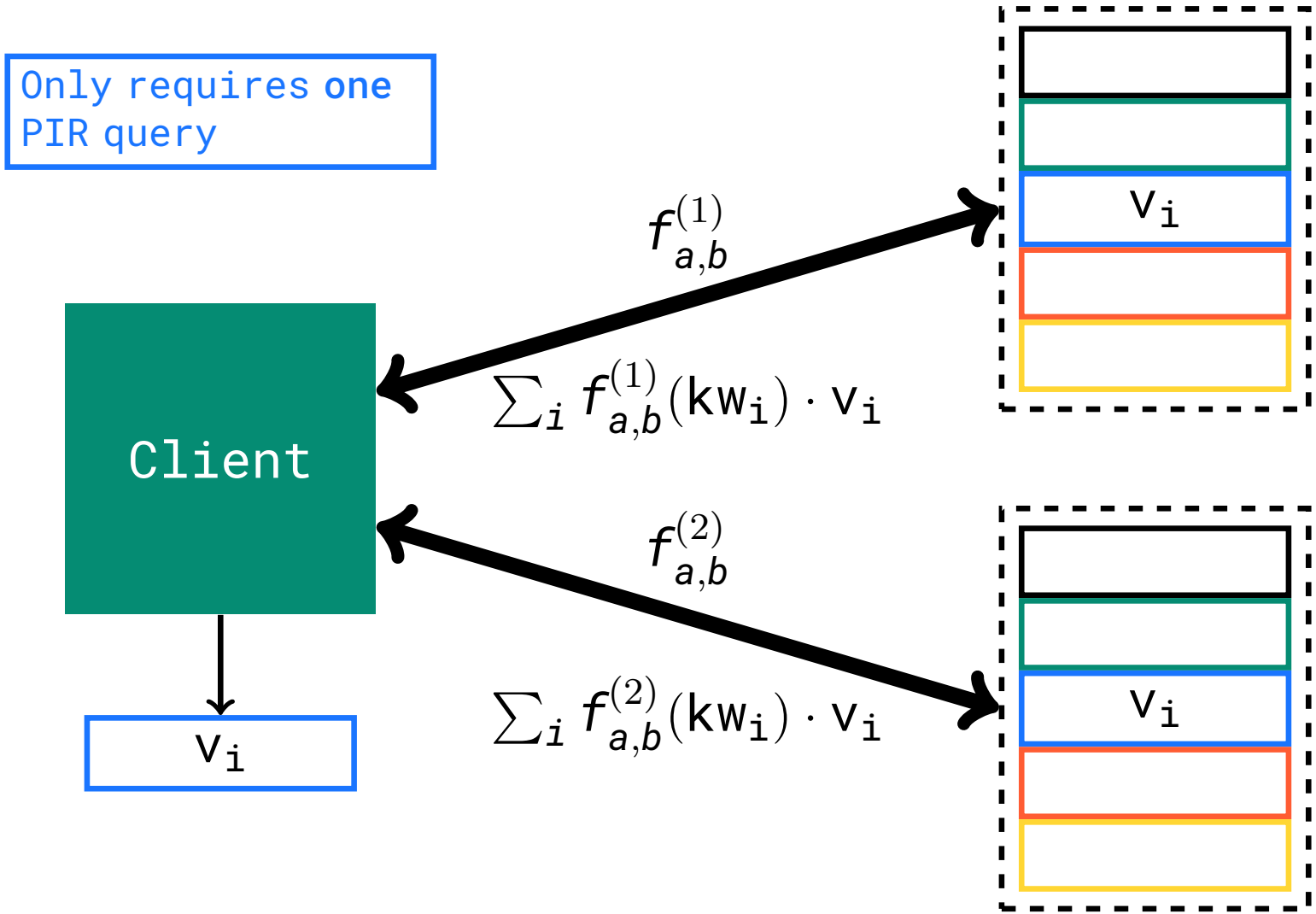
- $f_{a,b}(x) = 0$ for all $x \neq a$
- $f_{a,b}(x) = b$ for all $x \neq b$

Distributed point function [GI14]:

- Create function shares $f^{(1)}_{a,b}, f^{(2)}_{a,b} : \mathcal{X} \mapsto \mathcal{Y}$
- Shares satisfy $f^{(1)}_{a,b}(a) + f^{(2)}_{a,b}(a) = b$

## DISTRIBUTED POINT FUNCTIONS: DPFS

Client sets $a = \mathsf{kw_i}$, $b = 1$



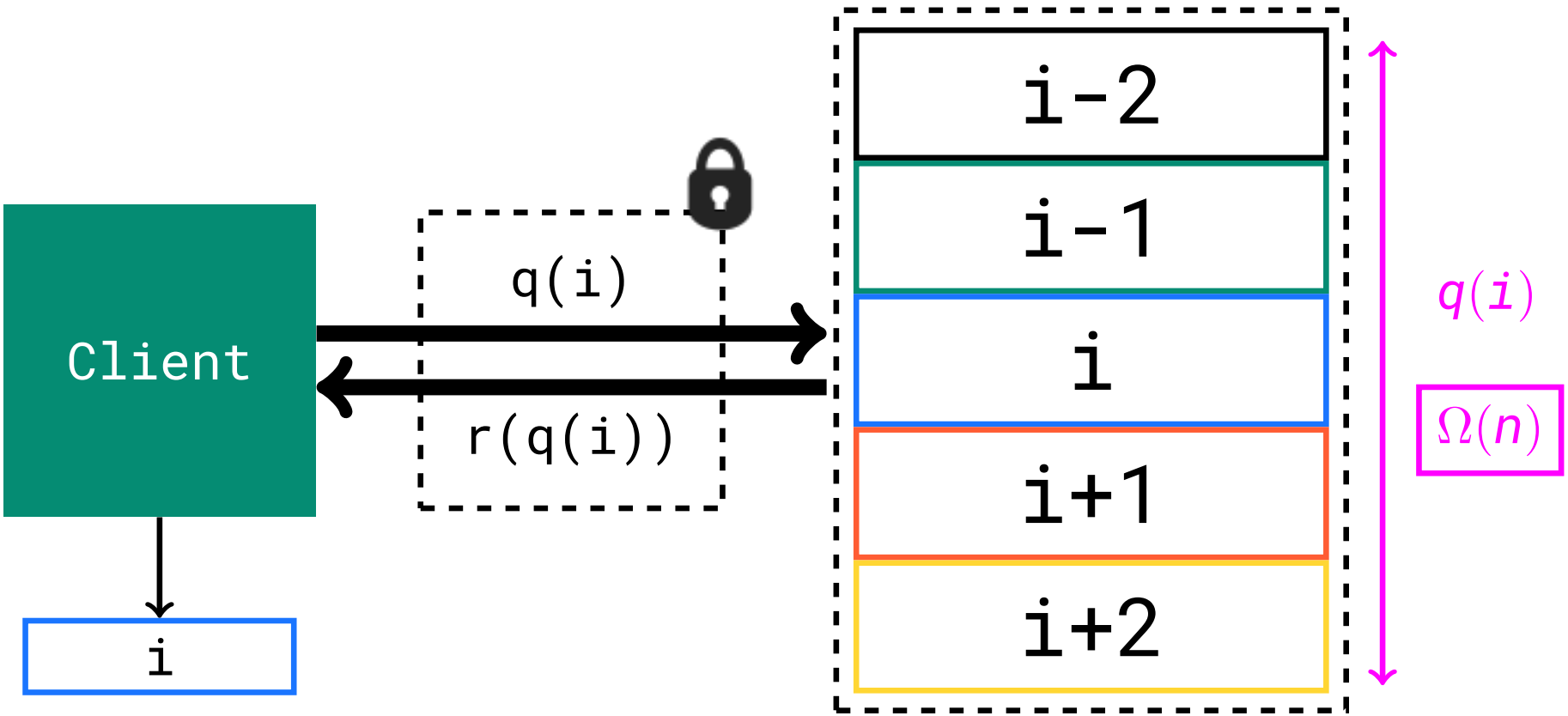
FUNDAMENTALS

FUNCTIONALITY EXTENSIONS

**PERFORMANCE OPTIMISATIONS**

CONCLUSIONS

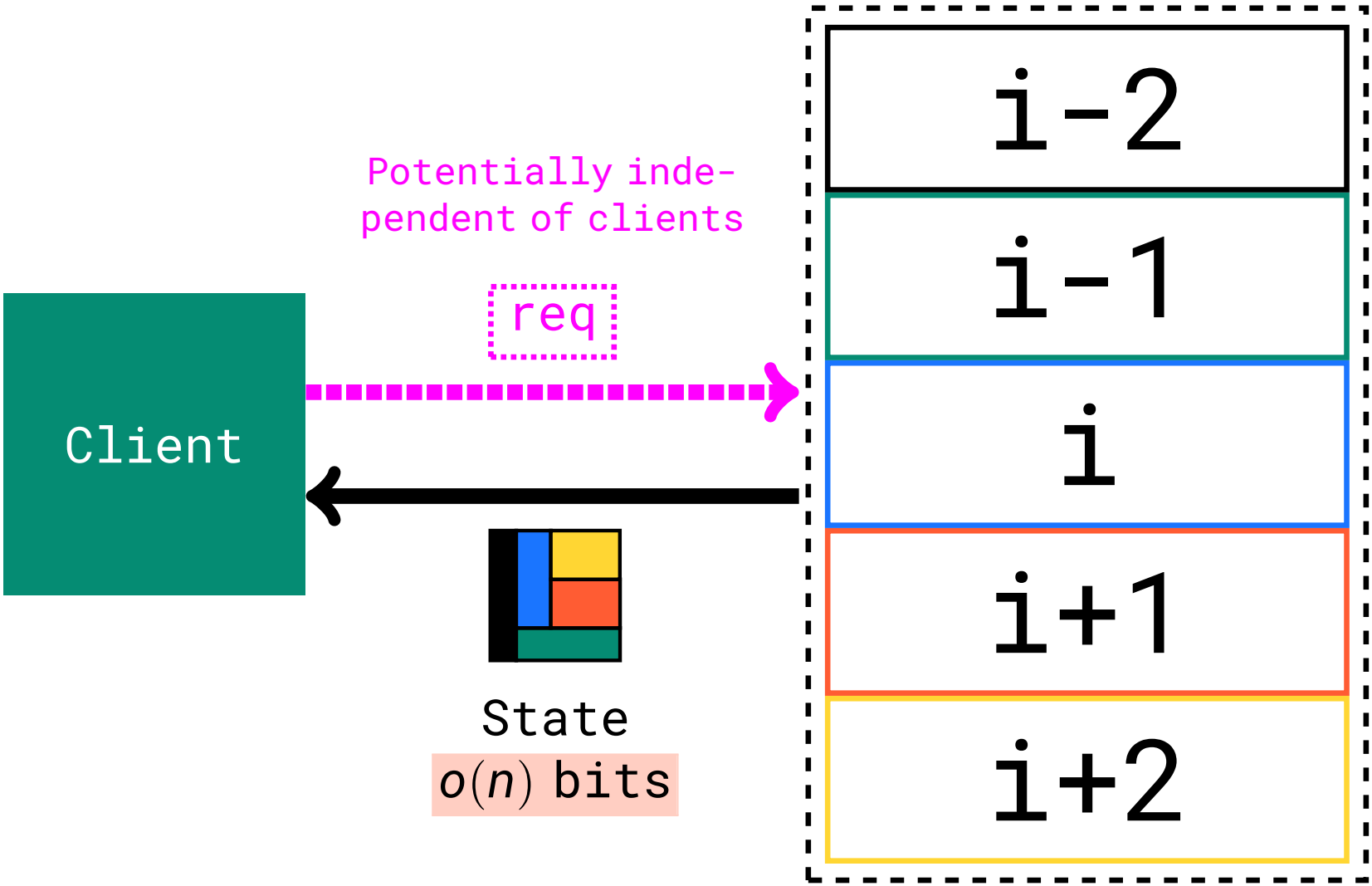
To hide query, server scans database linearly



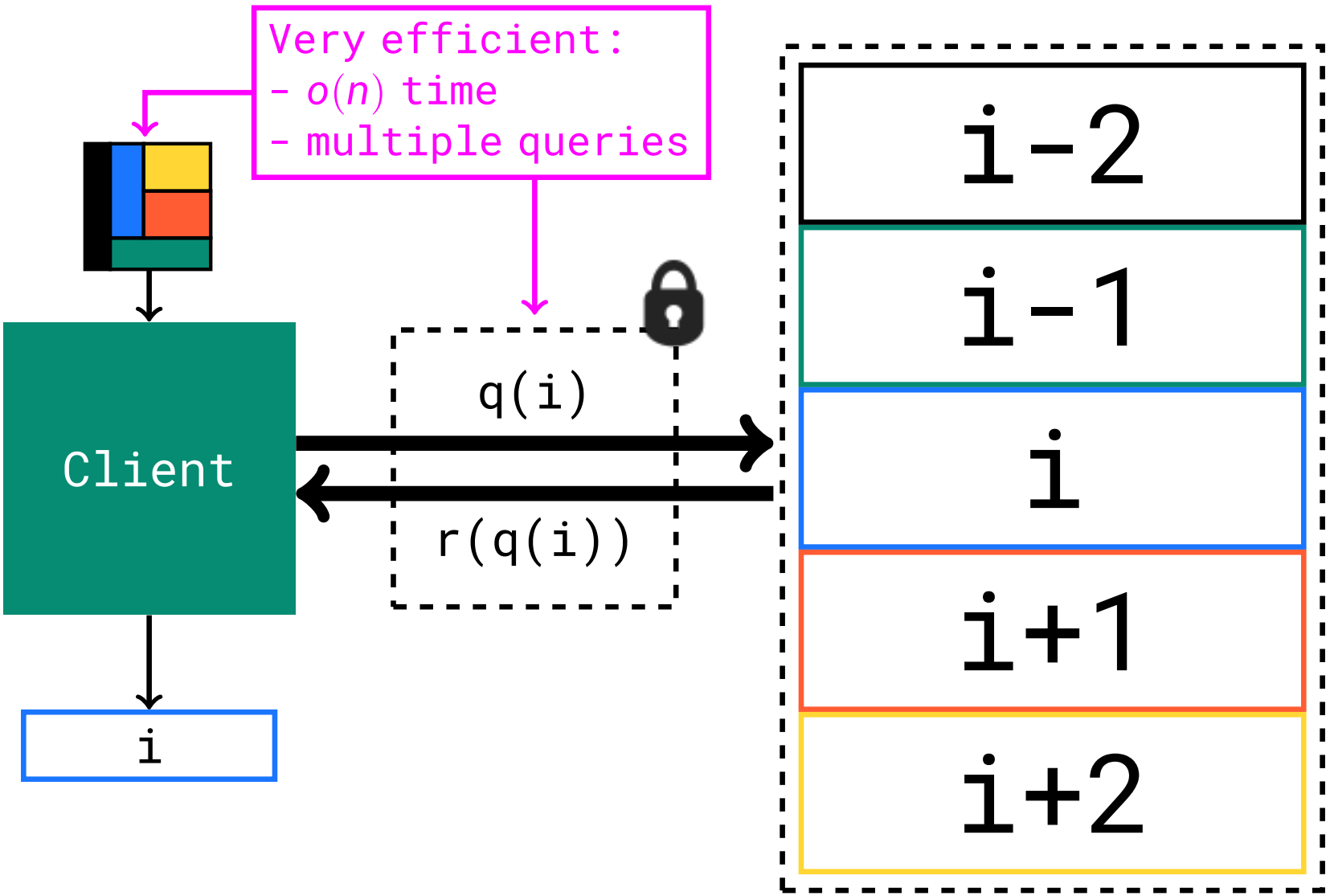

Irrespective of assumptions & number of servers

## COMPUTATIONAL COSTS

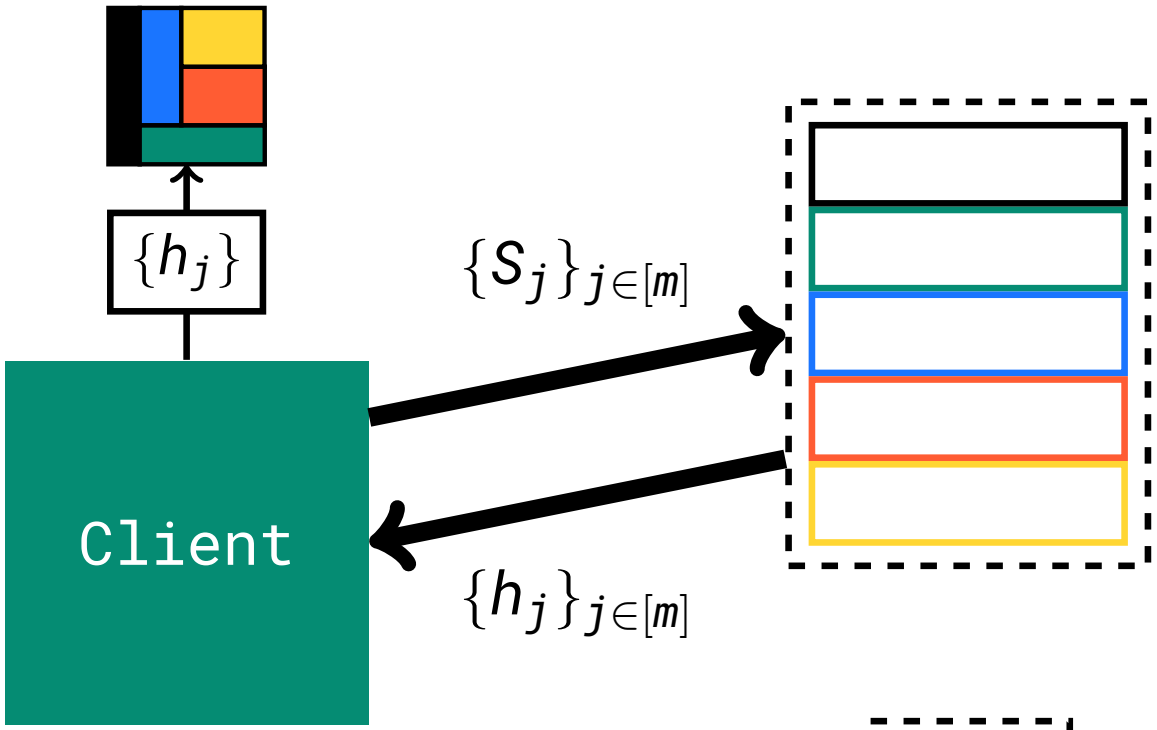
# Offline phase



# Online phase



[CK20; KC21; CHK22]

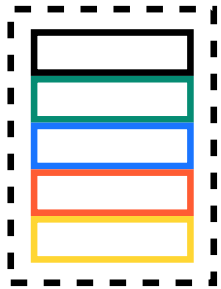



Server One

$h_j = \sum_{\iota\in S_j} DB[\iota]$

(mod 2)

$|S_j| = \sqrt{n}$



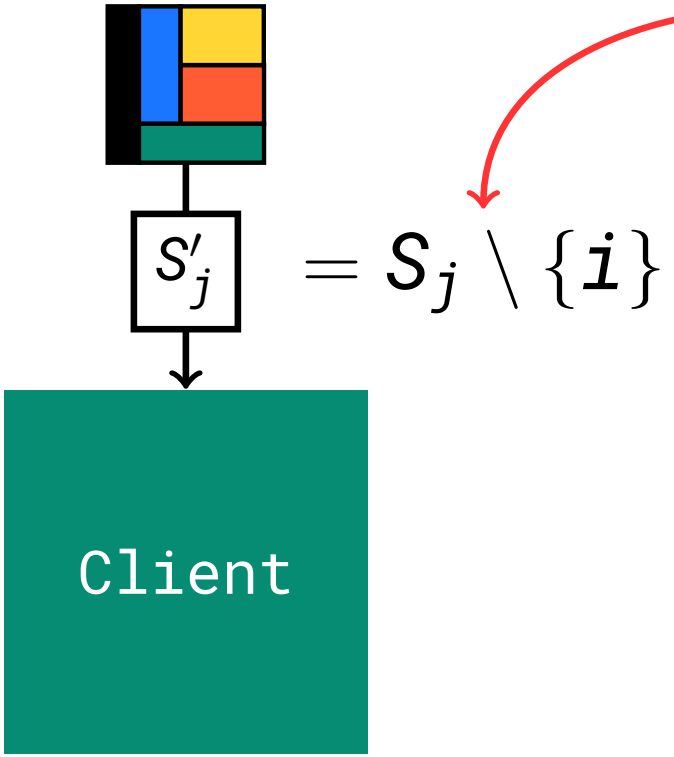
Server Two

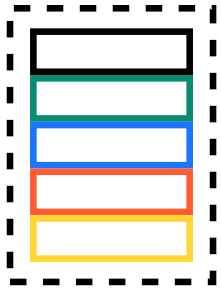
Online query unknown

## MULTI-SERVER: OFFLINE

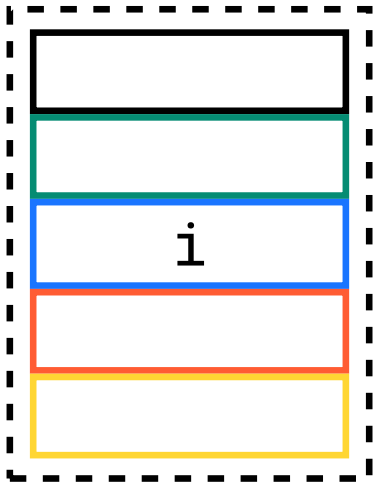
[CK20; KC21; CHK22]

If cannot find $j$ such that $i \in S_j$, protocol fails



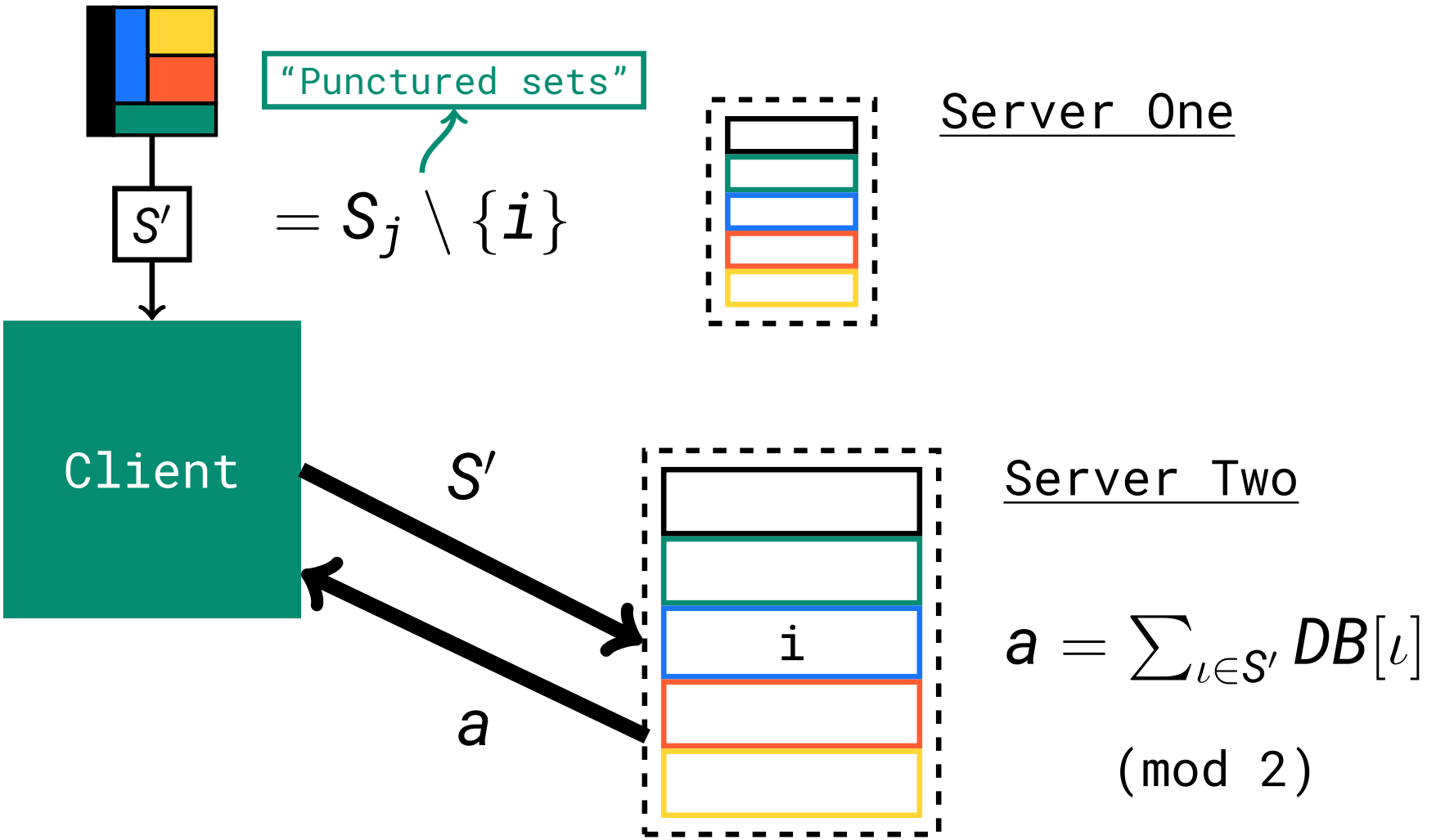
Server One




Server Two

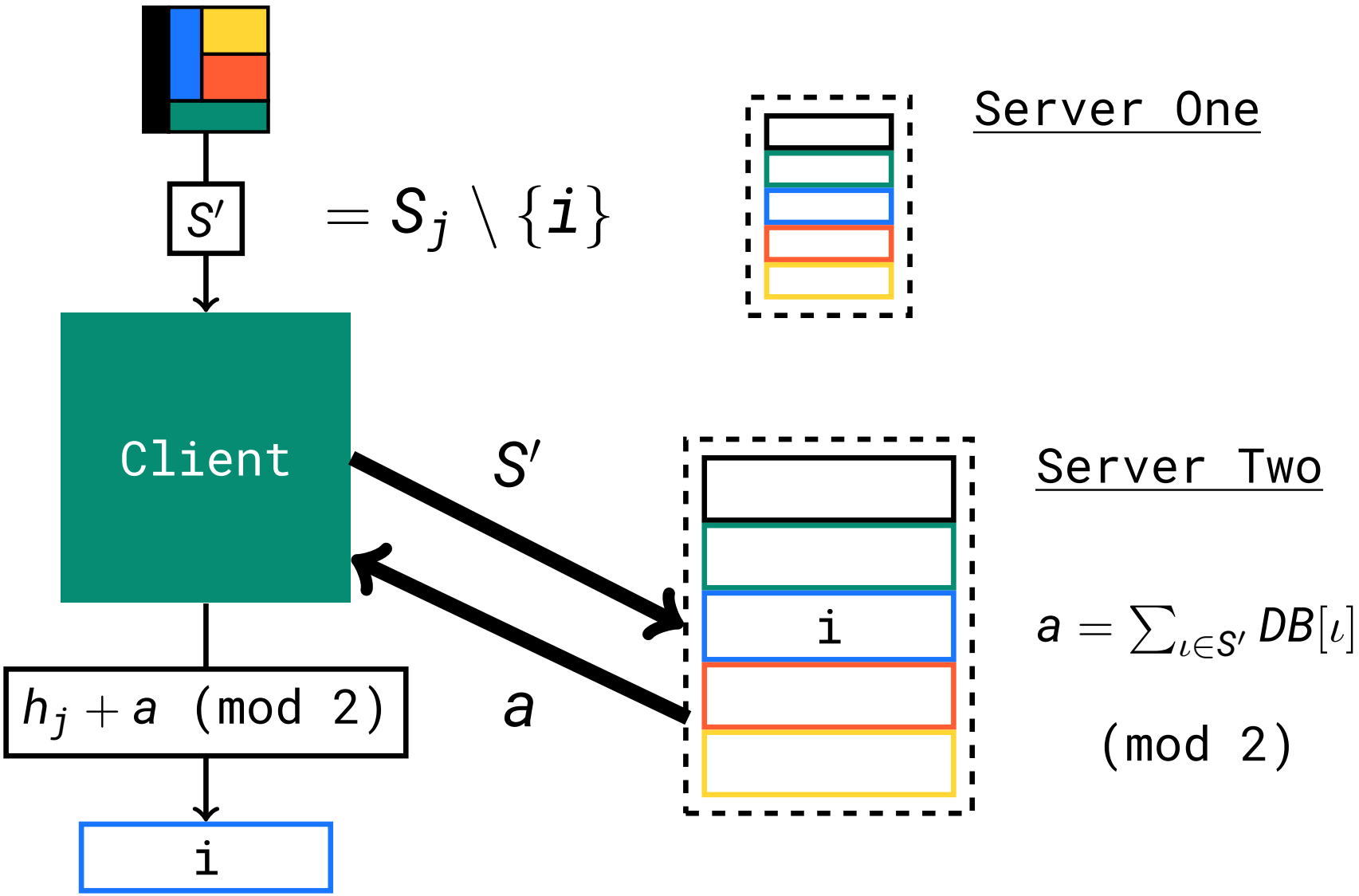
## MULTI-SERVER: ONLINE QUERY FOR $i$

[CK20; KC21; CHK22]



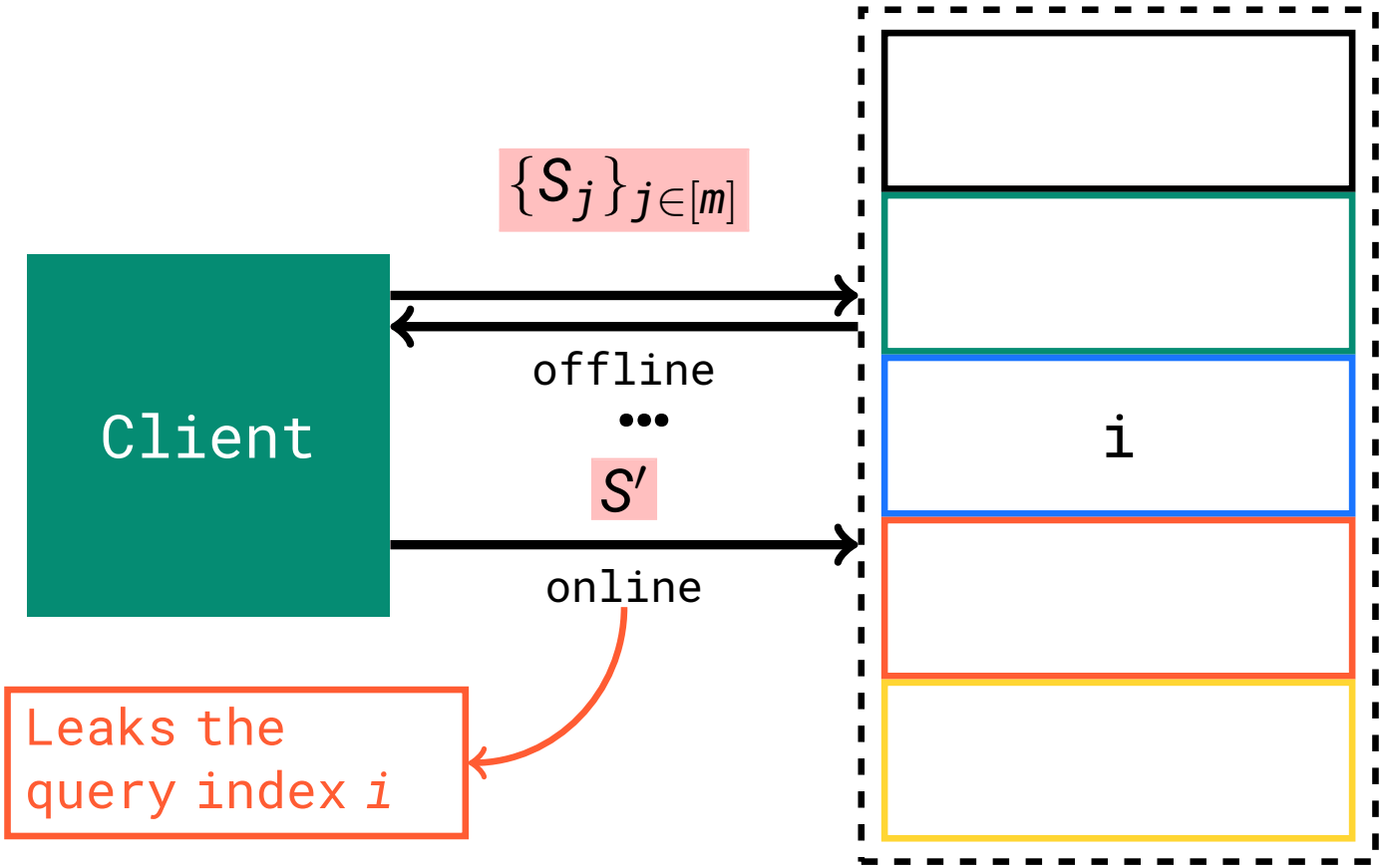

## MULTI-SERVER: ONLINE QUERY FOR $i$

[CK20; KC21; CHK22]




# MULTI-SERVER: ONLINE QUERY FOR $i$

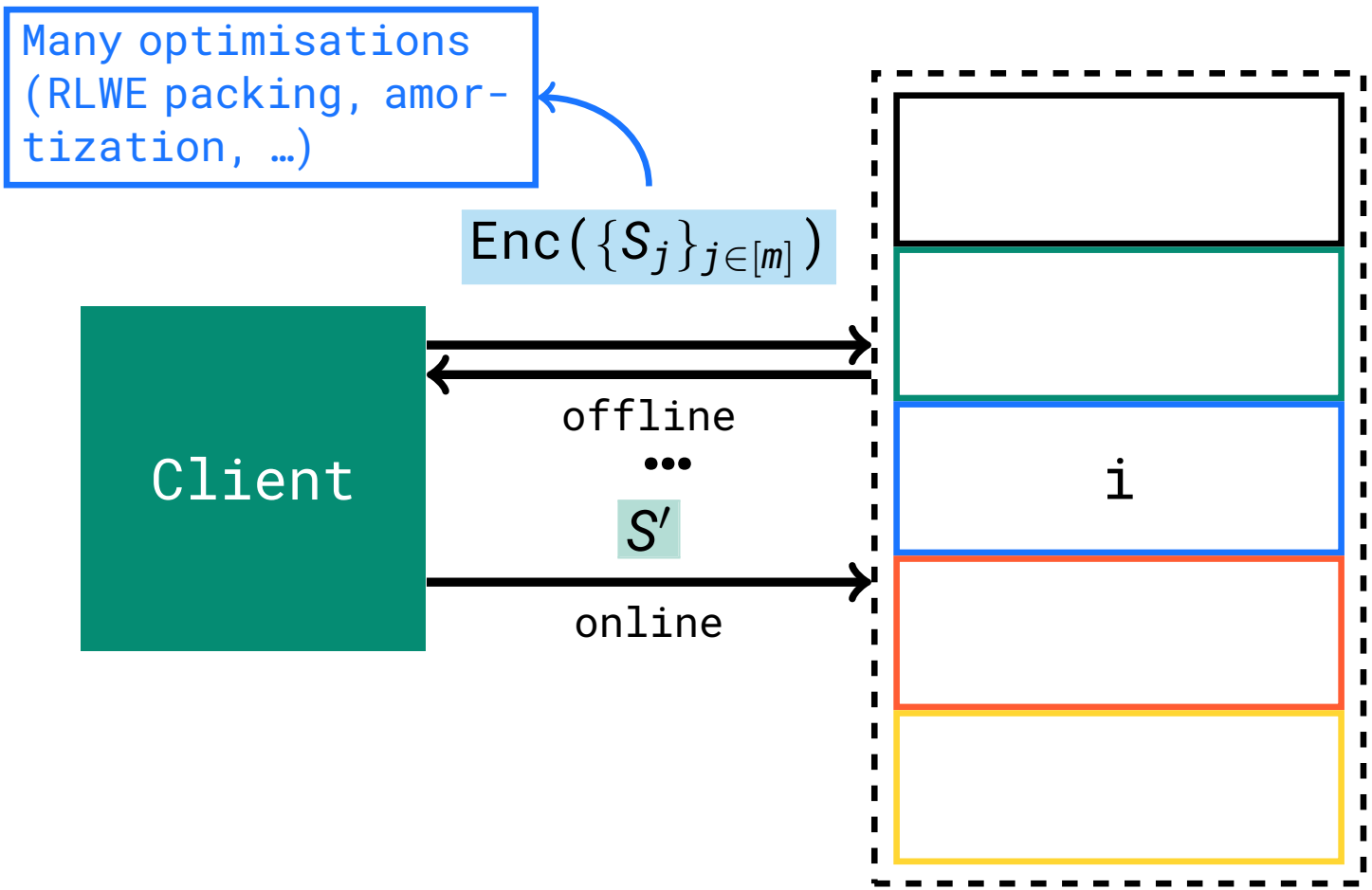
[Ang+18; PPY18; MCR21; Zho+23]




Clearly, the trivial solution is insecure

## SINGLE-SERVER CONSTRUCTION

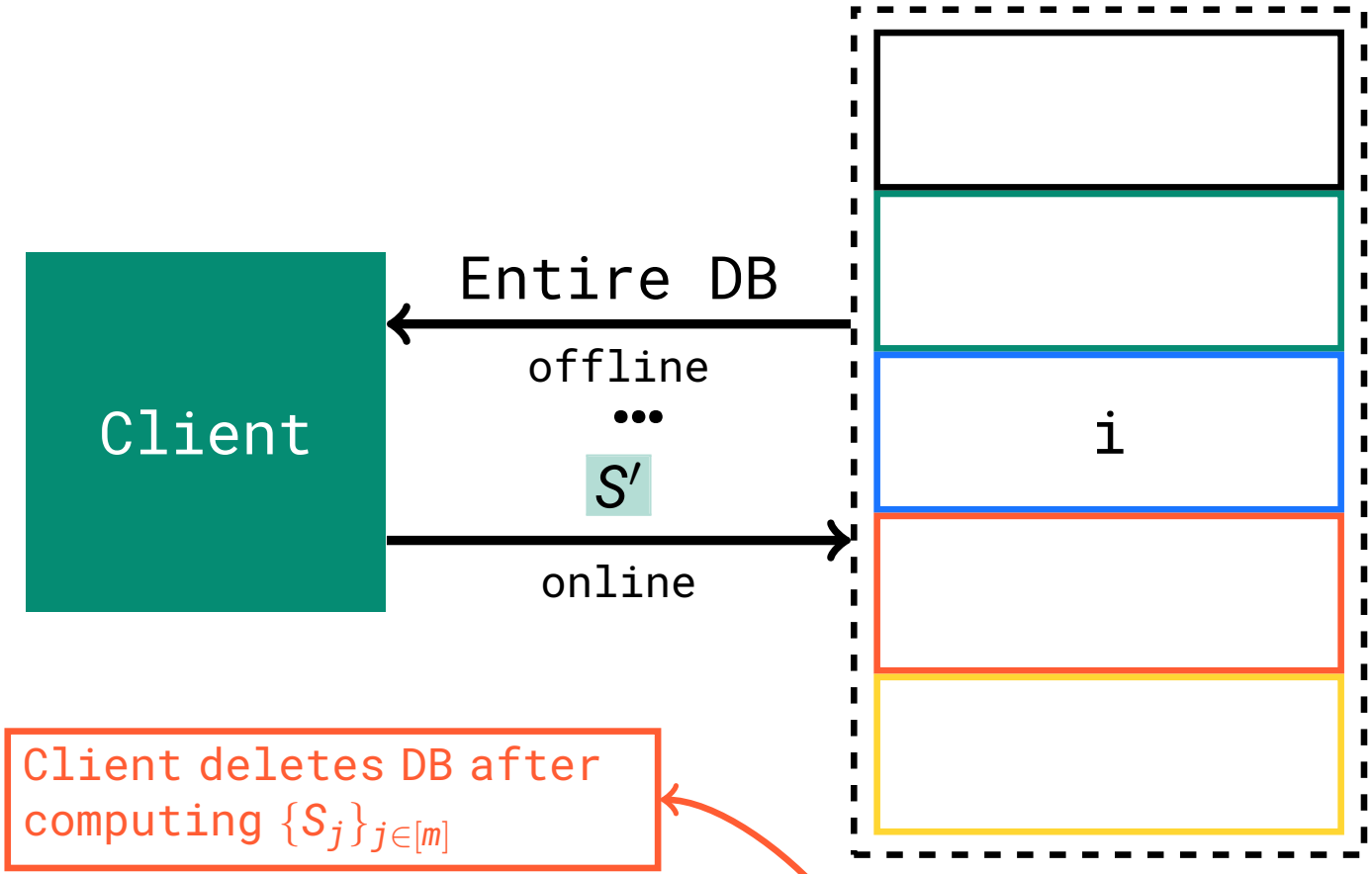
[Ang+18; MCR21]




**Answer #1:** Perform offline phase in FHE

## SINGLE-SERVER FHE

[Ang+18; MCR21]




**Answer #2:** Stream entire DB to client, and let them choose $\{S_j\}_{j\in[m]}$ locally.

**Multi-server constructions**

- $\sqrt{n}$ communication & computation (amortized)
- assumes $\sqrt{n}$ client queries

**Single-server constructions**

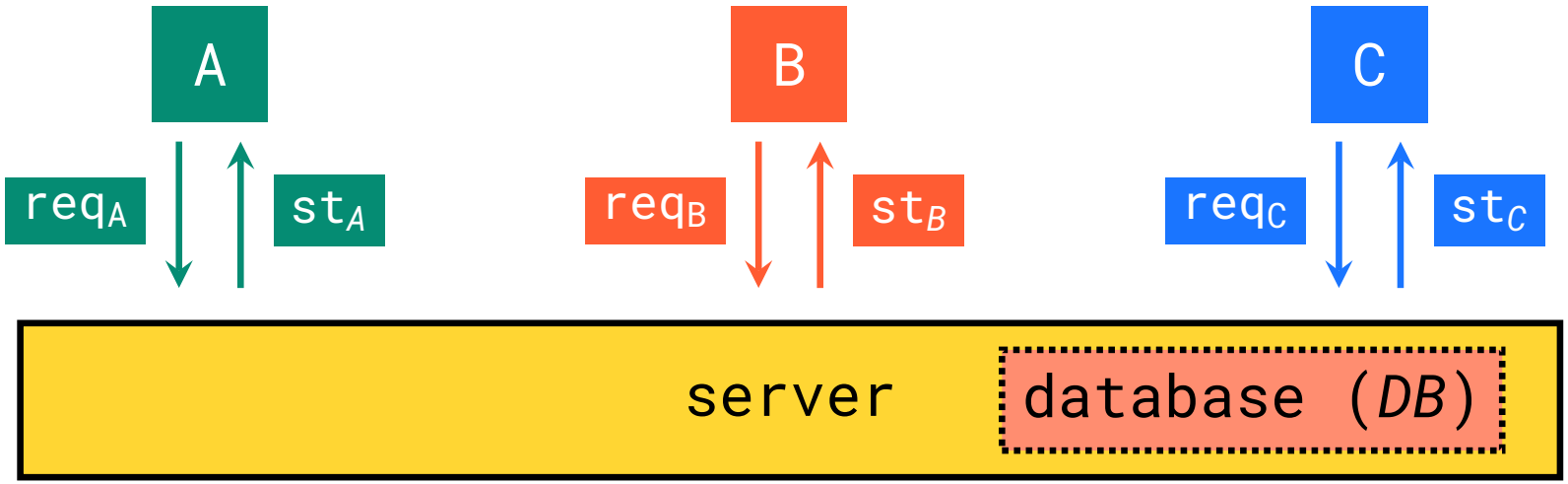
- Streaming: communication $> |DB|$
- FHE: $\sqrt{n}$ communication & computation

**Lower-bound**: For offline/online schemes storing DB in original form: $C \cdot T > n$ must hold

C=Communication
T=Online Time

## THEORETICAL EFFICIENCY

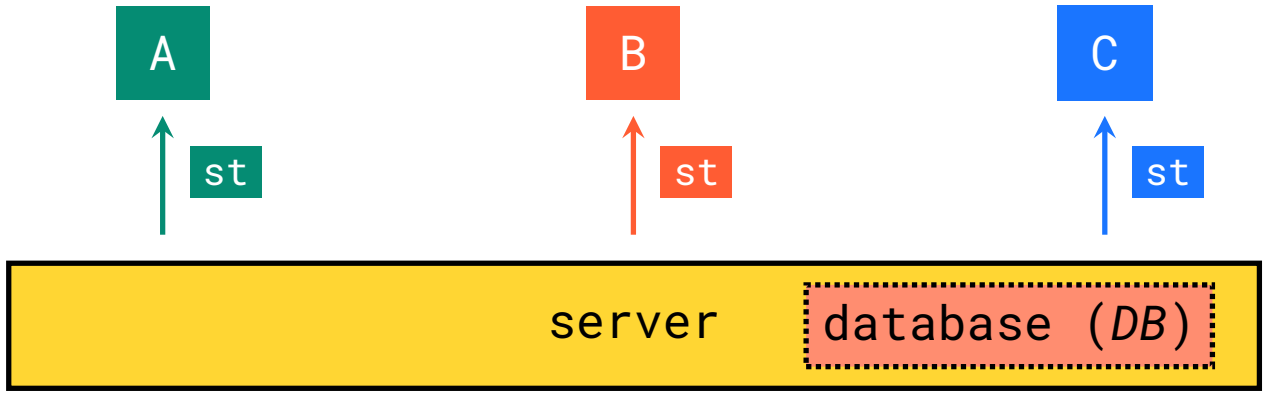
# Problem: Single-Server approaches




- Heavy offline-phase via FHE
- *Cannot* be amortised across clients

**CLIENT-DEPENDENT PREPROCESSING**

# Goals




- Amortise server offline computation
- Remove dependency on optimised FHE and RLWE
- Configurable and efficient

**Possible?** Yes!

*Opinion*: Libraries remain highly experimental

**Two schemes:**

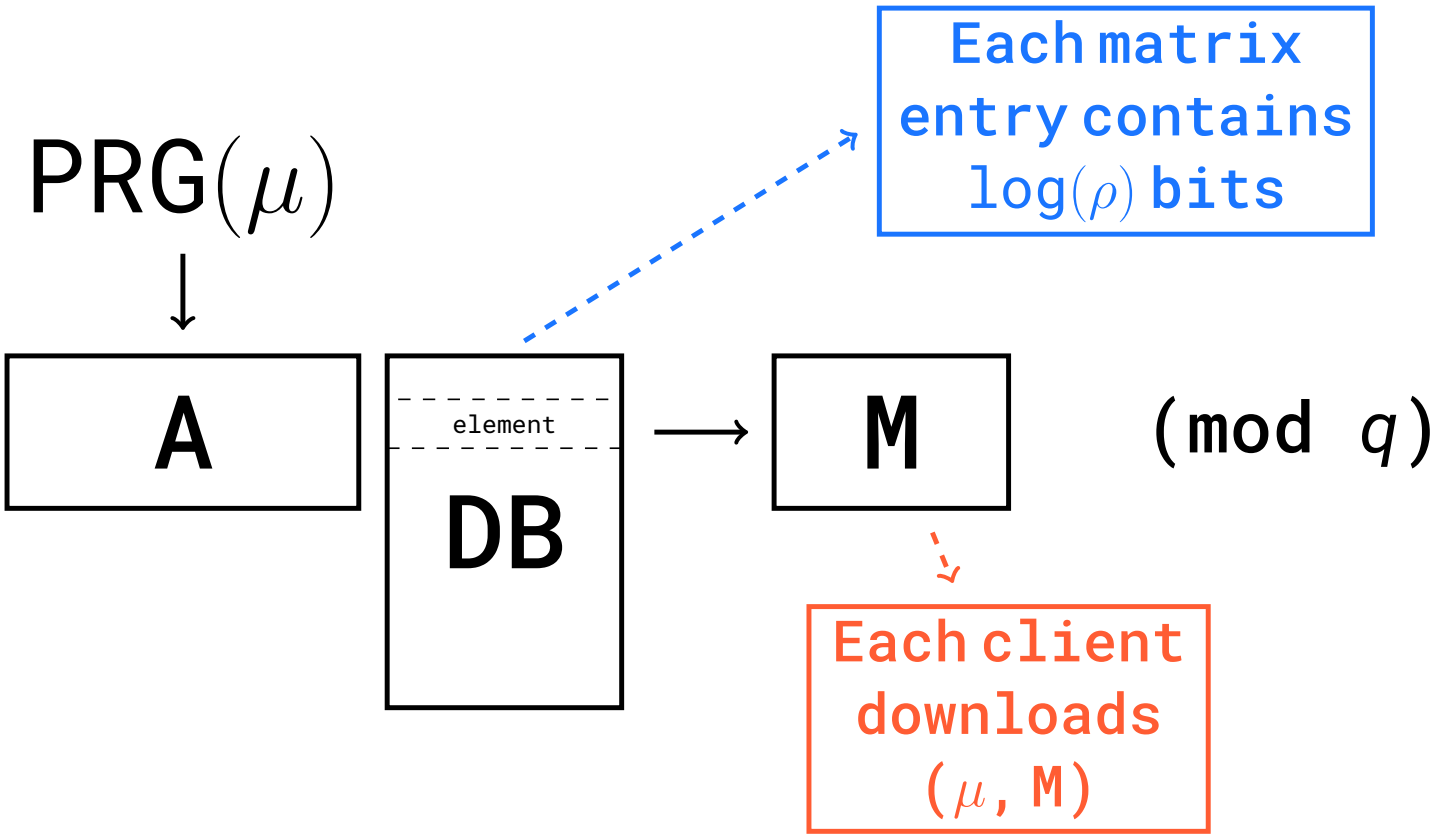
- Simple PIR [Hen+23]
- FrodoPIR [DPC23]
- **Same idea**: Regev-based AHE scheme with global pre-processing

Extra recursive techniques to improve efficiency not covered here (see DoublePIR)

**Idea:**

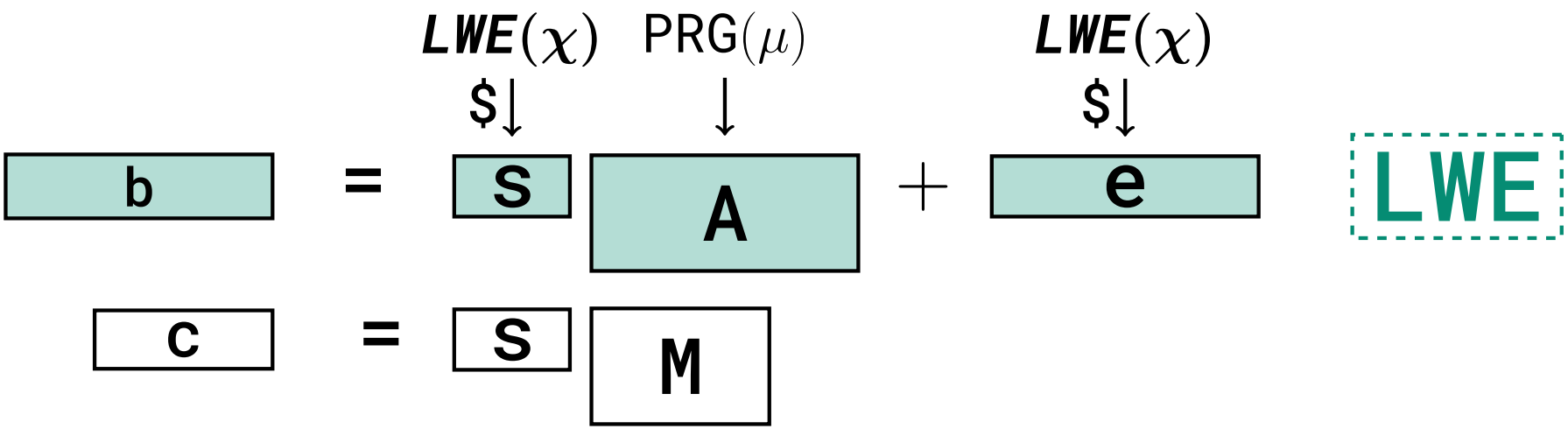
- Server produces DB digest ($\ll |DB|$) offline
- Clients use digest to speed up online phase

## Server preprocessing



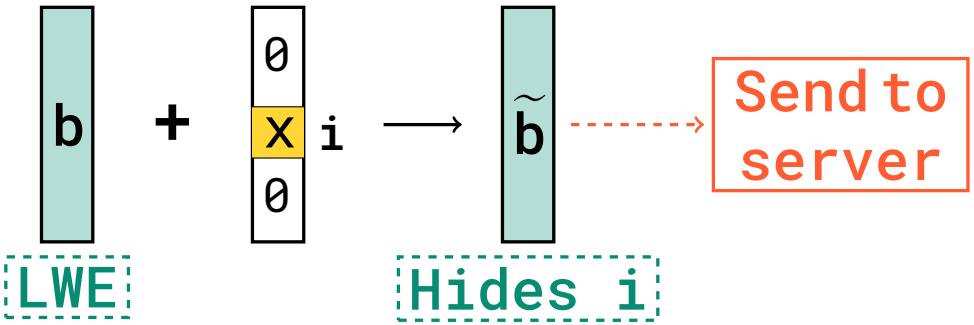

**SIMPLE PIR: OFFLINE**

## Client preprocessing

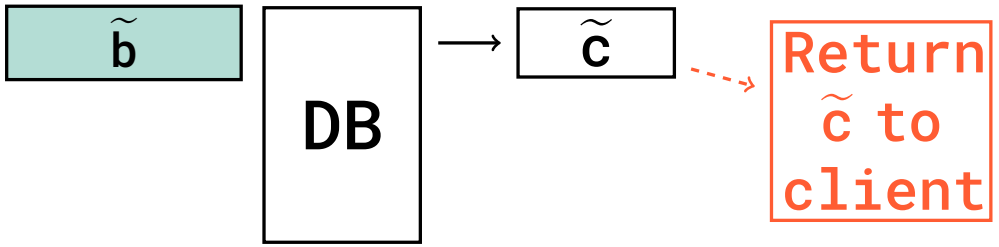



SIMPLE PIR: OFFLINE

## Client Query for Index i



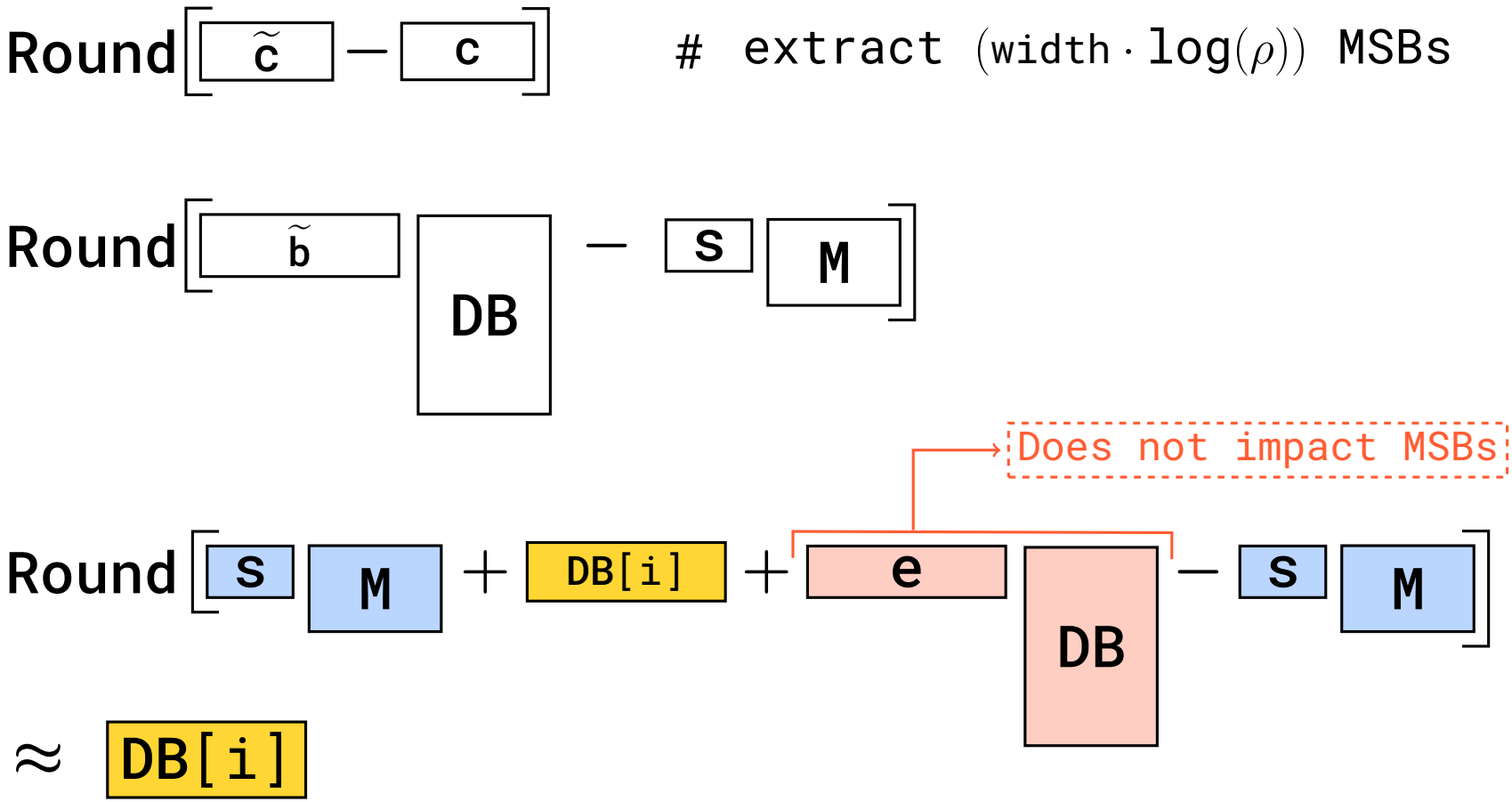

## Server Response




## Client Output

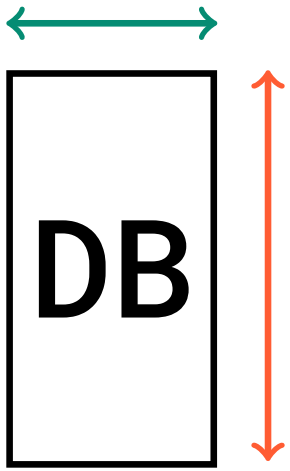
$$\text{Round}\left[\tilde{c} - c\right] \approx \text{DB}[i]$$

**SIMPLE PIR: ONLINE**



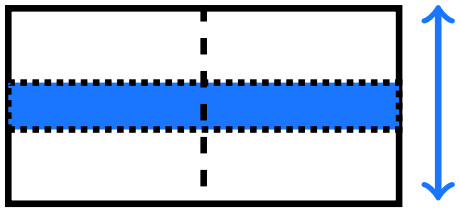

## CORRECTNESS INTUITION

Client upload $\gg$ download




Height defines size of client query

Width defines size of client download



Recover two DB elements per query

Perfectly balanced at $\sqrt{n} \times \sqrt{n}$

- $O(\sqrt{n})$ communication, $O(n)$ computation
- **However**: online phase is very cheap
- Interesting to analyse concrete costs
- Process DBs of up to 1 million 1KB elements
- Compare with streaming-/FHE-based PIR

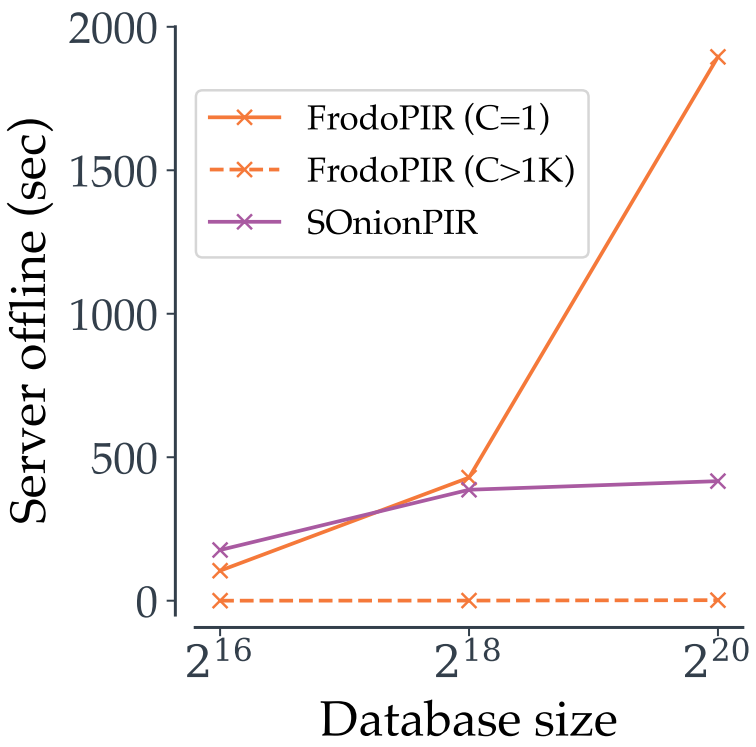
32-bit additions

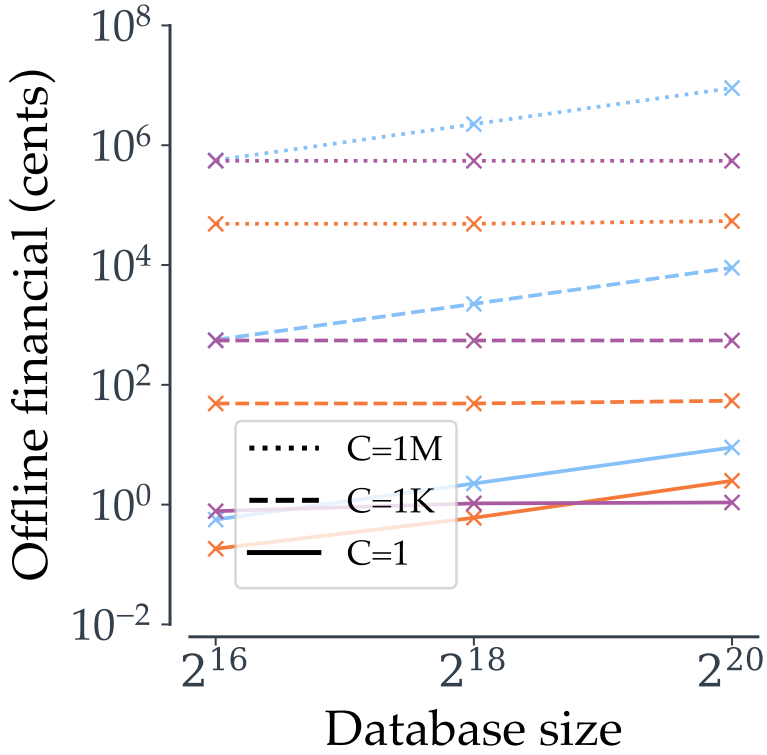
Financials from Amazon EC2

Costs taken from FrodoPIR

# CONCRETE COST COMPARISON

# Offline costs amortize to 0 as |Clients| grows

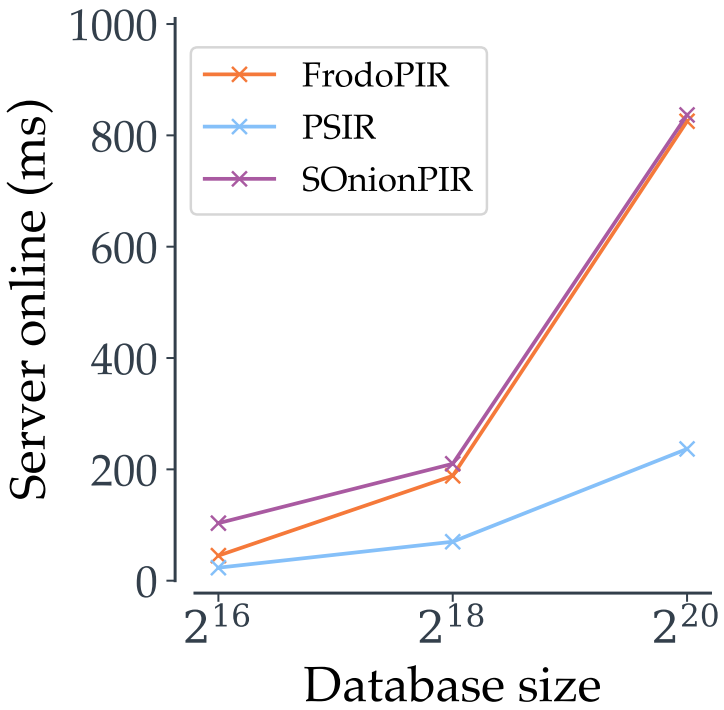


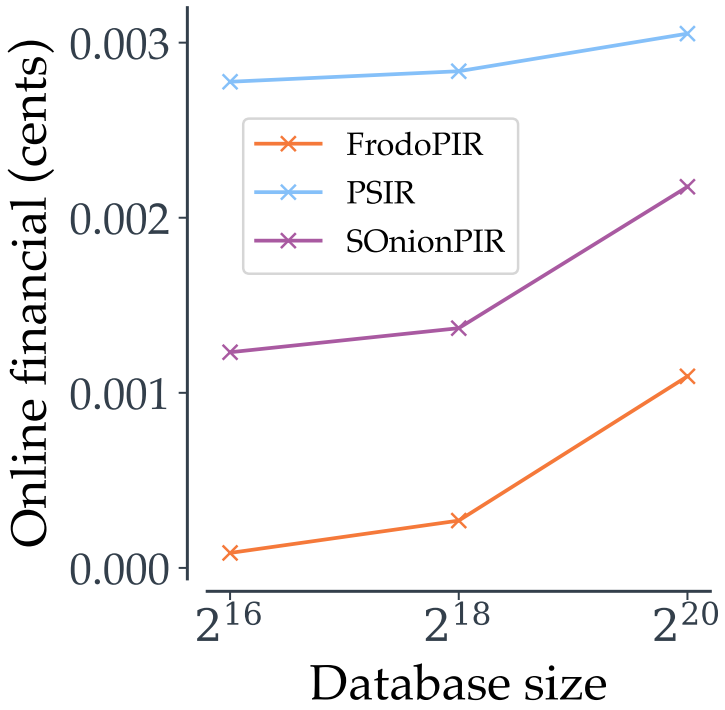




- Simple PIR $\leq$ \$550 of setup costs
- FHE $\approx$ \$5500; Streaming $\approx$ \$90000

## COMPARISON: SERVER

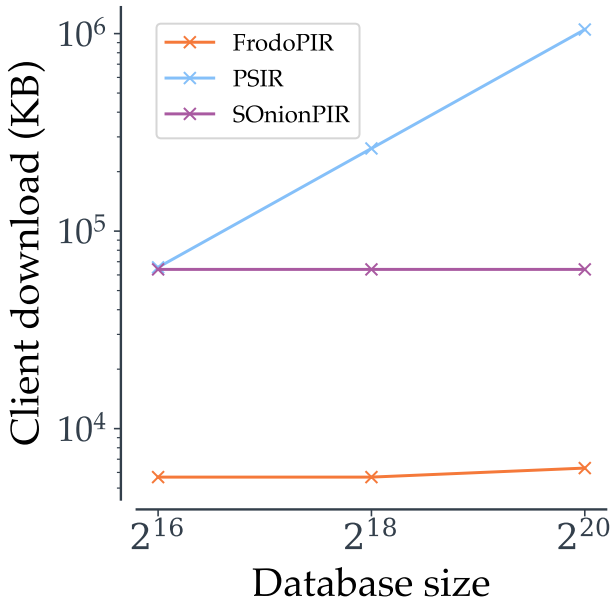
- Online running times are competitive
- Online $ costs $\leq$ 1/2 of alternatives



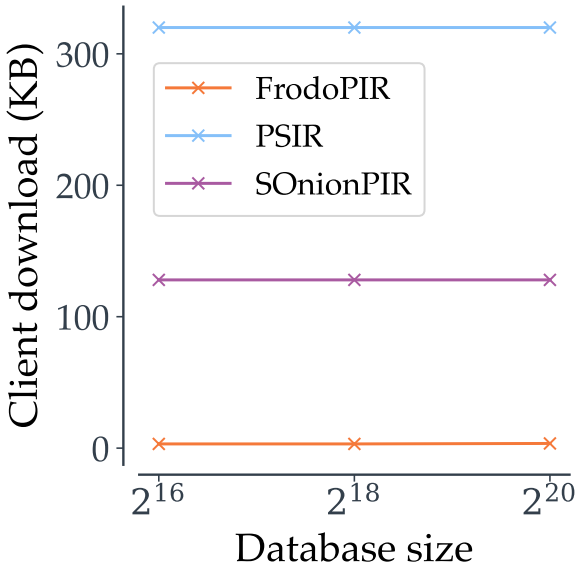



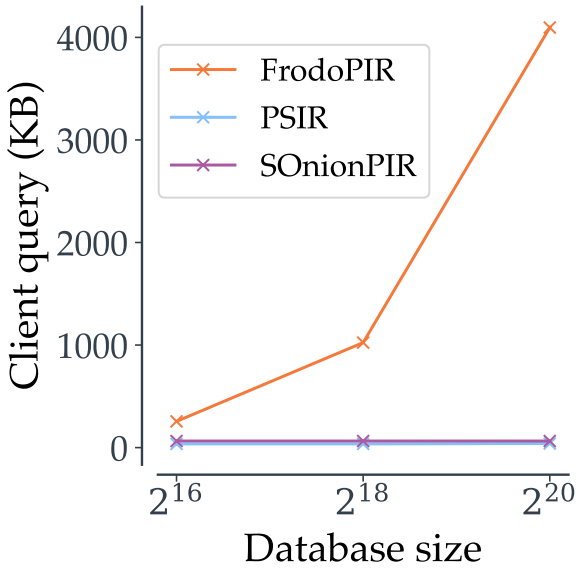

## COMPARISON: SERVER

## Offline



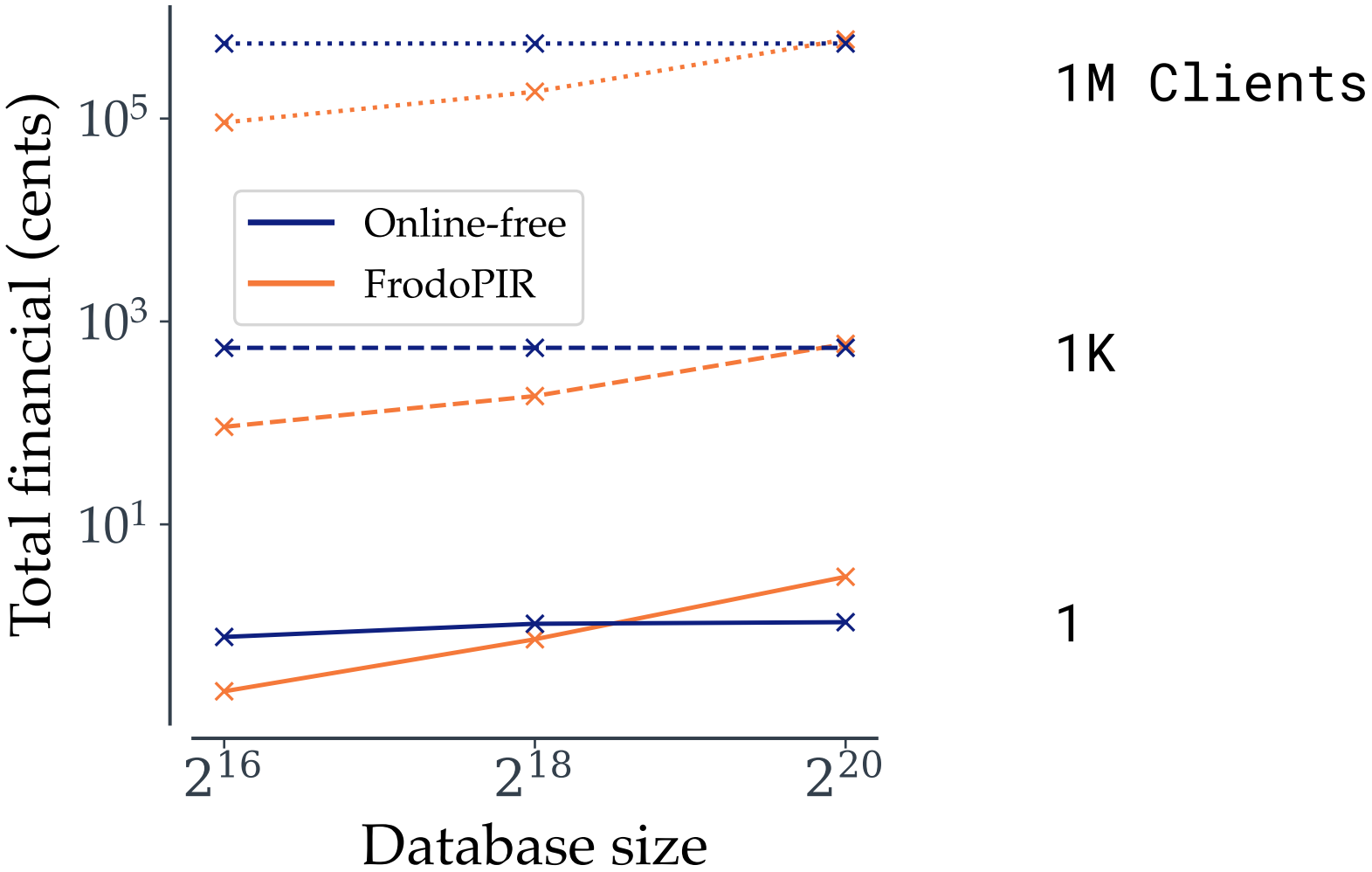

## Online




**COMPARISON: CLIENT**

FHE-based scheme with zero online costs only more expensive for large databases




## COMPARISON: ONLINE-FREE FHE

- Return to LWE-based ($\mathbb{Z}_q$) Regev encryption

Removes dependency on polynomial **rings**...

- Client-independent preprocessing phase

Server produces **amortisable** digest of *DB*

- Concretely efficient online phase

$<$**1s** and **3.6KB** for retrieving 1KB DB elements

- Arithmetic impl. via standard types

**u32 ops.** and 735 lines of code for FrodoPIR

# SUMMARY OF SIMPLE PIR

FUNDAMENTALS

FUNCTIONALITY EXTENSIONS

PERFORMANCE OPTIMISATIONS

**CONCLUSIONS**

- Many real-world applications
- Theoretical efficiency is reasonable
- Concrete costs are low
- Simple, implementable, & efficient schemes

## Many!

- Databases of $> 1M$ elements slow to process
- Rapidly-updating databases very expensive
- Keyword queries and beyond underexplored
- Many unknown applications

**OPEN PROBLEMS?**

- PIR is a fast-moving, exciting area
- Constructions relatively easy to understand
- Good time to start exploring!

## Thanks for listening!

a.davidson@fct.unl.pt https://alxdavids.xyz

[Ang+18] Sebastian Angel et al. “PIR with Compressed Queries and Amortized Query Processing”. In: *2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA*. IEEE Computer Society, 2018, pp. 962–979. DOI: 10.1109/SP.2018.00062. URL: https://doi.org/10.1109/SP.2018.00062.

[BGI15] Elette Boyle, Niv Gilboa, and Yuval Ishai. “Function Secret Sharing”. In: *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II*. Ed. by Elisabeth Oswald and Marc Fischlin. Vol. 9057. Lecture Notes in Computer Science. Springer, 2015, pp. 337–367. DOI: 10.1007/978-3-662-46803-6_12. URL: https://doi.org/10.1007/978-3-662-46803-6%5C_12.

# REFERENCES I

[CGN98] Benny Chor, Niv Gilboa, and Moni Naor. “Private Information Retrieval by Keywords”. In: *IACR Cryptol. ePrint Arch.* (1998), p. 3. URL: http://eprint.iacr.org/1998/003.

[CHK22] Henry Corrigan-Gibbs, Alexandra Henzinger, and Dmitry Kogan. “Single-Server Private Information Retrieval with Sublinear Amortized Time”. In: *Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part II*. Ed. by Orr Dunkelman and Stefan Dziembowski. Vol. 13276. Lecture Notes in Computer Science. Springer, 2022, pp. 3–33. DOI: 10.1007/978-3-031-07085-3_1. URL: https://doi.org/10.1007/978-3-031-07085-3%5C_1.

# REFERENCES II

[Cho+95] Benny Chor et al. “Private Information Retrieval”. In: *36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, USA, 23-25 October 1995*. IEEE Computer Society, 1995, pp. 41–50. DOI: 10.1109/SFCS.1995.492461. URL: https://doi.org/10.1109/SFCS.1995.492461.

[CK20] Henry Corrigan-Gibbs and Dmitry Kogan. “Private Information Retrieval with Sublinear Online Time”. In: *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I*. Ed. by Anne Canteaut and Yuval Ishai. Vol. 12105. Lecture Notes in Computer Science. Springer, 2020, pp. 44–75. DOI: 10.1007/978-3-030-45721-1_3. URL: https://doi.org/10.1007/978-3-030-45721-1%5C_3.

# REFERENCES III

[DG16] Zeev Dvir and Sivakanth Gopi. “2-Server PIR with Subpolynomial Communication”. In: *J. ACM* 63.4 (2016), 39:1–39:15. DOI: 10.1145/2968443. URL: https://doi.org/10.1145/2968443.

[DPC23] Alex Davidson, Gonçalo Pestana, and Sofía Celi. “FrodoPIR: Simple, Scalable, Single-Server Private Information Retrieval”. In: *Proc. Priv. Enhancing Technol.* 2023.1 (2023), pp. 365–383. DOI: 10.56553/popets-2023-0022. URL: https://doi.org/10.56553/popets-2023-0022.

[Gen09] Craig Gentry. “Fully homomorphic encryption using ideal lattices”. In: *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*. Ed. by Michael Mitzenmacher. ACM, 2009, pp. 169–178. DOI: 10.1145/1536414.1536440. URL: https://doi.org/10.1145/1536414.1536440.

# REFERENCES IV

[GI14] Niv Gilboa and Yuval Ishai. “Distributed Point Functions and Their Applications”. In: *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*. Ed. by Phong Q. Nguyen and Elisabeth Oswald. Vol. 8441. Lecture Notes in Computer Science. Springer, 2014, pp. 640–658. DOI: 10.1007/978-3-642-55220-5_35. URL: https://doi.org/10.1007/978-3-642-55220-5%5C_35.

[Hen+23] Alexandra Henzinger et al. “One Server for the Price of Two: Simple and Fast Single-Server Private Information Retrieval”. In: *32nd USENIX Security Symposium, USENIX Security 2023, Anaheim, CA, USA, August 9-11, 2023*. Ed. by Joseph A. Calandrino and Carmela Troncoso. USENIX Association, 2023. URL: https://www.usenix.org/conference/usenixsecurity23/presentation/henzinger.

[KC21] Dmitry Kogan and Henry Corrigan-Gibbs. “Private Blocklist Lookups with Checklist”. In: *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*. Ed. by Michael Bailey and Rachel Greenstadt. USENIX Association, 2021, pp. 875–892. URL: https://www.usenix.org/conference/usenixsecurity21/presentation/kogan.

[KO97] Eyal Kushilevitz and Rafail Ostrovsky. “Replication is NOT Needed: SINGLE Database, Computationally-Private Information Retrieval”. In: *38th Annual Symposium on Foundations of Computer Science, FOCS ’97, Miami Beach, Florida, USA, October 19-22, 1997*. IEEE Computer Society, 1997, pp. 364–373. DOI: 10.1109/SFCS.1997.646125. URL: https://doi.org/10.1109/SFCS.1997.646125.

# REFERENCES VI

[MCR21] Muhammad Haris Mughees, Hao Chen, and Ling Ren. "OnionPIR: Response Efficient Single-Server PIR". In: *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021*. Ed. by Yongdae Kim et al. ACM, 2021, pp. 2292–2306. DOI: 10.1145/3460120.3485381. URL: https://doi.org/10.1145/3460120.3485381.

[PPY18] Sarvar Patel, Giuseppe Persiano, and Kevin Yeo. "Private Stateful Information Retrieval". In: *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018*. Ed. by David Lie et al. ACM, 2018, pp. 1002–1019. DOI: 10.1145/3243734.3243821. URL: https://doi.org/10.1145/3243734.3243821.

# REFERENCES VII

[SC07] Radu Sion and Bogdan Carbunar. “On the Practicality of Private Information Retrieval”. In: *Proceedings of the Network and Distributed System Security Symposium, NDSS 2007, San Diego, California, USA, 28th February - 2nd March 2007*. The Internet Society, 2007. URL: https://www.ndss-symposium.org/ndss2007/practicality-private-information-retrieval/.

[Zho+23] Mingxun Zhou et al. “Piano: Extremely Simple, Single-Server PIR with Sublinear Server Computation”. In: *IACR Cryptol. ePrint Arch.* (2023), p. 452. URL: https://eprint.iacr.org/2023/452.